



Grado en Ingeniería en Tecnologías de Telecomunicación

Trabajo Final de grado

Detección de objetos mediante el uso de ondas milimétricas

Autor

Rubén Rodríguez Gómez

Supervisado por

Matthew Caesar

Madrid

Junio 2024

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
..... Detección de objetos mediante el uso de
ondas milimétricas.....

.....
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico2023-2024..... es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Fdo.:



Fecha: 12/ 06/ 2024

Autorizada la entrega del proyecto

Matthew Caesar Chapman

Fdo.:



Fecha: 12/ 06/ 2024



Grado en Ingeniería en Tecnologías de Telecomunicación

Trabajo Final de grado

Detección de objetos mediante el uso de ondas milimétricas

Autor

Rubén Rodríguez Gómez

Supervisado por

Matthew Caesar

Madrid

Junio 2024

Resumen

DETECCIÓN DE OBJETOS MEDIANTE EL USO DE ONDAS MILIMÉTRICAS

Autor: Ruben Rodriguez Gomez

Director: Professor Matthew Caesar

Entidad Colaboradora: University of Illinois at Urbana-Champaign

Palabras clave: ondas milimétricas, precisión, condiciones de visibilidad, PointNet, Machine Learning

Introducción

La detección de objetos identifica y localiza objetos en el entorno usando tecnologías avanzadas, cruciales en navegación autónoma, seguridad, vigilancia y control del tráfico. Históricamente, desde técnicas visuales antiguas hasta el radar en la Segunda Guerra Mundial, estas tecnologías han mejorado significativamente la seguridad, especialmente en la aeronáutica. En el siglo XXI, se centran en vehículos autónomos con sensores de ondas milimétricas y LIDAR, promoviendo una conducción segura y eficiente. Las tecnologías basadas en el espectro visible (380-700 nm) capturan imágenes de alta resolución, esenciales para vigilancia e inspección, pero tienen limitaciones: son sensibles a la iluminación, condiciones meteorológicas adversas, tienen limitada distancia de detección y son vulnerables a obstáculos físicos. Además, pueden plantear problemas de privacidad. Estas limitaciones han impulsado la exploración de tecnologías alternativas, como las ondas milimétricas, que operan entre 30GHz y 300GHz, capaces de detectar con precisión en condiciones de visibilidad deficientes, además de ser capaces de atravesar objetos. Es por ello por lo que este proyecto se centrará en desarrollar un sistema de detección usando sensores de ondas milimétricas, con el fin de ver como funcionan este tipo de tecnologías en situaciones donde las tecnologías de detección no funcionan.

Definición del Trabajo

Los objetivos principales de este proyecto son los siguientes:

- Reconocimiento de movimiento de personas sin ningún objeto de por medio:
Para comprobar la efectividad del sistema desarrollado se realizará la detección en dos escenarios de visibilidad distintos: con luz y sin luz. Esto es para comprobar si realmente dicho sistema es más óptimo que uno que utilizase tecnologías del espectro visible (las cuales no son óptimas para la detección en condiciones de visibilidad reducida o nula). Para el cumplimiento de dicho objetivo, los resultados tendrán que demostrar que el modelo es capaz de clasificar e identificar el movimiento que la persona realice con cierto nivel de precisión.
- Reconocimiento de movimiento de personas con una silla de por medio:
Dado que una de las ventajas que ofrecen las ondas milimétricas (aparte de ofrecer mayor precisión en condiciones de visibilidad reducida o nula que las tecnologías basadas en el espectro visible), es su capacidad de atravesar objetos no metálicos. Es por ello que para cumplir este objeto el modelo tendrá que ser capaz de identificar y detectar y clasificar el movimiento que la persona realice con la silla, con cierto nivel de precisión. Este escenario se realizará con luz y sin luz también.
- Reconocimiento de movimiento de personas con una mesa de por medio:
Con este objetivo se pretende conseguir exactamente lo mismo que para los objetivos anteriores pero esta vez empleando como obstáculo un objeto más voluminoso y grande como bien puede ser una mesa. De nuevo, este escenario se realizará con luz y sin luz.
- Diseño de una metodología óptima para la detección de objetos: Este primer objetivo será el primer paso que se deberá tomar en este trabajo para poder abordar la labor de detección de forma correcta. Se considerará que este objetivo se ha logrado con éxito si se consigue lo siguiente:
 - Se considerará un diseño óptimo si la metodología diseñada basada en métodos previamente utilizados en la labor de detección con éxito.
 - Se considerará un diseño óptimo si se consiguen cumplir los tres objetivos principales comentados en este apartados.

Los objetivos secundarios de este proyecto son:

- Explorar la importancia que tiene la ética en el desarrollo de este tipo de tecnologías.
- Ver futuras ramas en donde las aplicaciones de ondas milimétricas podrían ser una solución.

Metodología

Para este trabajo se ha desarrollado una metodología basada en la teoría de la técnica denominada Vision-Fusion, la cual implementa datos provenientes de distintos sensores [1]. Por tanto, el diseño del trabajo se puede visualizar en la siguiente Ilustración:

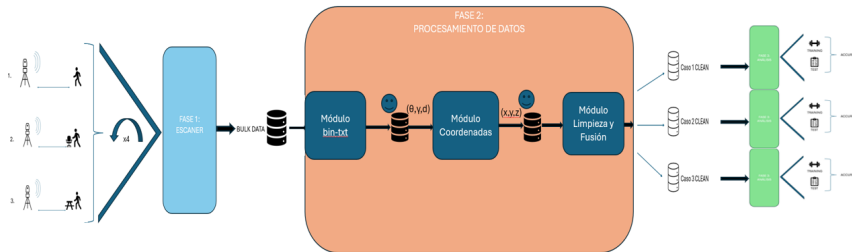


Figura 1: Metodología del trabajo

Dicha metodología está basada en tres fases:

- **Fase 1: Escaner.** En esta fase se formarán los tres escenarios pretendidos en los objetivos principales para proceder a la labor de detección. El procedimiento que se seguirá es el siguiente: se enciende la simulación del radar, la persona avanza progresivamente hasta el radar, se detiene la simulación. Este proceso se realizará 4 veces, con el fin de luego en la parte de fusión de datos, combinar dichos resultados para obtener un conjunto de datos más representativo. Para dicha labor, se empleará el sensor IRW6843OPEVM, el cual es óptimo para aplicaciones sencillas de detección de objetos dado que consume muy poca potencia (1.5W).
- **Fase 2: Procesamiento de datos.** En esta fase, se realizarán tres procesados de datos distintos. Un primer procesado para hacer legibles los datos obtenidos por el radar. Un segundo procesado que se encargará de transformar las coordenadas obtenidas por el sensor al plano (x,y,z). Un último procesado, que se encargará de limpiar los datos y hacer la fusión mencionada en la primera fase, con el fin de ya obtener un conjunto de datos robusto y limpio para su posterior implementación en la tercera fase de esta metodología.
- **Fase 3: Análisis computacional.** En esta fase se implementará el modelo PointNet. Dicho modelo, es no solo óptimo para poder identificar nubes de puntos de forma precisa si no además que este modelo no tiene en cuenta el tiempo y el orden en el que los puntos están representados. De este modelo, se obtendrán valores de pérdida de información y precisión para evaluar

hasta que punto los datos son representativos de cada uno de los escenarios escaneados.

Analisis de resultados

Tras aplicar el modelo, se puede observar en las figuras proporcionadas (ver Figura 2), que en cada uno de los tres escenarios propuestos para este trabajo, y en ambas condiciones de luminosidad propuestas (óptima y nula), se puede apreciar como en todos los casos se puede ver como existe un movimiento lineal hacia el radar, siendo en todos los casos, la nube de puntos más concentrada, la fase inicial del desplazamiento (en la que el sujeto empieza a acelerar bien sea con o sin objetos). La segunda fase se puede apreciar también (cuando el sujeto emprende el movimiento ya a ritmo constante), siendo la nube de puntos que sigue estando concentrada pero algo más dispersa que la de la primera fase. Luego la tercera fase de movimiento (que es cuando la simulación se detiene y por tanto la persona como el objeto involucrado están quietos) se puede apreciar que difieren en los distintos escenarios probados: en el escenario que no involucra objetos, se puede apreciar como la tercera fase se identifica claramente dado que hay una serie de puntos más dispersos que el resto, incluso podría decirse que son aleatorios. Esto se debe a que el radar introduce una variable de ruido para intentar posicionar a la persona una vez esta ya está quieta.

En el resto de escenarios, se puede apreciar como estas dos fases se fusionan en una. Esto se debe a que el movimiento realizado con los objetos no haya sido completo y que por tanto haya sido demasiado corto para que el radar identifique una tercera fase. Aún así, también se puede apreciar en estos dos escenarios como existen una serie de puntos mucho más dispersos que el resto, que se corresponderían con los clasificados como una tercera clase en el primer escenario.

En cuanto a las métricas, estos modelos presentan no solo robustez (dado que los resultados en condiciones luminosas nulas como óptimas son muy similares y siguen el mismo patrón), sino que demuestran efectividad a la hora de identificar el movimiento realizado.

Por tanto teniendo en cuenta que, los valores de precisión rondan el 93% como mínimo, y que provienen de un modelo totalmente entrenado y óptimo, se puede determinar que los resultados encontrados indican que el sistema desarrollado en este trabajo es óptimo y eficaz para la detección de objetos y personas.

Conclusiones del estudio

Tras observar los resultados obtenidos, se puede determinar que los objetivos propuestos para este trabajo se han satisfecho con creces, dado que no solo se ha podido demostrar que el sistema desarrollado es eficaz y óptimo para la detección de objetos y personas, sino que se ha podido demostrar que los sistemas basados en ondas milimétricas son consistentes al proporcionar resultados buenos tanto en

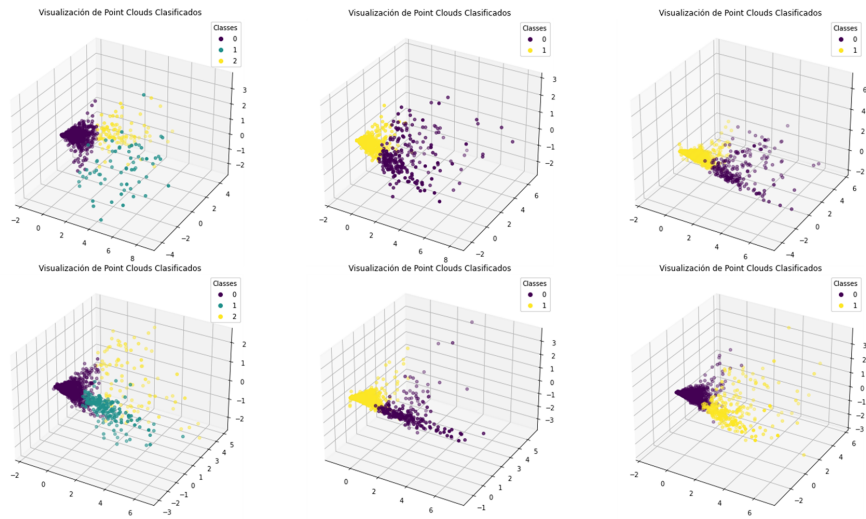


Figura 2: Visualizaciones 3D

condiciones de luminosidad óptimas como en condiciones de luminosidad nula.

Los pasos a seguir en este trabajo para alcanzar un sistema que sea capaz no solo identificar los movimientos realizados sino que sea capaz de construir una imagen clara que describa los objetos y personas que rondan un entorno, es necesario indagar sobre la aplicación de técnicas de Computer-Vision en el sistema desarrollado en este trabajo, implementar un sensor más para poder mapear de forma más óptima los puntos detectados y la implementación de la técnica SMOTE para evitar problemas de sobreajuste.

Bibliografía

1. Liggins, M. E., Hall, D. L., & Llinas, J. (2008). Handbook of Multisensor Data Fusion: Theory and Practice: Second Edition. In Handbook of Multisensor Data Fusion: Theory and Practice: Second Edition.

Abstract

OBJECT DETECTION USING MILLIMETER WAVES

Autor: Ruben Rodriguez Gomez

Director: Professor Matthew Caesar

Partnership Entity: University of Illinois at Urbana-Champaign

Key Words: Millimeter wave, accuracy, visibility conditions, PointNet, Machine Learning

Introduction

Object detection identifies and locates objects in the environment using advanced technologies, crucial in autonomous navigation, security, surveillance, and traffic control. Historically, from ancient visual techniques to radar in World War II, these technologies have significantly improved safety, especially in aviation. In the 21st century, they focus on autonomous vehicles with millimeter-wave and LIDAR sensors, promoting safe and efficient driving. Technologies based on the visible spectrum (380-700 nm) capture high-resolution images, essential for surveillance and inspection, but they have limitations: they are sensitive to lighting, adverse weather conditions, have limited detection range, and are vulnerable to physical obstacles. Additionally, they can pose privacy issues. These limitations have driven the exploration of alternative technologies, such as millimeter waves, which operate between 30GHz and 300GHz, capable of precise detection in poor visibility conditions and of penetrating objects. Therefore, this project will focus on developing a detection system using millimeter-wave sensors to evaluate how these technologies perform in situations where traditional detection technologies fail.

Project Definition

The main objectives of this project are as follows:

- *Recognition of people movement without any object in between:* To verify the effectiveness of the developed system, detection will be performed in two different visibility scenarios: with light and without light. This is to determine if the system is indeed more optimal than one that uses visible spectrum technologies (which are not optimal for detection in reduced or zero visibility conditions). To achieve this objective, the results must demonstrate that the model is capable of classifying and identifying the movement of the person with a certain level of precision.

- Recognition of people movement with a chair in between: Given that one of the advantages offered by millimeter waves (apart from offering greater precision in reduced or zero visibility conditions compared to technologies based on the visible spectrum) is their ability to penetrate non-metallic objects. Therefore, to achieve this objective, the model must be capable of identifying, detecting, and classifying the movement of the person with the chair, with a certain level of precision. This scenario will also be performed with and without light.
- Recognition of people movement with a table in between: With this objective, the aim is to achieve the same results as the previous objectives, but this time using a larger and bulkier object such as a table as an obstacle. Again, this scenario will be performed with and without light.
- Design of an optimal methodology for object detection: This first objective will be the initial step to be taken in this work to correctly address the task of detection. This objective will be considered successfully achieved if the following are accomplished:
 - The design will be considered optimal if the methodology developed is based on methods previously used successfully in detection tasks.
 - The design will be considered optimal if the three main objectives mentioned in this section are achieved.

The secondary objectives of this project are:

- Explore the importance of ethics in the development of these technologies
- Identify future areas where millimeter wave applications could be a solution.

Methodology

For this work, a methodology based on the theory of the Vision-Fusion technique has been developed, which implements data from different sensors [1]. Therefore, the work design can be visualized in the following illustration (see Figure 3): This methodology is based on three phases:

- **Phase 1: Scanning.** In this phase, the three scenarios described in the main objectives will be set up for detection. The procedure is as follows: start the radar simulation, the person progressively approaches the radar, stop the simulation. This process will be repeated four times to later combine the results in the data fusion part, obtaining a more representative dataset. The IRW6843OPEVM sensor will be used for this task, which is optimal for simple object detection applications due to its low power consumption (1.5W).

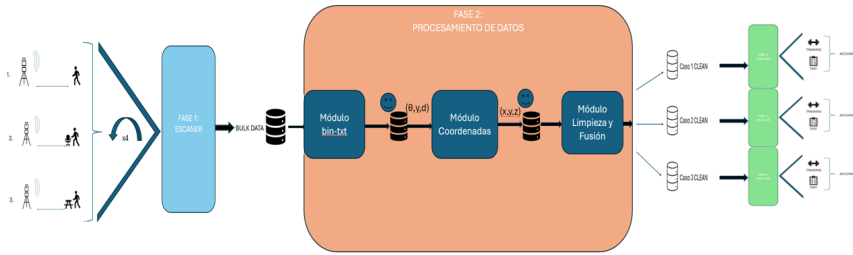


Figure 3: Methodology

- Phase 2: Data processing.** In this phase, three different data processing steps will be carried out. The first step will make the radar data readable. The second step will transform the coordinates obtained by the sensor to the (x,y,z) plane. The final step will clean the data and perform the fusion mentioned in the first phase, resulting in a robust and clean dataset for subsequent implementation in the third phase of this methodology.
- Phase 3: Computational analysis.** In this phase, the PointNet model will be implemented. This model is not only optimal for accurately identifying point clouds but also does not consider the time and order in which the points are represented. This model will provide information loss and accuracy values to evaluate how representative the data is for each of the scanned scenarios.

Results Analysis

After applying the model, it can be observed in the provided figures (see Figure 4) that in each of the three proposed scenarios for this work, and under both lighting conditions (optimal and none), a linear movement towards the radar is visible in all cases. In all cases, the most concentrated point cloud represents the initial phase of the movement (when the subject starts to accelerate, either with or without objects). The second phase can also be observed (when the subject is moving at a constant pace), with the point cloud remaining concentrated but slightly more dispersed than in the first phase. The third phase of movement (when the simulation stops and thus the person and the object involved are stationary) can be seen to differ in the various tested scenarios: in the scenario without objects, the third phase is clearly identifiable due to a series of points more dispersed than the rest, which could even be considered random. This is because the radar introduces a noise variable to try to position the person once they are stationary. In the other scenarios, these two phases merge into one. This is because the movement performed with the objects was incomplete and therefore too short for the radar to identify a third phase. Nevertheless, in these two scenarios, there is also a series of points much more dispersed than the rest, corresponding to those classified as a

third class in the first scenario. Regarding metrics, these models not only show robustness (since the results under both no-light and optimal lighting conditions are very similar and follow the same pattern) but also demonstrate effectiveness in identifying the performed movement.

Therefore, considering that the precision values are around 93% at a minimum and that they come from a fully trained and optimal model, it can be determined that the results found indicate that the system developed in this work is optimal and effective for the detection of objects and people.

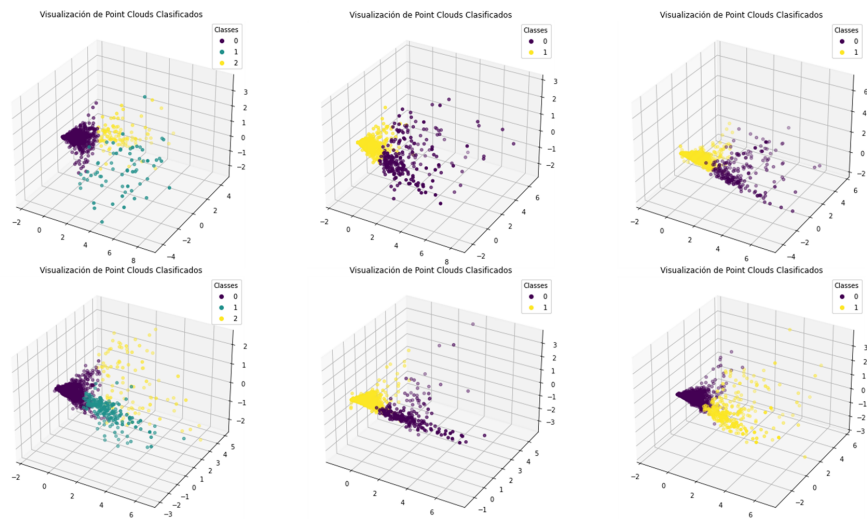


Figure 4: 3D Visualizations

Conclusions

After observing the obtained results, it can be determined that the proposed objectives for this work have been greatly exceeded. Not only has it been demonstrated that the developed system is effective and optimal for the detection of objects and people, but it has also been proven that systems based on millimeter waves are consistent in providing good results both in optimal lighting conditions and in complete darkness.

The steps to follow in this work to achieve a system capable of not only identifying the movements made but also constructing a clear image that describes the objects and people surrounding an environment include exploring the application of Computer Vision techniques in the developed system, implementing an additional sensor to map the detected points more optimally, and implementing the SMOTE technique to avoid overfitting issues.

References

1. Liggins, M. E., Hall, D. L., & Llinas, J. (2008). Handbook of Multisensor Data Fusion: Theory and Practice: Second Edition. In Handbook of Multisensor Data Fusion: Theory and Practice: Second Edition.

Gracias

Me gustaría agradecer a Matthew Caesar, Zihan, Shuo, Aryan y John por haberme aceptado en su proyecto en la universidad de Illinois y por haberme ayudado a integrarme con ellos. No me cabe la menor duda de que sin su ayuda, este trabajo no habría sido posible.

Me gustaría dar también las gracias a mis padres y mi hermana, por haberme apoyado (incluso cuando no lo merecía). No puedo ser más afortunado de tenerlos. Gracias a ellos soy el hombre que soy hoy en día.

Por último, me gustaría agradecer a Ruben (a mi), por haber luchado sin parar por cumplir tus sueños. Al final, después de todo lo que has pasado ha merecido la pena. Lo bueno está por llegar, ¡sigue luchando!

Salga bien este trabajo o no, simplemente el hecho de que lo haya terminado es motivo para sentirme muy orgulloso. ¡Espero que les guste!

Índice general

1	Introducción	1
1.1	Contexto histórico	1
1.2	Tecnologías de detección convencionales	2
1.2.1	Limitaciones de las tecnologías de detección convencionales	3
1.3	Las ondas milimétricas y sus aplicaciones en la detección de objetos	4
1.4	Comparativa entre las tecnologías de detección que aplican ondas milimétricas y las convencionales	5
1.5	Motivación del trabajo	7
1.6	Objetivos del trabajo	7
2	Estado de la cuestión	9
2.1	Exploración de los valores de precisión en investigaciones recientes	9
2.1.1	Factores que pueden condicionar los valores de precisión en la labor de detección	11
2.2	Hardware óptimo para la detección de objetos y personas: Vision-Fusion	14
2.3	Algoritmos óptimos para el análisis de datos	20
2.3.1	PointNet	22
2.3.2	SECOND	24
2.3.3	VoxelNet	26
3	Metodología	29
3.1	Diseño del trabajo	29
3.1.1	Planificación temporal	30
3.2	Fase 1: Escaner	31
3.2.1	Sensor IT IWR6843AOPEVM	33
3.2.2	Recogida de datos	35
3.3	Fase 2: Procesamiento de datos	38
3.3.1	Primer procesado: Obtención datos útiles	38
3.3.2	Segundo procesado: Obtención coordenadas 3D	42

3.3.3	Tercer procesado: Fusión de muestras obtenidas y limpieza de datos	46
3.4	Fase 3: Análisis	53
3.4.1	Software: Aplicación arquitectura PointNet	54
3.5	Implementación modular final del trabajo	60
4	Analisis de los resultados	61
4.1	Condición luminosa óptima	62
4.1.1	Escenario 1: Detección de objetos sin objetos	62
4.1.2	Escenario 2: Detección de objetos con una silla	65
4.1.3	Escenario 3: Detección de objetos con una mesa	68
4.2	Condición luminosa nula	71
4.2.1	Escenario 1: Detección de objetos sin objetos	71
4.2.2	Escenario 2: Detección de objetos con una silla	74
4.2.3	Escenario 3: Detección de objetos con una mesa	77
4.3	El problema de la privacidad de datos	80
4.3.1	Vigilancia Masiva y Derecho a la Privacidad	81
4.3.2	Seguridad de los Datos y Acceso No Autorizado	83
4.3.3	Discriminación algorítmica	85
5	Discusiones Finales	89
5.1	Conclusiones del estudio	91
5.1.1	Posibles mejoras y siguientes pasos	94
5.2	Trabajos futuros	97
5.3	Últimas palabras	99
	Bibliografía	101
6	Apéndices	105
6.1	Código relevante para el procesamiento de datos	105
6.1.1	Código para la poder hacer legibles los datos obtenidos por el radar: de .bin a .txt	105
6.1.2	Código empleado para el primer procesado de los datos obtenidos por el radar: Solo coordenadas (x,y,z)	106
6.1.3	Código empleado para el segundo procesado de los datos obtenidos por el radar: Eliminación de datos erróneos	107
6.2	Código completo de PointNet	109
6.2.1	Modelo PointNet y estimación del rendimiento	109
6.2.2	Visualizaciones 3D	115
6.2.3	Matriz de confusión y métricas asociadas	115
6.3	Objetivos de Desarrollo Sostenible (ODS)	117

Índice de figuras

2.1	Funcionamiento Metodología Vision-Fusion	20
2.2	¿Qué son los Point Clouds?	21
3.1	Vista holística del trabajo	30
3.2	Diagrama de Gant del trabajo	32
3.3	Sensor IWR6843AOPEVM	33
3.4	Indicaciones puertos USB del sensor	35
3.5	Herramienta de visualización	35
3.6	Procedimiento físico para la obtención de datos	36
3.7	Esquema interno de la Fase 1: Escaner	37
3.8	Esquema general de la Fase 2	38
3.9	Archivo en formato binario	39
3.10	Esquema del primer procesado	42
3.11	Datos legibles resultantes del primer procesado	43
3.12	Representación gráfica de las magnitudes del sensor	44
3.13	Esquema segundo procesado	46
3.14	Esquema tercer procesado	46
3.15	Ejemplo de elipsis	48
3.16	Ejemplo de datos no útiles	48
3.17	Concatenación del archivo localizado en “./binData-1/Dark/People_moving_with_chair”	49
3.18	Histogramas de la distribución del número de puntos detectados por archivo	51
3.19	Esquema completo de la fase 2	53
3.20	Esquema de la fase 3	54
3.21	Implementación Fase 3	59
3.22	Implementación final	60
4.1	Representación 3D Escenario 1 con luz	63
4.2	Gráfica de pérdida de información Escenario 1 con luz	64
4.3	Matriz de confusión Escenario 1 con luz	65
4.4	Representacion 3D Escenario 2 con luz	66
4.5	Gráfica de pérdida de información Escenario 2 con luz	67

4.6	Matriz de confusión Escenario 2 con luz	68
4.7	Representación 3D Escenario 3 con luz	69
4.8	Gráfica de pérdida de información Escenario 3 con luz	70
4.9	Matriz de confusión Escenario 3 con luz	70
4.10	Representación 3D Escenario 1 sin luz	72
4.11	Gráfica de pérdida de información Escenario 1 sin luz	73
4.12	Matriz de confusión Escenario 1 sin luz	74
4.13	Representación 3D Escenario 2 sin luz	75
4.14	Gráfica de pérdida de información Escenario 2 sin luz	76
4.15	Matriz de confusión Escenario 2 sin luz	77
4.16	Representación 3D Escenario 3 sin luz	78
4.17	Gráfica de pérdida de información Escenario 3 sin luz	79
4.18	Matriz de confusión Escenario 3 sin luz	80
6.1	Objetivos del Desarrollo Sostenible	118

Índice de Cuadros

2.1	Especificaciones TI AWR1642BOOST	16
2.2	Especificaciones Infineon BGT60ATR24C	17
2.3	Especificaciones NXP MR3003	17
2.4	Especificaciones Analog Devices ADAR1000	18
2.5	Especificaciones Continental ARS 408-21	18
2.6	Comparativa de algoritmos de deep learning	28
3.1	Especificaciones IWR6843AOPEVM	34

Índice de fragmentos de código

3.1	Proceso de lectura de archivos	40
3.2	Proceso de descriptación del formato binario	40
3.3	Llamamiento a la función de descriptación	41
3.4	Transformación de coordenadas	45
3.5	Limpieza de datos	47
3.6	Transformación de coordenadas	49
3.7	Algoritmo para procesar niveles del umbral	51
3.8	Añadir etiquetas a conjunto de datos	55
3.9	creación de los conjuntos de datos	56
3.10	Creación clase necesaria	57
3.11	Busqueda mejores parámetros	57
6.1	Función completa <code>decode_uart(file_path)</code>	105
6.2	Función completa transformación coordenadas	106
6.3	Limpieza y Parseado completo	107
6.4	Implementación modelo	109
6.5	Visualizaciones 3D	115
6.6	Código para representar matrices de confusión	115

Capítulo 1

Introducción

1.1 Contexto histórico

La detección de objetos se encarga de identificar y de localizar objetos en el entorno mediante el uso de tecnologías y métodos. Esta labor ha sido una tarea fundamental en múltiples disciplinas como la navegación autónoma, la seguridad y vigilancia, e incluso el control del tráfico. Esto se debe esencialmente a que las tecnologías de detección de objetos aportan precisión y fiabilidad que permiten mejorar la eficiencia y efectividad en estas áreas, haciendo posible la identificación de amenazas potenciales y la posterior toma de decisiones informadas en tiempo real en base a los datos recogidos.

Las primeras innovaciones en cuanto a la detección de objetos se remontan a la antigüedad, cuando se aplicaban técnicas visuales para la consiguiente identificación de objetos o amenazas en un entorno concreto. A partir del siglo XX, se fueron desarrollando métodos más sofisticados, como bien el radar durante la Segunda Guerra Mundial, el cual revolucionó la capacidad de detección a largas distancias y en condiciones adversas. Concretamente, en la década de 1940 según el estudio de Buderer en 1996 [81], el radar permitió a los aliados detectar aviones y barcos enemigos localizados a grandes distancias en condiciones meteorológicas adversas, lo que resultó crucial para la victoria en muchas batallas. Dicho radar sentó las bases para futuras innovaciones en la detección remota. Prueba de dichas innovaciones se reflejan en el trabajo de Skolnik en 2008 [62] donde explica que entre las décadas de 1960 y 1970, la tecnología de detección de objetos se integró en la aeronáutica civil, mejorando significativamente la seguridad y eficiencia del trabajo aéreo. Como resultado de ello se consiguieron reducir el riesgo de colisiones de aeronaves y optimizar las rutas de vuelo.

Ya en el siglo XXI, las aplicaciones de la detección de objetos se ha centrado en la industria de los vehículos autónomos. Un estudio reciente de Thrun et al. en

2006 [66] detalla como los sensores de ondas milimétricas combinados con otras tecnologías como LIDAR o incluso cámaras, permiten a los vehículos autónomos detectar y actuar con rapidez ante cualquier situación de peligro con una precisión sin precedentes. De nuevo, estos avances en la detección de objetos mejoran la seguridad de los vehículos autónomos sino que también promueven una conducción más eficiente y sostenible.

Estos ejemplos históricos reflejan la evolución y creciente interés en el desarrollo de tecnologías de detección de objetos cada vez más avanzado, enfatizando su relevancia en el aumento de la seguridad y la eficiencia en una variedad de aplicaciones a lo largo del tiempo.

1.2 Tecnologías de detección convencionales

Tras haber explorado la historia de las tecnologías de detección de objetos se ha podido identificar la importancia del desarrollo de tecnologías que sean cada vez más precisas para poder ser tanto más eficiente como más eficaz en lo que se refiere a la detección de objetos. Lo cierto es que aunque cada vez se innove más con este tipo de tecnologías (como se ha comentado con las implementaciones de ondas milimétricas en los sistemas operativos de coches autónomos), convencionalmente las tecnologías de detección de objetos se basan en ondas del espectro visible. Dicho espectro es la porción del espectro electromagnético que comprende las longitudes de onda entre los 380 y 700 nanómetros (longitudes de onda que hacen que dichas ondas sean "visibles" por el ojo humano) y que comprende todos los colores del arcoiris [20]. Las tecnologías que emplean el espectro visible son comúnmente utilizadas en el ámbito de la detección de objetos debido a que como el ser humano tiene la capacidad para ver este rango de longitudes de ondas, hace que dichas tecnologías sean intuitivamente útiles y fáciles de interpretar para la detección de objetos. En adición a lo dicho, el espectro visible se ha vuelto fundamental en la detección de objetos debido a varias razones:

En primer lugar, las tecnologías que operan en el espectro visible son ampliamente accesibles y relativamente económicos si se comparan con otras tecnologías (como por ejemplo aquellas que emplean sensores infrarrojos u ondas milimétricas) [65]. Esto da lugar a tecnologías tan vistas como aquellas que monitorean el tráfico, o incluso aquellas que pretenden dar seguridad del hogar.

En segundo lugar, las tecnologías que emplean el espectro visible pueden capturar imágenes de alta resolución, lo que facilita la labor de identificar de forma precisa y detallada los objetos de un entorno. Esto es clave para labores como la vigilancia y la inspección de infraestructuras [69].

Por último, una amplia mayoría de algoritmos de visión por ordenador y aprendizaje profundo (Machine Learning) están optimizados para emplear imágenes basadas en el espectro visible. Por tanto, no solo la labor de detección, sino que también la clasificación y seguimiento de objetos es más simple gracias a lo comentado. En adición, la evolución de algoritmos de Machine Learning que emplean redes neuronales convolucionales han conseguido hacer más robusta y fiable la detección de objetos aplicando algoritmos que se basan en el espectro visible [33].

Estas ventajas que ofrece el espectro visible hacen posible la existencia de tecnologías como las cámaras RGB (que permiten detectar y clasificar objetos basados en sus características de color y forma) [54]; también sistemas avanzados de monitoreo de tráfico para gestionar de forma óptima el tráfico y al mismo tiempo mejorar la seguridad vial [70]; y sistemas de seguridad y vigilancia para el control de áreas públicas y privadas [77].

1.2.1 Limitaciones de las tecnologías de detección convencionales

Aunque el espectro visible sea ampliamente implementado en tecnologías de detección de objetos (como se ha visto en esta sección), dichas tecnologías también presentan varias limitaciones que pueden afectar a su efectividad en diversas aplicaciones:

En primer lugar, las tecnologías basadas en el espectro visible son muy sensibles a las condiciones de iluminación: dado que la detección visual de objetos se basa en la luz visible, en entornos de baja o nula iluminación, la capacidad de estas tecnologías para detectar objetos se puede ver significativamente reducida [39]. Esto puede afectar gravemente en aplicaciones donde se pretenda monitorear por razones de seguridad espacios tanto públicos como privados: ya que si no hay condiciones de luz óptima, es más difícil identificar a un intruso o criminal en dicho espacio.

En segundo lugar, otra limitación apreciable es lo muy dependientes que son las tecnologías de detección en condiciones meteorológicas adversas. Esto se debe a que en situaciones de niebla, lluvia intensa o incluso nieve, puede hacer que se degrade la calidad de las imágenes capturadas [49]. Este se debe a que la dispersión y absorción de la luz visible en estas condiciones dificultan la obtención de imágenes claras, lo que puede comprometer la precisión de la detección de objetos.

Además, la distancia de detección a la que las tecnologías de detección pueden

operar es limitada. Esto puede afectar a la resolución de la detección ya que los objetos pueden llegar a aparecer borrosos o incluso indistinguibles [8].

También, otra limitación importante a tener en cuenta es la vulnerabilidad de las tecnologías de detección basadas en el espectro visible frente a obstáculos físicos. Esto es destacable dado que las ondas del espectro visible (que ronda el orden de los nanómetros) no puede penetrar objetos sólidos como vegetación densa o incluso paredes [23].

Por último, dichas tecnologías pueden llegar a capturar imágenes que muestren detalles específicos de lo que capturen. Esto puede plantear serias preocupaciones de privacidad [19].

Dichas limitaciones, aunque son inherentes en las tecnologías de detección de objetos que utilizan el espectro visible, han impulsado la exploración de otras modalidades de detección, como bien puede ser el caso de las tecnologías que emplean ondas milimétricas para la detección de objetos.

1.3 Las ondas milimétricas y sus aplicaciones en la detección de objetos

Las ondas milimétricas son aquellas que se sitúan entre las microondas y las ondas infrarrojas dentro del espectro electromagnético. Estas ondas tienen longitudes de onda que varían entre 1 milímetro y 10 milímetros, lo que corresponde a un rango de frecuencias de entre 30GHz y 300GHz [57]. Algunas de las propiedades de las ondas milimétricas son las siguientes:

Una de ellas es que este tipo de ondas tienen una alta frecuencia. Esto habilita la construcción de imágenes con alta resolución espacial, permitiendo la detección precisa y detallada de objetos pequeños o características finas.

También, dado que las ondas milimétricas disponen de un amplio espectro no licenciado, facilita la implementación de tecnologías de comunicación y detección de última generación.

En adición, aunque si este tipo de ondas no se aplican con una potencia óptima pueden llegar a ser bloqueadas por materiales densos, pueden penetrar a través de niebla, lluvia, humo y otras condiciones de baja o nula visibilidad. Esto hace a las ondas milimétricas ideales para condiciones meteorológicas adversas.

1.4. Comparativa entre las tecnologías de detección que aplican ondas milimétricas y las convencionales

Las comentadas propiedades ofrecen ventajas en el ámbito de la detección de objetos:

Dada la alta frecuencia inherente en este tipo de ondas, se facilita la labor de construcción de imágenes con alta resolución espacial, lo que permite una detección precisa y detallada de objetos pequeños o características finas. Esto se puede observar en un estudio donde se exploraba el uso de este tipo de ondas en sistemas de detección para vehículos autónomos [37]. Estas ondas permitieron la construcción de mapas de alta resolución del entorno del vehículo, mejorando la capacidad de detectar obstáculos y otras amenazas en tiempo real.

Otra ventaja que ofrecen las ondas milimétricas es su capacidad de detección en entornos con una condición luminosa y meteorológica adversa. Un estudio realizado en 2023 [36] explora la implementación de un sensor de ondas milimétricas para la detección de obstáculos en el ámbito de la conducción autónoma.

Luego, aunque como se ha dicho anteriormente, las ondas milimétricas no son capaces de penetrar a través de materiales densos se han realizado estudios con buenos resultados donde se observa que este tipo de ondas es capaz (con una potencia concreta) de atravesar materiales no metálicos como ropa y empaques, lo que es útil para aplicaciones de tanto seguridad como de control. Puede ser el ejemplo un estudio realizado en 2001 [58] donde se implementó un radar de ondas milimétricas para el monitoreo del tráfico vehicular empleando un radar de aproximadamente 77GHz.

1.4 Comparativa entre las tecnologías de detección que aplican ondas milimétricas y las convencionales

Tras haber explorado las características de las principales tecnologías de detección utilizadas en la actualidad (aquellas que emplean ondas del espectro visible y aquellas que emplean ondas milimétricas) y algunas de sus aplicaciones en el mundo real, parece claro que las tecnologías que emplean ondas milimétricas mejoran las labores de aquellas que emplean ondas del espectro visible. Aún así, para poder concluir de forma clara lo dicho, se va a presentar un estudio comparativo en tres escenarios clave y en del día a día. Para ello, se emplearán exploraciones recientes que se hayan realizado en condiciones similares de modo que la comparación proceda de manera correcta.

El primero de los escenarios es la labor que tienen los vehículos autónomos a la hora de detectar obstáculos. Como ya se ha comentado anteriormente, la labor de detección de objetos requiere precisión para que así se pueda aportar seguridad. En el caso de la conducción autónoma es imprescindible que esta labor se realice con la mayor precisión posible ya que si no se hace, podría provocar por ejemplo innumerables accidentes de tráfico. Se ha detallado ya en este trabajo, que aunque ambas tecnologías funcionan correctamente en condiciones de visibilidad perfecta, las ondas milimétricas ofrecen una ventaja frente a las ondas pertenecientes al espectro visible en condiciones de visibilidad prácticamente nulas. Esto se puede apreciar en dos estudios: El primero de ellos realizado en 2021 [61] emplea tecnologías convencionales como el sistema LiDAR, y sus conclusiones revelan precisiones entorno al 50 % cuando las condiciones de visibilidad se reducen. En cambio, en otro estudio realizado en 2022 [72], detalla que implementando sobre los sistemas LiDAR, sensores "mmWave" se alcanzan niveles de precisión entorno al 90 % en las mismas condiciones.

El segundo escenario en donde la labor de detección de objetos es clave, es en el monitoreo de tráfico vehicular. Esta labor es clave para las autoridades de tráfico de cualquier comunidad hoy en día para poder identificar, por ejemplo, a infractores que excedan los límites de velocidad en carretera. Un estudio realizado en 2023 [76], concluye que sin la implementación de sistemas que aporten una visión espacial más precisa (como aquellos sistemas que implementan ondas milimétricas), la labor de seguimiento de vehículos no solamente decae, sino que también hace que estos sistemas necesiten ser un 60 % más complejos (computacionalmente) para llegar a resultados aceptables de precisión. En cambio, otra exploración sobre el mismo escenario realizado en 2024 [28] indaga sobre la oportunidad que ofrecen los nuevos radares de ondas milimétricas 4D que proporcionan no solo más simplicidad en los sistemas de detección sino que se consiguen resultados de precisión de hasta el 98.58 % en condiciones de visibilidad variables.

Por último, el tercer escenario en el que es clave la detección de objetos, es en el ámbito de la seguridad pública, en concreto, en la labor de identificación de objetos ocultos. Una de las propiedades que tienen las ondas milimétricas (como ya se ha mostrado con anterioridad) es su capacidad para atravesar objetos no metálicos. Aprovechando dicha propiedad, un estudio realizado en 2020 [52] encontró la forma de detectar armas metálicas bajo ropajes de individuos empleando sistemas que capturaban imágenes pasivas de ondas milimétricas con una precisión del 95 % en un escenario con un gran flujo de personas y en condiciones irregulares de visibilidad. En cambio, otro estudio realizado en 2023 [59] demuestra como aunque la fusion

del sistema LiDAR con cámaras estereoscópicas para la detección 3D de objetos proporcionase datos precisos acerca del entorno que se estuviese analizando (en este caso una calle con vehículos estacionado), el alto costo y la complejidad de implementar dicho sistema limita la efectividad en la detección de objetos ocultos.

Tras haber explorado tres escenarios en donde la labor de detección en clave para distintos objetivos, se puede concluir como claramente las tecnologías de detección basadas en ondas milimétricas superan a las tecnologías del espectro visible en diversas condiciones adversas, pero sobre todo, en condiciones de visibilidad reducida: la capacidad de las ondas milimétricas para ofrecer mayor precisión y robustez, especialmente en condiciones de visibilidad reducida, así como también a la hora de detectar objetos ocultos, demuestra como las ondas milimétricas representan tanto una opción mucho más óptima, como también ofrecen más oportunidades en la detección de objetos en comparación con los sistemas convencionales basados en el espectro visible.

1.5 Motivación del trabajo

Tras haber concluido en el anterior apartado que las ondas milimétricas ofrecen mejores soluciones en el ámbito de detección de objetos, sobre todo en condiciones de visibilidad reducida, y con el objetivo de aprovechar la gran cantidad de oportunidades que ofrecen este tipo de ondas para la labor de detección, se pretende contribuir al contexto evolutivo del ámbito de la detección mediante el desarrollo de un sistema de detección, tanto de objetos como de personas, que tenga precisión en situaciones donde las tecnologías de detección basadas en el espectro visible no funcionan de manera correcta.

1.6 Objetivos del trabajo

Con el fin de proporcionar a este trabajo una estructura clara y alcanzable que contribuya al avance del conocimiento en este campo se van a establecer los siguientes objetivos principales:

- *Reconocimiento de movimiento de personas sin ningún objeto de por medio:* Para comprobar la efectividad del sistema desarrollado se realizará la detección en dos escenarios de visibilidad distintos: con luz y sin luz. Esto es para comprobar si realmente dicho sistema es más óptimo que uno que utilice tecnologías del espectro visible (las cuales no son óptimas para la detección en condiciones de visibilidad reducida o nula). Para el cumplimiento de dicho objetivo, los resultados tendrán que demostrar que el modelo es capaz de

clasificar e identificar el movimiento que la persona realice con cierto nivel de precisión.

- Reconocimiento de movimiento de personas con una silla de por medio:
Dado que una de las ventajas que ofrecen las ondas milimétricas (aparte de ofrecer mayor precisión en condiciones de visibilidad reducida o nula que las tecnologías basadas en el espectro visible), es su capacidad de atravesar objetos no metálicos. Es por ello que para cumplir este objeto el modelo tendrá que ser capaz de identificar y detectar y clasificar el movimiento que la persona realice con la silla, con cierto nivel de precisión. Este escenario se realizará con luz y sin luz también.
- Reconocimiento de movimiento de personas con una mesa de por medio:
Con este objetivo se pretende conseguir exactamente lo mismo que para los objetivos anteriores pero esta vez empleando como obstáculo un objeto más voluminoso y grande como bien puede ser una mesa. De nuevo, este escenario se realizará con luz y sin luz.
- Diseño de una metodología óptima para la detección de objetos: Este primer objetivo será el primer paso que se deberá tomar en este trabajo para poder abordar la labor de detección de forma correcta. Se considerará que este objetivo se ha logrado con éxito si se consigue lo siguiente:
 - Se considerará un diseño óptimo si la metodología diseñada basada en métodos previamente utilizados en la labor de detección con éxito.
 - Se considerará un diseño óptimo si se consiguen cumplir los tres objetivos principales comentados en este apartados.

Una vez determinados los objetivos principales de este trabajo, se tratará de cumplir los siguientes objetivos secundarios, los cuales serán claves para poder dar cohesión a la investigación y entender las aplicaciones futuras del trabajo:

- Entender la razón por la que la ética debe ser considerada en estas tecnologías:
Como bien se ha indicado anteriormente, pueden surgir dilemas éticos acerca de la privacidad de los datos recabados por las tecnologías de detección (tanto las convencionales como las milimétricas).
- Explorar futuras vías de investigación que puedan surgir a raíz de este trabajo:
Este objetivo ayudará a cerrar el trabajo en su conjunto con cohesión y de una forma clara se indicará los futuros pasos a seguir a raíz de las conclusiones obtenidas en esta investigación.

Capítulo 2

Estado de la cuestión

Tras haber expuesto los objetivos propuestos para este trabajo, en este capítulo se realizará una revisión exhaustiva del estado de la cuestión que esta puesta bajo estudio: la detección de objetos basada en tecnologías que empleen ondas milimétricas. Aquí se explorarán estudios con objetivos similares y se analizarán los elementos clave que influyen tanto en la precisión de los sistemas de detección, como en el tratamiento de los datos. Dado que en este trabajo se tratará de identificar tanto a objetos (de distintas formas y tamaños) como a personas en movimiento en diferentes escenarios de luminosidad, los estudios que se emplearán en esta sección, habrán sido realizados en condiciones similares de luminosidad y habrán sido capaces de identificar objetos de distintos tamaños y formas, en movimiento.

2.1 Exploración de los valores de precisión en investigaciones recientes

En primer lugar, es de vital importancia analizar los niveles de precisión obtenidos en estudios recientes. Esto se debe a que, por un lado, servirá para establecer si los objetivos propuestos para cada uno de los casos de análisis (alcanzar en todos los casos un nivel de precisión de al menos un 70%), y por otro lado también asegurar que el trabajo permita detectar posibles áreas de mejora para así contribuir al desarrollo de este tipo de tecnologías.

Un estudio de INFINEON Technologies [60] que se basó en detectar a personas en movimiento y su localización utilizando datos cardiopulmonares, alcanzó unos niveles de precisión de hasta el 90%. Aunque para conseguir los objetivos propuestos de este trabajo, no se van a emplear datos cardiopulmonares, lo interesante del estudio comentado es como llegaron a alcanzar dichos niveles: mediante

el uso de filtros paso banda se consiguió eliminar el ruido inherente en el espacio en el que se practicó dicho estudio. En adición, otro estudio de la universidad de Auckland, consiguió mejorar los niveles de precisión obtenidos por un sistema de seguimiento de varios objetos/ personas en movimiento desarrollado por Texas Instruments (TI) del 45 % al 65 % [32, 46]. Esto se consiguió sustituyendo el método de reducción de ruido empleado por TI (el cual era un filtro extendido de Kalman) por un filtro denominado "Filtro de Kalman Unscented". Este, proporciona una mejor aproximación de modelos no lineales y permite por tanto reducir los errores de linealización y manejar eficientemente la alta dimensionalidad y complejidad de las mediciones (lo que resulta como se ha visto en un mayor precisión y por tanto, mayor fiabilidad en aplicaciones de seguimiento de objetos). Aún así cabe destacar que todos los estudios mencionados, aunque consigan resultados prometedores de precisión, es cierto que se han realizado mediante la utilización simultánea de varios radares de ondas milimétricas.

Otro estudio realizado conjuntamente por dos universidades en Brasil, propone el uso de múltiples (en concreto cuatro) radares de ondas milimétricas combinados con técnicas de fusión de datos y eliminación de ruido para mejorar la precisión en la detección [48].

En cuanto a las técnicas de fusión de datos, de los métodos probados el que dió mejores resultados tanto de precisión como de sensibilidad, fue el que combinó los datos brutos de todos los radares justo antes de entrenar modelos de Machine Learning. Por otro lado, cabe destacar que la utilización de nuevo (como en [32, 46]) de filtros Kalman, consiguió dar lugar a mejores datos para el entrenamiento de los modelos.

Lo cierto es que, aunque los resultados de dicho estudio obtuvo resultados entorno al 90 % y se cumplieron con éxito los objetivos propuestos por el estudio (detección precisa en entornos cerrados de objetos, animales y personas en movimiento de distintos tamaños), se destaca como limitación del estudio la complejidad logística que implica el tener que montar cuatro radares y todos ellos tomen datos de forma correcta. Es por lo que se propone al final de dicho estudio investigaciones que se focalizen en cumplir los mismos objetivos pero reduciendo la cantidad de sensores.

Los estudios comentados hasta ahora en este capítulo demuestran una clara efectividad en la labor de detección la aplicación de sensores de ondas milimétricas. Aún así, ninguno de los estudios se ha probado bajo condiciones de visibilidad nula. Esto prueba que hoy en día, la detección de objetos aplicando varios sensores en condiciones de visibilidad óptima está siendo exitosa.

Aún así, es cierto que como se ha comentado en la introducción sobre las tec-

nologías convencionales, estas también son capaces de detectar con precisión en entornos donde la visibilidad es buena. Sin embargo, como se ha comprobado en la comparativa entre las tecnologías de detección de ondas milimétricas y las tecnologías de detección convencionales, se observan valores mayores de precisión en los estudios que emplean sensores de ondas milimétricas [72, 28, 59] son no solo más fiables, sino que también aportan más simplicidad y fiabilidad en la labor de detección frente aquellas tecnologías que no aplican sensores basadas en ondas milimétricas [61, 76, 52].

En adición a dichos estudios, la universidad de Ciencias Eléctricas de China presenta un estudio innovador para la detección y clasificación de objetos mediante la fusión de sensores de radar de ondas milimétricas y cámaras basadas en ondas del espectro visible [71]. Aunque más adelante en este capítulo, se profundizará sobre esta metodología (la cual es la más empleada hoy en día para detectar de forma precisa objetos y personas), en dicho estudio se presenta una fusión de datos mediante un proceso de alineación espacial y temporal de los datos recolectados por ambos sensores, lo que permite una correlación precisa entre las detecciones de ambos sensores. Los resultados de dicho estudio muestran hasta un 89.42 % de precisión promedia en escenarios de baja iluminación y baja interferencia magnética. En condiciones similares, el estudio “A Millimeter-Wave Radar-Aided Vision Detection Method for Water Surface Small Object Detection” presenta un enfoque basado de nuevo en la fusión de sensores de radar de ondas milimétricas y cámaras de visión RGB [80]. El sistema propuesto alcanzó una precisión del 87.3 % para la detección de objetos en superficies acuáticas (lo cual es clave para la navegación de vehículos marítimos).

Como se ha especificado sobre los dos estudios mencionados que exploran sobre la detección de objetos en condiciones de visibilidad variables, la precisión encontrada es impresionantemente buena. En dichos estudios se demuestra otra vez que los sensores de ondas milimétricas mejoran las capacidades que ofrece la detección de objetos basada únicamente en la utilización de cámaras de visión (como las RGB o LiDAR, que están basadas en la construcción de imágenes basadas en el espectro visible).

2.1.1 Factores que pueden condicionar los valores de precisión en la labor de detección

La gran mayoría de estudios (o como en todos los estudios presentados en este trabajo) presentan resultados excelentes a la hora de detectar objetos o personas (o incluso también animales pequeños) en condiciones de visibilidad variable.

Gracias a estos estudios se han podido identificar los factores que más pueden

influir a la hora de obtener valores de precisión óptimos y que servirán para poder dar un análisis más preciso a los resultados obtenidos en este trabajo. Dichos factores incluyen interferencias ambientales, la configuración y posicionamiento de los sensores, los algoritmos empleados de procesamiento de datos y las características particulares del objetivo que se quiere identificar:

- **Interferencias Ambientales:** Como se indicó en la introducción de este trabajo, una de las ventajas que dan los radares de ondas milimétricas sobre los que emplean ondas del espectro visible, es su robustez frente a condiciones climáticas distintas, como la lluvia, niebla o polvo. Estudios como el que ha realizado Texas Instruments, destaca que los radares de ondas milimétricas mantienen una alta precisión en las condiciones mencionadas debido a la capacidad de las ondas milimétricas para penetrar a través de interferencias ambientales [35]. En comparación, como se ha descubierto en la comparativa presentada en el capítulo de Introducción, las cámaras ópticas y los sensores LiDAR pueden experimentar una disminución significativa en la precisión bajo condiciones adversas. En adición, se puede concluir que en adición a lo explicado, que tanto las superficies reflectantes como las oclusiones, pueden causar problemas de detección debido a que tanto los reflejos como la concentración de objetos densos, pueden llegar a influir un ruido excesivo en la recopilación de datos de los sensores [46, 80].
- **Configuración y Posicionamiento de Sensores:** Se ha podido comprobar que el uso de múltiples sensores de ondas milimétricas mejora significativamente la precisión de detección al proporcionar datos redundantes y complementar la cobertura del entorno que se pretende analizar. Si bien es cierto que en estudios que indagan sobre la detección de tráfico en sistemas de transporte inteligentes empleando radares hasta 4D colocados en distintos ángulos y posiciones para así poder cubrir de mejor manera más áreas y minimizar las zonas muertas [8], en otro estudio mencionado en este trabajo, sin embargo, aunque se alcanzasen niveles de precisión de mérito, se recomienda la futura investigación de implementar menos radares en la detección de objetos para así poder reducir la complejidad logística que implica el tener que instalar radares de una forma determinada para poder alcanzar niveles de precisión óptimos [46]. En adición a la importancia de como configurar físicamente el hardware, se añade la necesidad intrínseca de entender el radar que se emplee para la detección de objetos. Esto se debe a que sin una correcta alineación y calibración de los sensores, es muy probable que no se alcancen niveles de precisión óptimos. Esto se demuestra en el estudio "Research of Target Detection and Classification Techniques Using Millimeter-Wave Radar and Vision Sensors", en donde tras un proceso de entendimiento de la herramienta MATLAB, una herramienta de calibración de cámara para poder calcular

la matriz intrínseca de la cámara y la matriz de rotación-traslación, lo que aseguró una alineación precisa de los datos de imagen y radar [46].

- **Algoritmos de Procesamiento de Datos:** En múltiples estudios presentados se descubre la efectividad que tiene el procesamiento de datos justo después de su colección. La implementación de técnicas avanzadas de reducción de ruido y filtrado de SNR (relación señal-ruido) es fundamental para mejorar la precisión de detección. Se puede observar como en el estudio que analizaba la detección de personas y objetos pequeños en movimiento, donde aplicaron múltiples módulos de FFT y CFAR para así poder eliminar eficazmente el ruido estático y mejorar la detección de objetos móviles. Además, en el mismo estudio, los umbrales de SNR específicos para diferentes zonas de rango permitieron una separación, dicho de otra forma, una clasificación más precisa de los objetivos [48]. En adición a lo comentado, la fusión de datos provenientes de múltiples sensores mediante el uso de algoritmos avanzados, como el Filtro de Kalman Unscented (UCK), mejora significativamente la precisión del seguimiento de objetos. Este filtro se demuestra como un algoritmo óptimo por delante del Filtro de Kalman Extendido (EKF) al manejar mejor la no linealidad de datos a nivel de decisión, que combina las detecciones de radar y cámara, lo que permite la extracción de resultados más robustos y adaptativos, especialmente en condiciones de alta interferencia [46, 48].
- **Características del Objetivo a Detectar:** La precisión en la detección también puede variar según el tamaño y la forma del objeto que se pretenda detectar. Los objetos más grandes y con formas definidas son más fáciles de detectar y rastrear con alta precisión. En un estudio realizado en el ámbito de la conducción autónoma se demuestra que los radares de ondas milimétricas son efectivos en la detección de objetos grandes y bien definidos, mientras que los objetos pequeños y de formas irregulares pueden representar un desafío mayor [39]. En adición a la morfología de los objetos que se pretendan detectar, hay que tener en cuenta si bien estos objetos se encuentran en movimiento ya que este factor afecta la precisión de detección. Los algoritmos de seguimiento deben ser capaces de diferenciar entre movimientos rápidos y lentos para mantener una alta precisión en la detección de objetos en movimiento. Aunque se debe elegir un algoritmo de análisis óptimo para el caso de estudio, los radares de ondas milimétricas ayudan en la labor de detectar los movimientos ya que son capaces de medir la velocidad (Doppler) de los objetos [59].

Tras este apartado se ha podido concluir que la alta precisión en la detección de objetos utilizando radares de ondas milimétricas se debe a una combinación de factores que incluyen la robustez frente a interferencias ambientales, una configuración y posicionamiento óptimos de los sensores, el uso de algoritmos avanzados

de procesamiento de datos y las características del objetivos. Aún así, de manera recurrente se puede también concluir que si no se encuentra la forma de hacer que estos factores ayuden en el análisis, pueden afectar negativamente a los resultados finales de detección. Es por ello por lo que para el desarrollo del proyecto se tendrán muy en cuenta estos factores para poder acercarse a los mejores resultados en precisión posibles.

2.2 Hardware óptimo para la detección de objetos y personas: Vision-Fusion

En el apartado anterior se han analizado en profundidad los valores de precisión y los factores más relevantes en estudios recientes que pueden afectar a dichos valores. Como se ha comentado en múltiples ocasiones, la gran mayoría de estudios realizados hasta la fecha (o todos los explorados en este trabajo) han basado su metodología en un diseño concreto: la fusión de radares de ondas milimétricas con cámaras ópticas, las cuales son tecnologías que emplean ondas del espectro visible. Es por ello que en este apartado se explorará en profundidad acerca del denominado método "Vision-Fusion".

La detección de objetos mediante fusión de radares de ondas milimétricas y cámaras de visión es una técnica avanzada que combina las capacidades de sensores de radar y cámaras ópticas para identificar, localizar y seguir objetos en escenarios complejos. Gracias a que los radares de ondas milimétricas emplean ondas de radio de alta frecuencia para detectar objetos, y gracias a que las cámaras de visión proporcionan imágenes visuales detalladas del entorno, se permite una detección más robusta y precisa que simplemente empleando cámaras de fusión.

Este concepto tiene surje de la necesidad de mejorar la precisión y fiabilidad de los sistemas de percepción en aplicaciones críticas como la conducción autónoma y los sistemas avanzados de asistencia al conductor (ADAS).

La fusión de radares de ondas milimétricas y cámaras de visión por tanto surgió como una solución natural debido a las características complementarias de estos sensores: mientras que las cámaras de visión proporcionan una alta resolución espacial y son capaces de detectar detalles visuales de forma precisa, los radares de ondas milimétricas son robustos en condiciones adversas como la lluvia, la niebla y oscuridad, y son capaces de medir distancias y velocidades con alta precisión [73].

Hoy en día, la fusión de radares de ondas milimétricas y cámaras de visión se utiliza principalmente (y como ya se ha comentado) en ámbito de la conducción

autónoma y sistemas ADAS debido a la necesidad creciente de mejorar la seguridad y la fiabilidad de estos sistemas en una variedad de condiciones de conducción. Es por ello por lo que se encuentran diversos estudios empleando dicha metodología para vigilancia y control de vehículos [43] o incluso para la detección de peatones [42].

De forma más precisa, el estudio "Handbook of Multisensor Data Fusion: Theory and Practice" permite entender las razones por las que este método es tan empleado hoy en día [44]:

1. **Detección en todas las condiciones:** Mientras que las cámaras dependen de las condiciones de iluminación y visibilidad (como se ha comentado y mostrado en múltiples ocasiones en este trabajo), los radares de ondas milimétricas proporcionan una detección continua y confiable en cualquier condición de visibilidad variable.
2. **Mejora en la percepción de la profundidad:** Los radares de ondas milimétricas pueden medir distancias con alta precisión. Esto complementa la capacidad que tienen las cámaras de visión de percibir la profundidad de los entornos. Esta fusión por lo tanto mejora la estimación de los objetos en el espacio tridimensional.
3. **Detección de materiales no visibles:** Como se ha explicado ya en este trabajo, los radares de ondas milimétricas pueden detectar objetos hechos de materiales que no son fácilmente visibles para las cámaras, como plásticos o vidrios. Lo comentado, por tanto, aumenta la capacidad de detección en entornos complejos.
4. **Reducción de ruido en la recolección de datos:** Gracias a la fusión de datos, se habilita la posibilidad de poder filtrar y procesar la información de manera más eficiente, lo que reduce el ruido. Esto mejora la claridad de los datos de detección.

Estas mejoras se consiguen gracias al proceso que este método emplea, el cual se basa principalmente en 3 módulos: La fase de obtención de datos, la fase de fusión, y la fase de análisis computacional y humano (dicho proceso se puede apreciar en la Figura 2.1, la cual ha sido extraída del mismo "Handbook of Multisensor Data Fusion: Theory and Practice" [42]):

En la fase de obtención de datos el objetivo es realizar el proceso de detección del objeto o entorno que se pretenda analizar por medio de radares. En el ámbito de radares de ondas milimétricas, en función de los objetivos que se pretendan conseguir, distintos radares se tendrán que utilizar en esta fase. Dada la importancia de

los sensores de ondas milimétricas aplicados a la detección de objetos en industrias como la automotriz o como la robótica (como se ha visto en múltiples estudios ya reflejados en este trabajo), el desarrollo de sensores precisos ha sido exponencial (a parte de necesario).

Algunos de los sensores de ondas milimétricas más utilizados se presentan a continuación junto con sus modelos y características claves:

Característica	Detalle
Modelo	AWR1642BOOST
Frecuencia de operación	76-81 GHz
Rango de detección	Hasta 100 metros
Resolución de distancia	4 cm
Ángulo de detección	± 60 grados horizontal, ± 15 gradis vertical
Características Adicionales	Integración de procesador DSP y microcontrolador ARM Cortex-R4F, soporta procesamiento de señales en el chip, interfaces CAN y UART para comunicación
Referencia	Ficha Técnica [6]

Cuadro 2.1: Especificaciones TI AWR1642BOOST

2.2. Hardware óptimo para la detección de objetos y personas: Vision-Fusion

Característica	Detalle
Modelo	BGT60ATR24C AEC Q100
Frecuencia de operación	60 GHz
Rango de detección	Hasta 10 metros
Resolución de distancia	Precisión milimétrica
Ángulo de detección	± 60 grados horizontal
Características Adicionales	Alta integración con antenas en el chip, baja potencia, y alta resolución, apto para aplicaciones automotrices
Referencia	Ficha Técnica [34]

Cuadro 2.2: Especificaciones Infineon BGT60ATR24C

Característica	Detalle
Modelo	MR3003
Frecuencia de operación	77 GHz
Rango de detección	Hasta 250 metros
Resolución de distancia	5 cm
Ángulo de detección	± 90 grados horizontal, ± 10 grados vertical
Características Adicionales	Procesamiento de señal avanzado, interfaz SPI y CAN, diseñado para aplicaciones ADAS
Referencia	Ficha Técnica [50]

Cuadro 2.3: Especificaciones NXP MR3003

Característica	Detalle
Modelo	ADAR1000
Frecuencia de operación	24 GHz y 77 GHz
Rango de detección	Varía según la configuración de la antena
Resolución de distancia	Alta resolución gracias al control de fase
Ángulo de detección	Configurable según diseño de la antena
Características Adicionales	Controlador de matriz de antenas de fase con capacidad para manejar cuatro canales de transmisión y recepción, apto para aplicaciones de radar phased array
Referencia	Ficha Técnica [18]

Cuadro 2.4: Especificaciones Analog Devices ADAR1000

Característica	Detalle
Modelo	ARS 408-21
Frecuencia de operación	77 GHz
Rango de detección	Hasta 250 metros
Resolución de distancia	0.1 metros
Ángulo de detección	± 60 grados horizontal
Características Adicionales	Alta precisión, capacidad de rastrear hasta 256 objetos, adecuado para aplicaciones de conducción autónoma
Referencia	Ficha Técnica [5]

Cuadro 2.5: Especificaciones Continental ARS 408-21

Los radares empleados son utilizados en distintos ámbitos en función de los requisitos y objetivos que se busquen satisfacer en proyectos concretos. Aún así, dado que estos radares aseguran una detección óptima (debido a las características similares que tiene cada uno), se tomarán como referencia para encontrar un radar

2.2. Hardware óptimo para la detección de objetos y personas: Vision-Fusion

óptimo y adecuado para poder desarrollar este trabajo con el fin de conseguir cumplir los objetivos propuestos para el mismo.

En la siguiente fase denominada fase de fusión el principal objetivo es transformar los datos obtenidos por todos los radares implementados en la primera fase con el fin de tener un conjunto de datos lo suficientemente robusto para su posterior implementación en la siguiente fase. En esta fase de fusión, se encuentran a su vez 5 subniveles que dividen y detallan los pasos requeridos para llegar a unos datos robustos:

- Nivel 0: Preprocesamiento de la fuente. Aquí se preacondicionan los datos para corregir sesgos, realizar alineación espacial y temporal, y estandarizar las entradas de datos.
- Nivel 1: Refinamiento de objeto. En este nivel se pretende asociar los datos obtenidos, para estimar la posición, cinemática o atributos de un objeto o entidad.
- Nivel 2: Refinamiento de situación. Esto se conseguiría mediante la agregación de eventos u objetos para realizar análisis relacionales y estimaciones de las relaciones que tiene la situación en estudio, con el contexto del entorno operativo.
- Nivel 3: Evaluación de impacto. Aquí se construiría una proyección de la situación actual para poder realizar una predicción de eventos, la estimación de la intención de amenazas, encontrar vulnerabilidades, entre otros.
- Nivel 4: Refinamiento de procesos. En este último nivel, se evalúa el proceso de fusión para poder proporcionar avisos a los usuarios o al control adaptativo de la fusión para solicitar información adicional de los sensores si se requiriese.

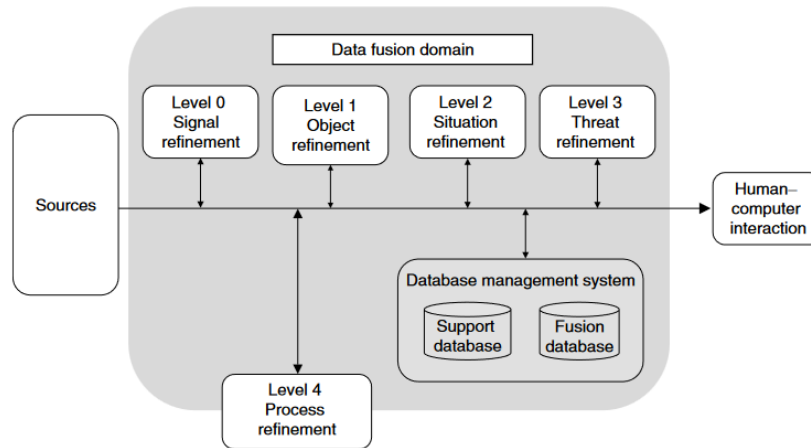


Figura 2.1: Funcionamiento Metodología Vision-Fusion

Tras haber aplicado cada uno de los niveles con éxito, se habría llegado a unos datos que en su conjunto pueden ser considerados robustos y que por tanto pueden aplicarse en la siguiente fase de análisis computacional y humano. En esta fase, se aplicarían los procesos convenientes sobre los datos obtenidos del proceso de fusión (generalmente algoritmos de Machine Learning) para conseguir información que consiga explicar los datos analizados. Tras esto, queda analizar de forma humana los resultados obtenidos tras todo el proceso.

En su conjunto, “este modelo ha sido aceptado internacionalmente” [44]. Aunque este trabajo de fin de grado no vaya a aplicar el método de Vision-Fusion, dicho modelo puede servir de base para diseñar una metodología sencilla y clara de modo que se mantenga la “esencia” principal del modelo mostrado: una fase de detección (que emplee un sensor que se conozca por dar buenos resultados), una fase de procesamiento de datos y una posterior fase de análisis computacional y humano.

2.3 Algoritmos óptimos para el análisis de datos

Tras haber concluido que para poder desarrollar un trabajo de forma y ordenada se tomará el modelo descrito en la Figura 2.1 como base, queda por tanto determinar que algoritmos podrían aplicarse en la fase de análisis computacional y humano. Para ello se explorará en profundidad sobre algunos modelos de Machine Learning que están dando excelentes resultados en el ámbito de detección de objetos con datos extraídos sensores de ondas milimétricas. En concreto, se explorará sobre algoritmos que indaguen sobre la detección de objetos mediante la construcción de los llamados “Point-Clouds”.

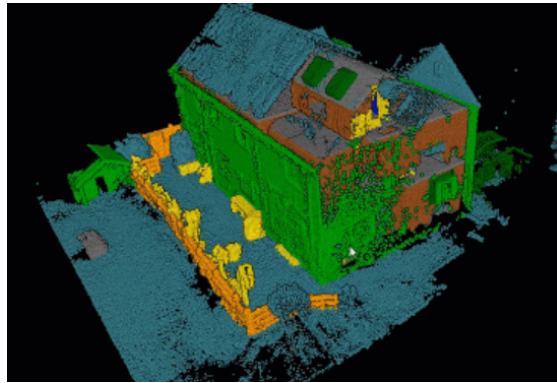


Figura 2.2: ¿Qué son los Point Clouds?

Estos Point-Clouds (o también conocidos como nubes de puntos), son una representación tridimensional de objetos o entornos obtenida mediante la recopilación de puntos en el espacio (veáse Figura 2.2). Cada punto en una nube contiene coordenadas espaciales (x,y,z) e incluso puede llegar a contener más información como colores, intensidades o incluso tiempos de retorno [29].

Durante los últimos años, dada la capacidad que tienen las nubes de puntos para capturar detalles precisos de la geometría de objetos y entorno, se ha investigado el desarrollo de algoritmos que se basen en capturar características concretas empleando nubes de puntos.

Es por ello que, aprovechando la ventaja que dan las nubes de puntos y conociendo la limitación que tienen los sensores de ondas milimétricas para capturar detalles extremadamente precisos (que no es lo mismo que la capacidad de detección), en sistemas de detección de objetos que emplean radares de ondas milimétricas se están implementando algoritmos de deep Learning que sean capaces de construir de forma precisa lo que se quiera detectar.

Las aplicaciones fundamentales en detección de objetos empleando sensores de ondas milimétricas en donde se aplican los Point-Clouds para la labor de detección, residen en el ámbito de la robótica (dado que permiten aumentar la capacidad de los robots de comprender su entorno tridimensional) [26], la realidad aumentada (para la creación de videojuegos y escenarios digitales) [30], en la conducción autónoma (para identificar y evitar obstáculos)[16], en la industria de la construcción y el urbanismo (para permitir una planificación y monitoreo de los proyectos más precisos) [12], y hasta en el ámbito de la arqueología y la conservación del patrimonio (dado que permite documentar y analizar tanto sitios históricos como artefactos) [53].

Una vez introducida la relevancia de los Point-Clouds en el ámbito de la detección de objetos y habiendo comentado como pueden suplir la limitación que tienen los sensores de ondas milimétricas para dar detalles precisos acerca de las características de lo que se esté detectando, se va a proceder a explorar tres algoritmos clave que han revolucionado el procesamiento y análisis de Point-Clouds: PointNet, SECOND y VoxelNet.

Esta exploración no solo permitirá que se vean las características de cada modelo en profundidad, sino que servirá como método comparativo para poder determinar que modelo puede ser más óptimo para intentar cumplir los objetivos que se han marcado en este trabajo.

2.3.1 PointNet

PointNet es un algoritmo innovador que se ha diseñado con el propósito de procesar y analizar PointClouds directamente. Fue desarrollado en 2017 y marcó un cambio significativo en el ámbito de procesar Point-Clouds debido a que dicho algoritmo trata directamente con datos de nubes de puntos, en de transformarlos en representaciones volumétricas o de imágenes, como se ha hecho hasta su desarrollo. Dicho enfoque permite en definitiva un procesamiento mucho más eficiente y una mejor preservación de la información espacial [55].

Desde la introducción de este algoritmo, el mismo ha evolucionado para abordar diversas limitaciones que tiene (se explorarán más adelante) y mejorar su rendimiento. Algun avance notable en el desarrollo de PointNet ha dado lugar al nacimiento de PointNet++, el cual agrupa jerarquías de agrupación y convoluciones puntuales. Esto haría a PointNet++ muy capaz de capturar con mayor precisión las relaciones locales y globales dentro de las nubes de puntos (mientras que PointNet solo se centra en capturar características globales). Aunque parezca prometedor, dicho avance se ha explorado poco en el ámbito de detección de objetos con resultados muy variados y por tanto no se tomará en cuenta para la realización de este trabajo.

PointNet se emplea en una variedad de aplicaciones que incluyen:

- **Clasificación de objetos:** Ya que es capaz de identificar y categorizar objetos 3D como sillas, mesas, autos y otros elementos en modelos tridimensionales.
- **Segmentación de escenas:** Debido a que divide los entornos que se analizan en distintas regiones. Esto permite por ejemplo que en entornos urbanos, se pueda segmentar de forma precisa vehículos, peatones, edificios y hasta otros elementos complejos del entorno.

- **Estimación de pose:** Dada la precisión que se puede obtener en cuanto al alto nivel de información que pueden llegar a ofrecer sus representaciones en 3D, pueden llegar a determinar la orientación y posición exacta de los objetos.

En estos ámbitos, PointNet ha sido aplicado con éxito en múltiples estudios que emplean distintos conjuntos de datos: puede ser el ejemplo de un estudio que empleó el benchmark ModelNet40 y en donde se alcanzó una precisión del 89.2% en la tarea de clasificación de objetos 3D [3]. Otro estudio empleó este modelo con el conjunto de datos ShapeNet, utilizado para segmentación de partes, y se encontraron resultados muy competitivos con una precisión del 83.7% [55].

Estos resultados prometedores se deben a dos motivos a su funcionamiento basado en Deep Learning y que se puede resumir en 5 simples pasos [55]:

- **Paso previo: Asegurar Formato.** Antes de ejecutar la arquitectura es necesario que los datos que se vayan a emplear se presenten como matrices de “n” puntos, donde cada uno de los puntos tiene coordenadas (x,y,z) y opcionalmente otras características (como color o intensidad). Luego, es necesario normalizar y alinear los datos de manera adecuada para garantizar una precisión óptima.
1. **Transformación de puntos:** En primer lugar, el modelo aplica redes de transformación para alinear los puntos y asegurar que las características sean invariables a las rotaciones y traslaciones.
 2. **Extracción de características:** A continuación, se emplean redes neuronales para obtener las características de cada punto (x,y,z). Aquí se aplican múltiples capas de perceptrones multicapa (MLP) los cuales procesan cada punto individualmente.
 3. **Agregación de características:** Posteriormente, utilizando el método de max pooling [68], el cual asegura que la salida sea invariante al orden de los puntos, se agregan las características a nivel global.
 4. **Clasificación/ Segmentación:** En el último paso, se aplican capas completamente conectadas para que de lugar bien sea la clasificación o la segmentación (para este último se combinan las características globales de cada punto con las locales para así poder predecir la etiqueta de cada punto).

Dicho procedimiento no solo habilita la obtención de resultados precisos si no que también da las siguientes ventajas [55]:

- **Eficiencia computacional:** Ya que evita las conversiones que se realizan por ejemplo en representaciones volumétricas. Como resultado, se consigue reducir de forma significativa el coste computacional y de memoria.
- **Simplicidad:** Dado que es una arquitectura sencilla de implementar y de entrenar, no requiere operaciones complicadas de preprocesamiento o conversión de datos.
- **Flexibilidad:** Esta arquitectura es capaz de manejar nubes de puntos de distintos tamaños y resoluciones, lo que la hace adecuada para una amplia variedad de aplicaciones y entornos.

2.3.2 SECOND

Posteriormente, SECOND (Sparsely Embedded Convolutional Detection) fue diseñado en 2018 también para la detección de objetos en nubes de puntos 3D. Esta arquitectura se distingue ya que emplea convoluciones esparcidas para procesar de forma eficiente grandes volúmenes de datos en formato Point-Clouds [75].

El desarrollo de SECOND supuso una evolución significativa en la detección de objetos 3D, ya que mejora la eficiencia y precisión a modelos anteriores (como PointNet). Desde su publicación, se han ido implementado mejoras y se ha ido adaptando para que este modelo sea útil con diferentes sensores y contextos. Esto se ha conseguido gracias a la implementación paulatina de redes neuronales más avanzadas.

Lo que diferencia SECOND frente a otros algoritmos desarrollados anteriormente es el método que implementa de convoluciones esparcidas, el cual permite realizar operaciones convolucionales solo en regiones ocupadas de los voxeles, reduciendo por tanto el costo computacional [75].

Dadas las mejoras que implementa SECOND, se ha incrementado su uso en ámbitos como los siguientes [75]:

- **Conducción autónoma:** Para la identificación y clasificación de obstáculos, vehículos y peatones en el entorno.
- **Robótica:** De nuevo, por su capacidad precisa de detectar obstáculos y la manipulación precisa de objetos.
- **Seguridad y vigilancia:** Donde principalmente se aplica a la detección de intrusos y el monitoreo de áreas.

En los ámbitos mostrados, se han mostrado resultados óptimos en diversos estudios: puede ser un ejemplo una exploración realizada con el benchmark KITTI, en el que se alcanzó una precisión de hasta el 78.62% [75]. Otro estudio que explora sobre ingeniería civil sobre la construcción de puentes, obtuvo un error de solo el 2% [31].

Los resultados óptimos que muestra SECOND en diversas aplicaciones se debe principalmente a su funcionamiento basado en nubes de puntos basados en "voxeles". SECOND requiere que los datos que vaya a emplear se voxelicen. Esto implica que dichos datos deben discretizarse en el espacio 3D en una rejilla regular de voxeles. Los voxeles permiten representar de manera estructurada y de forma eficiente, operaciones convolucionales, al mismo tiempo que facilita el procesamiento de grandes volúmenes de datos.

El funcionamiento de esta arquitectura por tanto puede describirse en los siguientes 5 pasos [75] :

1. **Voxelización:** En este primer paso, la nube de puntos obtenida por el radar se divide en un conjunto de vóxeles.
2. **Codificación de vóxeles:** El segundo paso consta de representar cada voxel obtenido del paso anterior en una característica agregada de los puntos que contiene. Dichas características pueden incluir la media, varianza y otras estadísticas de los puntos dentro de cada voxel.
3. **Convoluciones esparcidas:** El tercer paso consiste en aplicar convoluciones solamente en los voxeles no vacíos. Esto reducirá significativamente el coste computacional del algoritmo. Esto se lograría eventualmente mediante el uso de convoluciones 3D esparcidas las cuales operan eficientemente sobre una estructura de voxeles dispersos.
4. **Extracción de características:** En este paso, se volverán a aplicar capas convolucionales para así poder extraer las características de los voxeles esparcidos (los cuales se han obtenido del anterior paso). Dichas características capturan información de tipo global y local (la cual es necesaria para efectuar con precisión la labor de detección).
5. **Red de detección:** En el último paso, se aplica una red de detección de objetos basada en las características extraídas del anterior paso. Para lograr esto, en este paso se incluyen capas totalmente conectadas y otras técnicas de detección para poder identificar y clasificar objetos que se localicen dentro de la nube de puntos.

Después de haber descrito el funcionamiento, parece intuitivo extraer las ventajas que ortoga el uso de la arquitectura SECOND:

- **Eficiencia:** Mediante el uso de convoluciones esparcidas se consigue mejorar la eficiencia computacional al enfocarse únicamente en regiones ocupadas por puntos. Por ende, se reduce el costo de procesamiento.
- **Escalabilidad:** La capacidad de SECOND para procesar nubes de puntos de alta densidad de manera efectiva, lo hace extremadamente útil en aplicaciones en tiempo real.
- **Precisión:** Como ya se ha especificado anteriormente, SECOND muestra un rendimiento aparentemente superior en comparación con otros métodos de detección basados también en nubes de puntos.

Cabe destacar que aunque este modelo consiga resultados de precisión mejores que los que se obtienen con otros modelos también basados en nubes de puntos, dichos resultados se han obtenido de estudios que en su mayoría se han empleado múltiples sensores para proporcionar vóxeles con mayor información. Lo comentado, debe ser tenido en cuenta dado que en aplicaciones donde solo se emplee un radar, es probable que no sea tan efectivo.

2.3.3 VoxelNet

El último modelo que se explorará en esta sección es VoxelNet. Este modelo fue desarrollado en 2018 también con el objetivo de mejorar la detección de objetos basada en nubes de puntos. Este modelo destaca por integrar la voxelización y el aprendizaje profundo en una sola red. Esto permite una detección precisa y eficiente. En adición, VoxelNet fue pionero al combinar técnicas de voxelización con redes neuronales profundas, lo que habilita una representación rica y detallada de los datos de nubes de puntos.

El enfoque unificado que integra VoxelNet ha inspirado el desarrollo de otros modelos, como puede ser el explorado en el apartado anterior: SECOND, el cual se centra en mejorar la eficiencia que presenta VoxelNet, manteniendo al mismo tiempo una alta precisión [79].

Con el paso del tiempo, VoxelNet ha ido incluyendo mejoras en la codificación de características y la eficiencia computacional, así también como adaptaciones para diferentes aplicaciones y contextos. Algunos de estos en donde se aplica mayoritariamente VoxelNet pueden ser [79]:

- **Conducción autónoma:** Para la detección de obstáculos, peatones y otros vehículos, VoxelNet es eficaz ya que analiza el entorno de los vehículos autónomos y habilita la toma de decisiones, en ámbitos como la navegación, de manera segura.

- **Drones y UAVs:** De nuevo, por su capacidad de detección, esta arquitectura puede emplearse en la navegación y evitación de colisiones de este tipo de vehículos.
- **Logística y almacenes:** VoxelNet puede ayudar a la gestión eficiente de inventarios y la automatización de procesos en entornos industriales.

En algunos estudios se demuestra la efectividad de VoxelNet obteniendo resultados óptimos: en un estudio empleando el mismo benchmark KITTI [75], se obtienen resultados de hasta el 77.47 %. Esto es un resultado peor en comparación con el obtenido por SECOND en el mismo escenario (ver apartado anterior).

Al igual que como se ha descubierto en el apartado anterior (que explora sobre la arquitectura SECOND), VoxelNet también toma como punto de partida la utilización de vóxeles para poder luego aplicar redes neuronales y así llegar a resultados precisos (lo cual tiene sentido ya que SECOND se basó en VoxelNet). Por tanto su funcionamiento también es muy similar al de VoxelNet [79]:

1. **Voxelización:** En este paso se divide la nube de puntos en vóxeles mediante la conversión de los datos de puntos dispersos en una estructura regular que ser procesada mediante convoluciones 3D (tercer paso).
2. **Codificación de vóxeles:** En este paso, se consigue la representación de cada voxel obtenido en el paso anterior, con una característica basada en los puntos que contiene. La codificación de este paso incluye la agregación de información de los puntos dentro de cada voxel, como la media y la varianza de las coordenadas (entre otras características).
3. **Red Convolutiva 3D:** En este paso se aplican capas convolucionales 3D para así poder extraer las características volumétricas de los datos ya procesados por los pasos anteriores. Dichas capas operan sobre la representación voxelizada para capturar así patrones espaciales y características relevantes.
4. **Red de detección:** En este último paso (y como en los dos algoritmos ya explicados en esta sección), se emplea una red de detección para así poder identificar y localizar objetos de forma precisa. Aquí se emplean capas ya conectadas al mismo tiempo que técnicas de anclaje para predecir las posiciones y las clases de los objetos en la escena que se pretenda analizar.

El funcionamiento de VoxelNet proporciona una serie de ventajas como las siguientes:

- **Precisión:** La integración completa de la voxelización y la aplicación de redes convolucionales (deep learning) permite capturar características detalladas y realizar detecciones precisas.

- **Robustez:** Esta arquitectura también permite manejar datos de alta densidad y resolución, lo que la hace óptima para aplicaciones que requieran la monitorización en tiempo real.
- **Versatilidad:** Debido a que VoxelNet puede adaptarse con éxito a distintos tipos de nubes de puntos y requerimientos de detección.

Al igual que con SECOND, la gran mayoría de resultados óptimos que se han obtenido mediante la utilización de VoxelNet, han sido a través de estudios que han empleado varios sensores. De nuevo, es posible que la efectividad de dicho modelo se vea reducida en aplicaciones donde solo se disponga de un radar.

A continuación, tras haber identificado tres posibles algoritmos que puedan ser de utilidad para este trabajo, el Cuadro 2.6 muestra de manera resumida y clara los aspectos fundamentales de cada uno:

Característica	PointNet	VoxelNet	SECOND
Descripción breve	Procesa nubes de puntos directamente	Integra voxelización y aprendizaje profundo (convoluciones 3D unicapa)	Integra voxelización y aprendizaje profundo (convoluciones esparcidas multicapa)
Formato de datos	Matrices de "n" puntos con coordenadas (x,y,z)	Rejilla tridimensional de voxels	Rejilla tridimensional de voxels
Ventajas	-Eficiente -Simple -Flexible	-Alta precisión -Robustez -Versatilidad	-Alta eficiencia -Escalabilidad -Alta precisión
Complejidad Computacional	Baja: procesamiento de puntos individuales	Alta: integración completa de voxelización y convoluciones 3D	Moderada: uso eficiente de convoluciones esparcidas

Cuadro 2.6: Comparativa de algoritmos de deep learning

Tras haber identificado y explorado en profundidad los modelos más destacables de la detección de objetos mediante nubes de puntos, se procederá a discutir en el siguiente capítulo como se realizará el diseño del trabajo. En adición, se identificarán, gracias al estudio que se ha hecho en este capítulo, los recursos a emplear para la exploración de este trabajo como así también se detallarán los detalles técnicos del proceso.

Capítulo 3

Metodología

3.1 Diseño del trabajo

Hasta ahora, gracias a la investigación realizada en este trabajo, se ha podido descubrir que exploraciones funcionan de forma óptima para detectar objetos. Como consecuencia de haber identificado métodos óptimos y que han funcionado con anterioridad con niveles de precisión que superan los propuestos para este trabajo, la metodología que se desarrollará en este estudio empleará algunos de los recursos que han sido de utilidad para así poder alcanzar los mejores resultados posibles.

Una vez dicho lo anterior, con el fin de abordar de la mejor manera posible todos los objetivos principales propuestos para este trabajo se ha desarrollado una metodología basada en la descubierta en la Figura 2.1 y la cual se presenta en la Figura 3.1. En adición a lo comentado, cabe destacar que el trabajo ha sido desarrollado en un periodo de 36 semanas.

Para poder desarrollar el trabajo con orden y que se puedan abordar la logística interna del mismo con precisión, se van a realizar un estudio basado en tres distintas fases que se explicarán a continuación:

1. **Fase 1: Escaner.** En esta parte se incluye el montaje y toda la logística que se empleará para poder montar un escenario que permita la detección de objetos y que se puedan obtener los datos necesarios para aplicar en la siguiente fase de "Procesamiento de datos". Aquí se incluyen actividades adicionales como el proceso de entendimiento y la documentación sobre las herramientas que se emplearán a lo largo del trabajo.

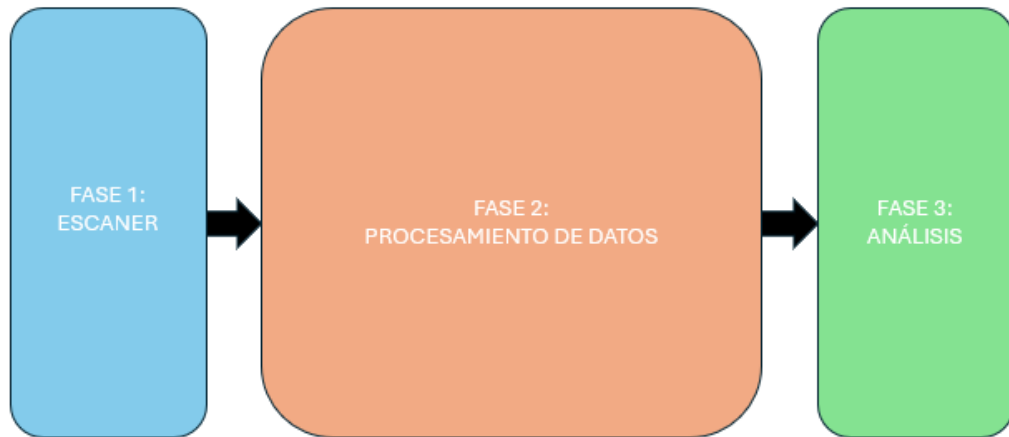


Figura 3.1: Vista holística del trabajo

2. **Fase 2: Procesamiento de datos.** En esta fase del trabajo, se emplearán los recursos hardware y se realizarán las conversiones pertinentes para obtener un conjunto de datos sólidos y representativos que permitan ser analizados con precisión.
3. **Fase 3: Análisis.** En esta fase se empleará un modelo de machine learning que permita procesar los datos resultantes de la anterior de forma precisa y al mismo tiempo que permita determinar cuantitativamente con que precisión se ha realizado la labor de detección en la que se emarcará este trabajo. En otras palabras, a partir de los resultados obtenidos de esta fase se podrá determinar si los objetivos principales propuestos al principio de esta memoria se han cumplido o no.

Esta metodología se ha basado, como se ha dicho anteriormente, en la propuesta en el Manual de fusion de datos [44]. Esto se debe a que dado que dicho modelo se ha empleado como base en otras exploraciones con buenos resultados, partir de dicho modelo parece lo indicado para poder proceder de forma correcta a lo largo de esta investigación.

El procedimiento concreto que se realizará en cada una de las fases indicadas se irá detallando a medida que se avance en esta sección.

3.1.1 Planificación temporal

Tras haber determinado como se desarrollará el trabajo que se propone en esta memoria, ahora se dispone a explicar los plazos propuestos para el desarrollo de este trabajo mediante un diagrama de Gant.

Un diagrama de Gant es una herramienta empleada principalmente para gestionar de modo ordenado la planificación, la coordinación, al mismo tiempo que realizar un seguimiento de las tareas específicas dentro de un proyecto. Este gráfico viene dado por dos ejes: el horizontal y el vertical.

Mientras que en el eje vertical del diagrama se representan las actividades del proyecto, en el eje horizontal se representa el tiempo que se empleará para cada uno de las tareas indicadas en el eje vertical (en este caso, el tiempo viene dado por número de semanas).

En adición, en esta memoria en particular, se indican con colores aquellas actividades que pertenecen a cada una de las fases marcadas siendo el color verde indicando la primera fase de este trabajo; color naranja para las actividades relacionadas con la segunda fase; y color rojo para la tercera y última fase del proyecto.

Por tanto, el diagrama de Gant que refleja los plazos y actividades marcadas para este proyecto se puede visualizar en la figura que se presenta a continuación en la Figura 3.2.

3.2 Fase 1: Escaner

Como bien se ha introducido al principio de este capítulo de Metodología, la primera fase de este proyecto constará de la realización de todo el montaje del modelo físico que permitirá la detección en cada uno de los escenarios que se detallarán en el apartado de “Recogida de datos”.

Para el montaje del escenario que se empleará para realizar la labor de detección, se empleará únicamente un sensor de ondas milimétricas. Esto se debe a que como bien se ha podido comprobar en los capítulos de Introducción y del Estado del arte, la gran mayoría de los estudios mencionados (que abordan la labor de detección de objetos empleando sensores de ondas milimétricas) emplean técnicas de fusión de datos provenientes de distintos radares (Técnica Vision-Fusion) [72, 46, 73, 43, 42] o si no por el contrario emplean varios radares de ondas milimétricas [28, 59]. En adición en el estudio “Human and Small Animal Detection Using Multiple Millimeter-Wave radars and Data Fusion: Enabling Safe Applications” [46], se insta al desarrollo de sistemas de detección de objetos empleando un menor número de radares con el fin de restar complejidad al montaje del escenario y también reducir el coste computacional que implicaría la fusión de datos provenientes de distintos radares. Por tanto, con el fin de seguir contribuyendo

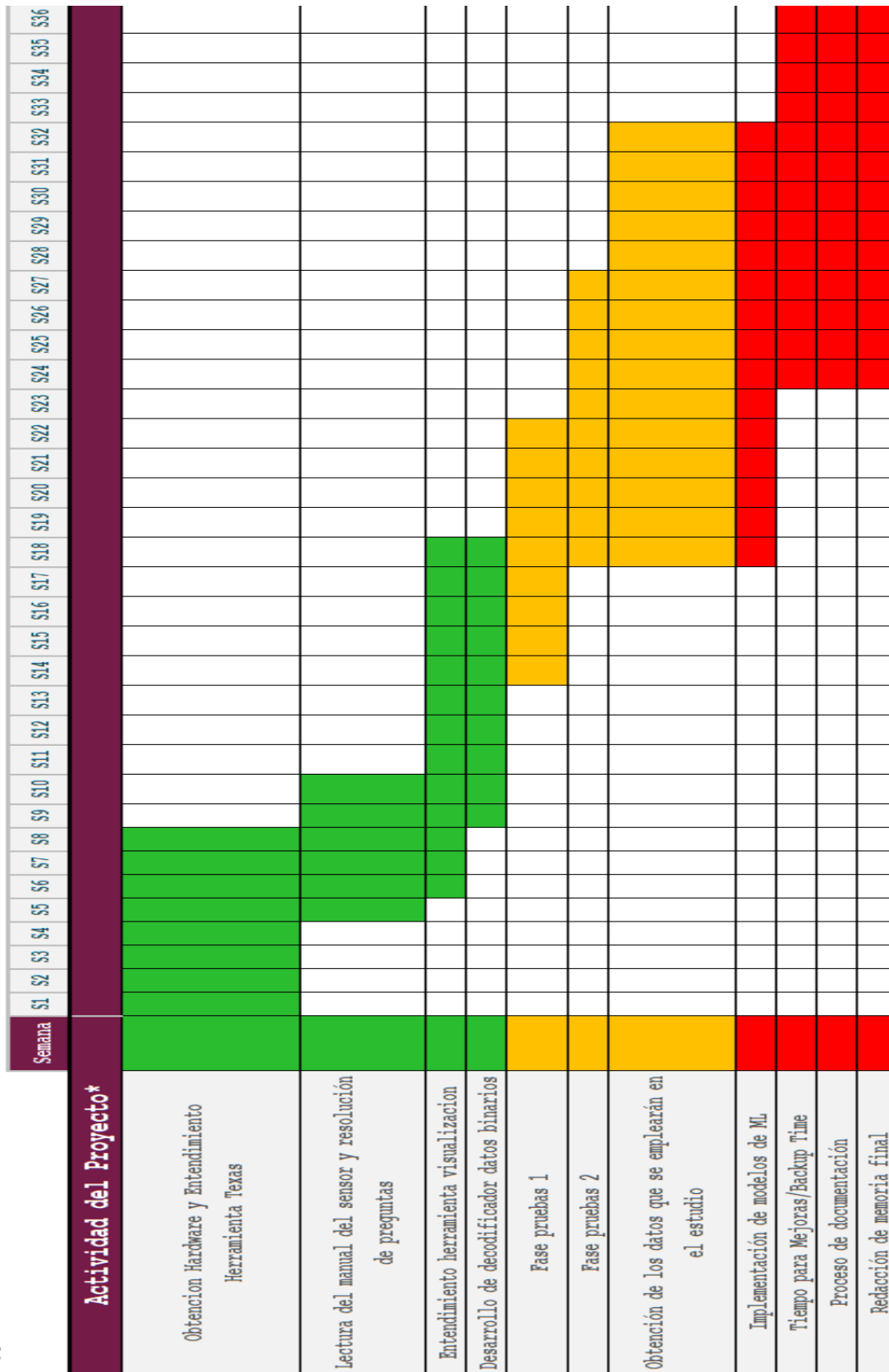


Figura 3.2: Diagrama de Gant del trabajo

a la investigación sobre este tema de forma lógica se empleará simplemente un radar.

Habiendo ya determinado que se va a utilizar simplemente un radar y los motivos, a continuación se explorará sobre el sensor concreto que se empleará: TI IWR6843AOPEVM.

3.2.1 Sensor IT IWR6843AOPEVM

El sensor IWR6843AOPEVM es un radar avanzados de ondas milimétricas desarrollado por Texas Instruments [67]. Dicho radar está basado en tecnología de radar FMCW (Frequency-Modulated Continuous Wave) y opera en el rango de frecuencias de 60-64GHz. Dicho sensor está diseñado para aplicaciones de detección de alta precisión en diversos entornos y condiciones, ofreciendo soluciones robustas y eficientes para la captura de datos tridimensionales.

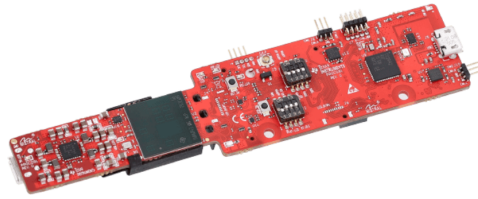


Figura 3.3: Sensor IWR6843AOPEVM

Las características concretas del radar se pueden observar en la siguiente tabla:

Las razones por las que se han elegido este sensor para desarrollar este trabajo son varias:

1. En primer lugar, para poder llevar a cabo la labor de obtención de datos se requiere un sensor preciso. Tal y como se descubrió en el capítulo anterior, hay una serie de sensores que se han probado con éxito en aplicaciones de detección de objetos. Si se observa con atención, el sensor que se propone para este estudio tiene un rango de operación de frecuencias muy similar a los sensores que se exponieron en el capítulo anterior: el sensor propuesto para este trabajo ronda los 60GHz, mientras que los otros sensores expuestos están alrededor de los 70GHz. Esto es fundamental, ya que el hecho de que el sensor que se propone tenga un rango de frecuencias de operación similar

Especificación	Detalle
Frecuencia de operación	60-64GHz
Rango de detección	hasta 100 metros
Resolución de rango	Menos de 4 cm
Ángulo de cobertura	Horizontal: 60°, Vertical: 15°
Velocidad máxima de detección	Hasta 30m/s
Temperatura de operación	-40°C a 85°C
Consumo de energía	1.5W
Procesador integrado	DSP y RTU
Capacidad de múltiples objetos	Sí, detección y seguimiento simultáneo
Resistencia a condiciones ambientales	Alta (poco afectado por condiciones de baja visibilidad)
Aplicaciones típicas	Automoción, robótica, seguridad y vigilancia, industria y automatización

Cuadro 3.1: Especificaciones IWR6843AOPEVM

quiere decir implícitamente que a la hora de realizar el estudio se contará con una resolución parecida. Como los otros sensores se han empleado en aplicaciones parecidas a la que se pretende en este trabajo, se puede por tanto determinar que el sensor es adecuado para este contexto.

2. En segundo lugar, dicho sensor simplemente consume la cantidad de 1.5W. Esto lo hace ideal para aplicaciones portátiles y sencillas como la que se pretenden en este estudio.
3. En tercer lugar, dado que no se contaba con un presupuesto económico para la realización de este proyecto, lo hice ideal para el propósito académico de este estudio.
4. En último lugar, la especificación “EVM” quiere decir “Módulo de evaluación integrado”. Esto implica que el sensor permite la obtención de datos en formato de nube de puntos a través de su puerto micro-USB (ver Figura 3.4). En adición, debido la especificación “EVM” se incluye una herramienta virtual y manejable que permite la representación en vivo de los datos que se capturen durante el proceso de detección.

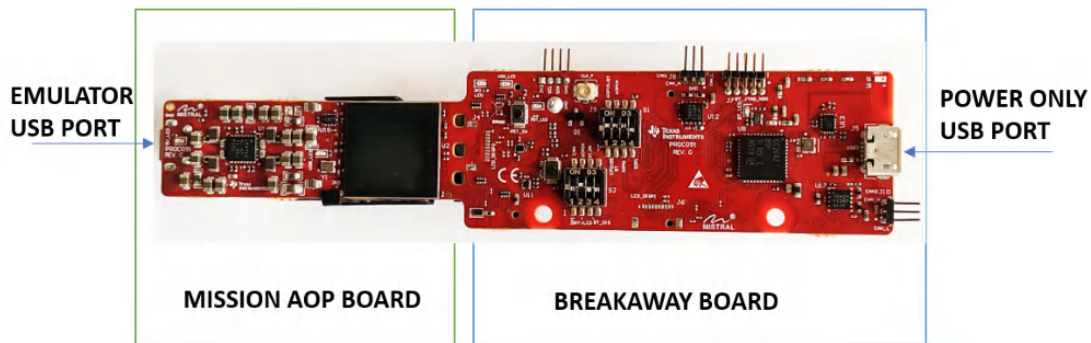


Figura 3.4: Indicaciones puertos USB del sensor

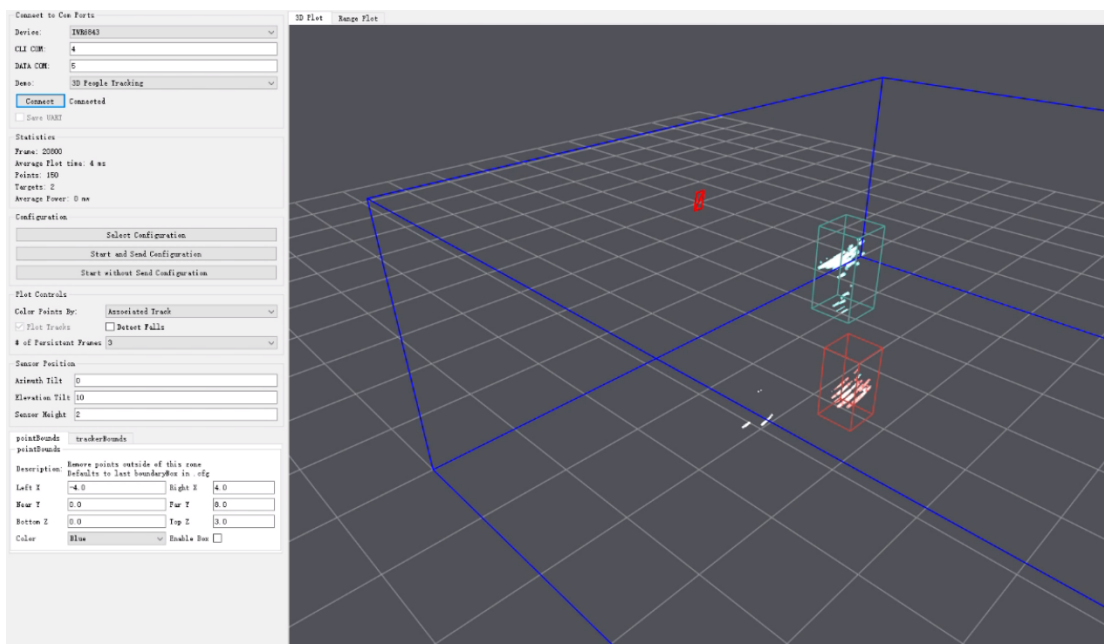
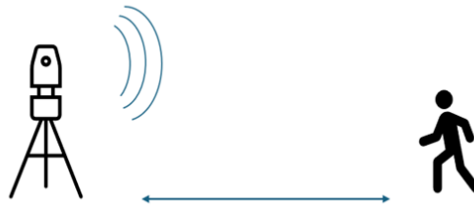


Figura 3.5: Herramienta de visualización

3.2.2 Recogida de datos

Para la recogida de datos, se requiere un proceso que permita el estudio de los objetivos propuestos para este trabajo. Dado que el objetivo de este trabajo es medir la precisión de detección en tres escenarios distintos (detección de personas sin ningún objeto de por medio, detección de personas con una silla de por medio y detección de personas con una mesa de por medio), se proponen los siguientes esquemas para cada uno de los escenarios (ver Figura 3.6).



(a) Sin objeto



(b) Con silla de por medio



(c) Con mesa de por medio

Figura 3.6: Procedimiento físico para la obtención de datos

El procedimiento que se sigue en cada uno de los casos es muy sencillo y se basa principalmente en avanzar unidireccionalmente hacia el radar:

1. Iniciar simulación del radar y dirigirse a zona indicada.
2. Recorrer 4 metros de frente hacia el radar de forma lenta.
3. Detener simulación

Dado que solo se dispone de un radar, para asegurarse de que los resultados finales sean los más robustos posible, se realizará este mismo proceso para cada uno de los escenarios cuatro veces.

Aunque el procedimiento que se va a emplear sea aparentemente sencillo de aplicar,

hay que tener en cuenta que dado el rango de alcance del radar, se van a detectar puntos en situaciones que no se quieren. Un ejemplo que es después de encender el radar, el mismo detectará puntos de la persona dirigiéndose a la zona de inicio para proceder con el desplazamiento de 4 metros. Esto es una fuente inherente de ruido y puede llegar a alterar los resultados finales para mal.

En adición, para los escenarios que involucran la detección de personas a través de objetos (los cuales también se pretenden que sean detectados), es necesario que los objetos se muevan al mismo tiempo que la persona. Esto se debe principalmente a que este sensor no es capaz de detectar objetos estáticos y por tanto se requiere de movimiento para que las reflexiones de las ondas sean lo suficientemente significativas para que se pueda formar un punto en tres dimensiones. Esto puede complicar la labor de detección dado que puede existir la posibilidad de que se confundan puntos entre la persona y los objetos de cada escenario, lo que reduciría la precisión del proyecto.

Tras los últimos comentarios (que se tendrán en cuenta para la parte de análisis de este trabajo), en la siguiente figura se refleja un esquema interno representativo de la Fase 1 de la metodología empleada.

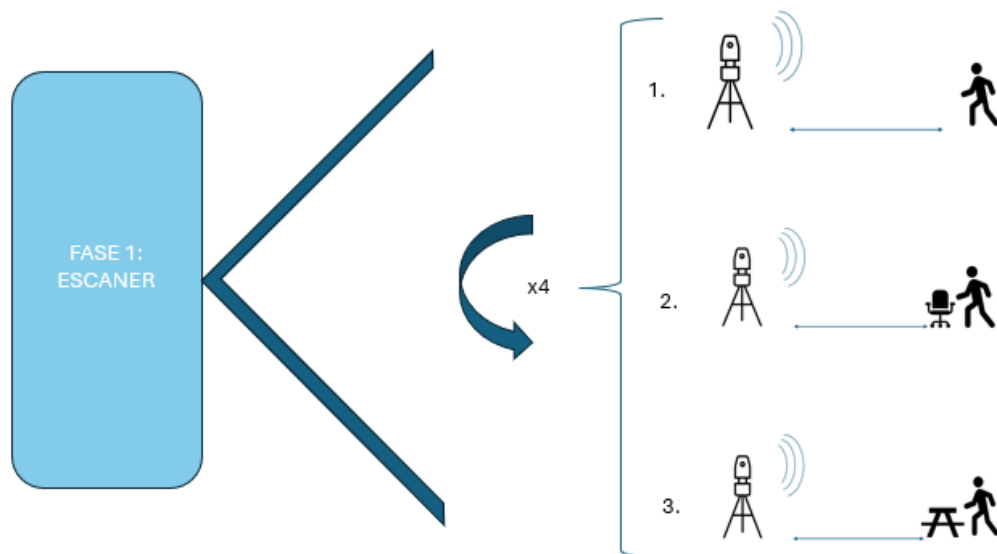


Figura 3.7: Esquema interno de la Fase 1: Escaner

3.3 Fase 2: Procesamiento de datos

En la fase 2 de la metodología propuesta para este trabajo, se tendrá como objetivo realizar las pertinentes transformaciones de datos para conseguir un conjunto robusto y apto para su posterior aplicación al modelo de machine learning elegido para la tercera fase (ver Figura).

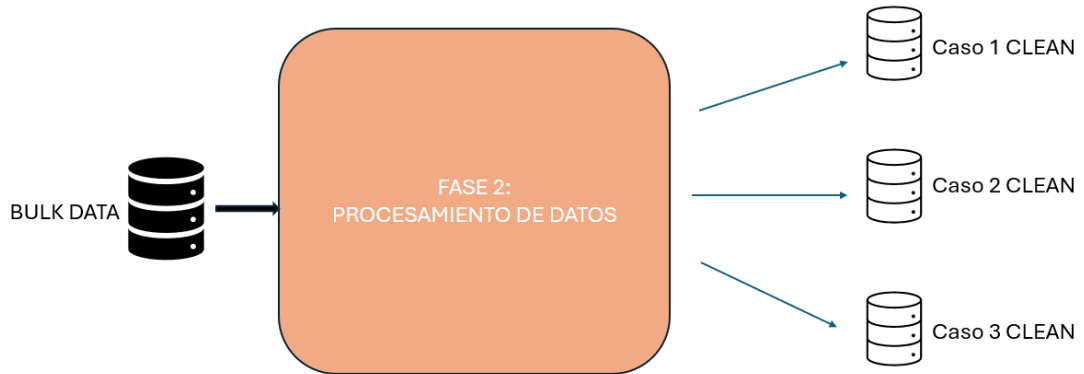


Figura 3.8: Esquema general de la Fase 2

Esta fase de la metodología, se dividirá a su vez en tres niveles que se han colocado en un orden lógico y similar al indicado en la metodología indicada en la Figura 2.1.

3.3.1 Primer procesado: Obtención datos útiles

Tras obtener los datos después de haber realizado la mecánica indicada en el apartado anterior, hay que tener en cuenta la herramienta del sensor empleado en este trabajo. Esto se debe a que dicha herramienta, aunque habilita de forma muy sencilla descargar los archivos que contienen los datos recogidos durante el proceso de detección, dichos datos son procesados en formato “.bin”, lo que indica que están codificados en formato binario. Es por ello que el primer procesado de esta fase consistirá en hacer que los datos obtenidos por el sensor sean legibles y aptos para su manipulación.

Los archivos binarios son conjuntos de datos que son almacenados en una serie de bits, los cuales pueden representar información tal como imágenes, audio, vídeo, ejecutables y otros tipos de datos. La razón por la que dicho formato de

datos no es procesable directamente se debe a que están en un formato codificado. Por tanto, sin un software específico que interprete su estructura, el contenido aparecería como una serie de caracteres incomprensibles (ver Figura 3.9). Visto

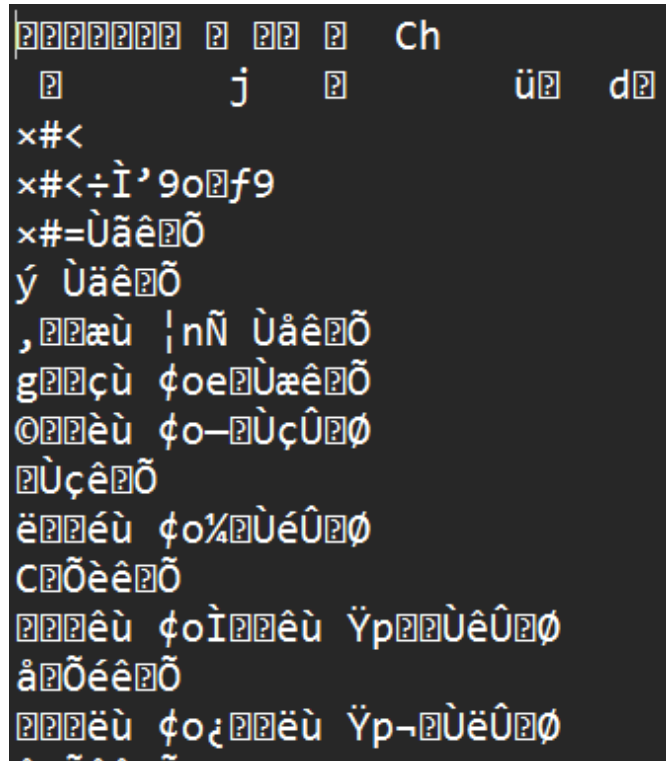


Figura 3.9: Archivo en formato binario

el problema que causan los archivos en formato binario, se ha desarrollado un algoritmo para poder convertir los archivos binarios a archivos “.txt”. El formato “.txt” permite que los datos sean legibles y manipulables de forma sencilla ya que almacenan datos en caracteres ASCII o Unicode (lo que permite que el ser humano sea capaz de interpretar los datos de forma sencilla e intuitiva).

A continuación, el código desarrollado para realizar esta labor se explicará con detalle para que se entiendan los motivos de sus componentes.

Antes de dar paso a la explicación del código desarrollado, es necesario destacar que para desarrollar el mismo de forma sencilla y óptima se han utilizado tres librerías distintas:

- **uartParser:** Este módulo es específico del paquete “gui_parser” y permite parsear datos UART.

- **sys:** Permite acceso a variables y funciones que interactúan con el intérprete Python.
- **os:** Para poder interactuar de forma sencilla con el sistema operativo.

Una vez mencionadas las librerías, cabe destacar que el código se compone de dos partes: una función y el proceso de llamamiento a dicha función.

En primer lugar, la función empleada para la transformación de los datos binarios a un formato legible, se denomina “decode_uart(file_path)”. Esta función admite por parámetro la dirección del archivo que se quiere transformar a formato “.txt”.

A su vez, la función mencionada está dividida en dos partes: la que implica el proceso de lectura del archivo objetivo y luego un bucle “while”. La parte de lectura del archivo objetivo se ha desarrollado como se muestra a continuación:

```
1     with open(file_path, 'rb') as f:
2         parser = uartParser()
3         parser.dataCom = f
4         output_path = file_path + ".txt"
5         with open(output_path, 'w') as output_file:
```

Listing 3.1: Proceso de lectura de archivos

Este proceso se puede describir de la siguiente manera:

- Se abre el archivo “.bin” en modo de lectura binaria (“rb”) para poder leer internamente los archivos resultantes de la fase 1.
- Se define la creación de un nuevo archivo “.txt” mediante la especificación de una ruta que no existiese previamente en la localización donde se quiera poner el nuevo archivo (que será legible)

La segunda parte de esta función se presenta a continuación:

```
1     with open(output_path, 'w') as output_file:
2         while (1):
3             try:
4                 frameData = parser.readAndParseUartDoubleCOMPortBreak
5                 ()
6                 except:
7                     print("Error reading data.")
8                     break
9                 if (len(frameData) > 1):
10                    print(frameData, file=output_file)
11                else:
12                    print("End.")
13                    break
```

```

13     output_file.close()
14     f.close()

```

Listing 3.2: Proceso de descryptación del formato binario

El funcionamiento de esta parte de la función se detalla de la siguiente manera:

- Con el archivo ya abierto, se entra dentro del bucle con el propósito de pegar los resultados en la variable que se ha denominado “output_file”.
- El método “readAndParseUartDoubleCOMPortBreak()” intenta leer y parsear los datos del archivo binario en un formato legible.
- Se implementan otras funcionalidades como la detección de errores en la lectura de los archivos binarios (para que se salga del bucle si hay algún error y no provoque que se corrompa el archivo), o incluso la inclusión dentro de los datos del tamaño de los mismo a modo de guía.

Habiendo explicado los bloques principales que componen la función desarrollada queda explicar como se llama a dicha función en el bloque principal del código desarrollado:

```

1 folder_path = sys.argv[1]
2     for x in os.walk(folder_path):
3         print(x[2])
4         for file in x[2]:
5             if file.endswith(".bin"):
6                 file_path = os.path.join(x[0], file)
7                 decode_uart(file_path)

```

Listing 3.3: Llamamiento a la función de descryptación

El funcionamiento de esta parte se puede resumir de la siguiente manera:

- Un bucle “for” que permite iterar sobre la ruta donde se encuentran todos los archivos resultantes de la fase 1 y así ser más eficiente en el proceso de conversión.
- Una condición que indica que si el archivo termina en “.bin”, debe ser procesado por la función “decode_uart” (que transformará los archivos a formato “.txt”).

Tras la aplicación de este proceso, los nuevos archivos (ya legibles), quedarán listos que puedan ser manejados en los posteriores niveles de esta fase del proyecto de procesamiento de datos. Una visualización esquemática se muestra a continuación:

Para la completa visualización del código desarrollado para esta parte, se proporciona el código completo en el Anexo 7.1.1.

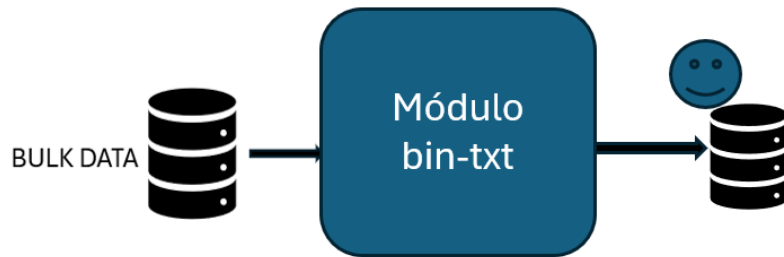


Figura 3.10: Esquema del primer procesado

3.3.2 Segundo procesado: Obtención coordenadas 3D

Tras aplicar el código desarrollado en el apartado anterior, los archivos por tanto ya estarían listos para ser manipulados en las transformaciones correspondientes.

Como se puede apreciar en la Figura 3.11.a, los datos recopilados por el radar vienen descritos como un conjunto de “arrays”, encapsulados por otros “arrays”. Alguno de los elementos claves de dichos datos son:

- Una característica que muestra si ha habido algún tipo de error en la recopilación de datos.
- El número de conjunto de datos recogido durante el proceso de detección.
- El tipo de dato que el radar ha recogido (que como bien se indica son “Point-Clouds”).
- Un array que contiene una serie de características relevantes del proceso de detección.

El último elemento que se ha indicado (el que se refiere al array con una serie de características relevantes del proceso de detección), se refiere a los datos que se emplearán para poder aplicar técnicas de análisis en la tercera fase de este trabajo (ver Figura 3.11.b).

Gracias al manual de indicaciones que llevaba adjunto el sensor empleado en este trabajo [67], se han podido identificar específicamente los datos que explican concretamente las siguientes magnitudes:

- **Primer elemento:** Este elemento se refiere al denominado “elevationvalue”. Dicho valor representa al valor en radianes del ángulo entre el plano horizontal

```
{'error': 0, 'frameNum': 2794, 'pointCloud': array([[ -2.33944270e+00,  6.61351107e+00,  2.10515925e-01,
  6.97200019e-02,  5.11999989e+00,  0.00000000e+00,
  2.55000000e+02],
 [-2.29367822e+00,  6.69638469e+00,  2.12413140e-01,
  6.97200019e-02,  5.35999988e+00,  0.00000000e+00,
  2.55000000e+02],
 [-2.85785164e-01,  3.56469044e+00,  2.60740749e+00,
 -6.97200019e-02,  5.99999987e+00,  0.00000000e+00,
  2.55000000e+02]], 'numDetectedPoints': 3, 'numDetectedTracks': 1, 'trackData': array([[ 1.00000000e+00, -1.93121076e-
00,  6.77698135e+00,
 -2.82603726e-02, -3.52925458e-03, -1.45182665e-03,
  6.73616156e-02, -1.19293528e-03, -6.20605843e-03,
  4.69188169e-02,  3.00000000e+00,  9.98842835e-01,
  2.55000000e+02,  2.54000000e+02,  2.54000000e+02,
  2.54000000e+02]), 'numDetectedHeights': 1, 'heightData': array([[1.
, 0.14292577, 0.14260897]])}
{'error': 0, 'frameNum': 2795, 'pointCloud': array([[ 2.36796957e-01,  1.23126644e+00, -1.63922377e-01,
 -3.40956009e+00,  5.99999987e+00,  0.00000000e+00,
  2.55000000e+02],
 [ 2.34795812e-01,  1.15828413e+00, -2.15059281e-01,
 -3.40956009e+00,  5.99999987e+00,  0.00000000e+00,
  2.55000000e+02]), 'numDetectedPoints': 2, 'numDetectedTracks': 1, 'trackData': array([[ 1.00000000e+00, -1.93638730e-
00,  6.77564669e+00,
 -2.56578680e-02, -1.21022295e-02, -3.27539910e-03,
  7.12474659e-02, -7.75927491e-03, -7.28110410e-03,
  4.96247858e-02,  3.00000000e+00,  9.98842835e-01,
  2.55000000e+02,  2.54000000e+02,  2.54000000e+02,
  2.54000000e+02]), 'numDetectedHeights': 1, 'heightData': array([[1.
, 0.15379302, 0.15349206]]), 'trackIndexes':
array([ 1.,  1., 254.])}
{'error': 0, 'frameNum': 2796, 'pointCloud': array([], shape=(0, 7), dtype=float64), 'numDetectedTracks': 1, 'trackData':
array([[ 1.00000000e+00, -1.93638730e+00,  6.77564669e+00,
 -2.56578680e-02, -1.21022295e-02, -3.27539910e-03,
  7.12474659e-02, -7.75927491e-03, -7.28110410e-03,
  4.96247858e-02,  3.00000000e+00,  9.98842835e-01,
  2.55000000e+02,  2.54000000e+02,  2.54000000e+02,
  2.54000000e+02]), 'numDetectedHeights': 1, 'heightData': array([[1.
, 0.15379302, 0.15349206]]), 'trackIndexes':
array([255., 255.])}
{'error': 0, 'frameNum': 2797, 'pointCloud': array([[ -2.20671343e+00,  6.65897414e+00,  2.10515925e-01,
 -6.97200019e-02,  5.63999987e+00,  0.00000000e+00,
```

(a) Estructura general de los datos obtenidos

```
{'error': 0, 'frameNum': 1, 'pointCloud': array([[ 8.21073418e-01,  2.63733542e-01, -8.36232480e-01,
 -2.78320007e-01,  9.35999979e+00,  0.00000000e+00,
  2.55000000e+02],
 [-8.53553322e-01,  5.84945657e-01, -1.27750599e+00,
  6.97200019e-02,  5.99999987e+00,  0.00000000e+00,
  2.55000000e+02],
 [-8.86262482e-01,  6.07361457e-01, -1.32646151e+00,
  6.97200019e-02,  6.11999986e+00,  0.00000000e+00,
  2.55000000e+02],
 [-2.08007816e+00,  6.72432738e+00,  7.77395057e-01,
 -6.97200019e-02,  4.99999989e+00,  0.00000000e+00,
  2.55000000e+02],
 [-1.92768276e+00,  6.96522542e+00,  7.98198046e-01,
 -6.97200019e-02,  5.99999987e+00,  0.00000000e+00,
  2.55000000e+02],
 [-1.83340696e+00,  6.89194939e+00,  4.28412855e-01,
 -6.97200019e-02,  1.21199997e+01,  0.00000000e+00,
  2.55000000e+02],
 [-1.84843447e+00,  6.94843921e+00,  5.04130571e-01,
 -6.97200019e-02,  1.21199997e+01,  0.00000000e+00,
  2.55000000e+02],
 [ 3.86062915e-02,  3.86050054e+00,  2.52940455e+00,
  6.97200019e-02,  6.51999985e+00,  0.00000000e+00,
  2.55000000e+02],
 [ 7.82667771e-02,  3.91281715e+00,  2.56406707e+00,
  6.97200019e-02,  6.23999986e+00,  0.00000000e+00,
  2.55000000e+02]], 'numDetectedPoints': 9}
```

(b) Datos útiles

Figura 3.11: Datos legibles resultantes del primer procesado

(xy) y la línea de visión hacia el punto en cuestión. Normalmente oscila entre $[-\frac{\pi}{2}, \frac{\pi}{2}]$ radianes.

- **Segundo elemento:** Este elemento se refiere al denominado “azimuthvalue”. Este valor representa al valor en radianes del ángulo medido en el plano horizontal desde una dirección de referencia (normalmente el eje x positivo), hacia la línea de visión del punto en cuestión. Normalmente oscila entre $[0, \frac{\pi}{2}]$

radianes.

- **Cuarto elemento:** Este elemento se refiere al denominado “rangevalue”. Este valor representa al valor en metros de la distancia desde el origen hasta el punto en cuestión.
- **Otra información relevante:** El resto de los elementos que conforman el array dan valores como indicaciones de la relación señal a ruido (SNR) y información doppler para calcular velocidades entre otras. Estas características sirven principalmente para poder realizar transformaciones complejas de los datos obtenidos para así poder obtener valores más precisos. Aún así, dado que estos parámetros solo se utilizan en metodologías que implementan técnicas SAR [9] que emplean por normal general varios radares, no se tendrán en cuenta para el estudio. Esto implica que no se tendrán en cuenta a lo largo de este trabajo consideraciones de ruido (SNR) y tampoco magnitudes derivadas de la magnitud Doppler (como la velocidad). Esta consideración puede influir de forma negativa en los resultados dado que es posible que el ruido se mezcle entre los puntos obtenidos y que por tanto no puedan ser representados de forma correcta.

Con el propósito de obtener los datos requeridos en tres dimensiones, es necesario realizar una serie de transformaciones trigonométricas. Esto se debe a que como bien se puede observar, los valores de “elevation”, “azimuth” y “range” conforman un conjunto de coordenadas esféricas (θ, γ, d). La representación gráfica de dichas magnitudes se puede reflejar en la siguiente figura:

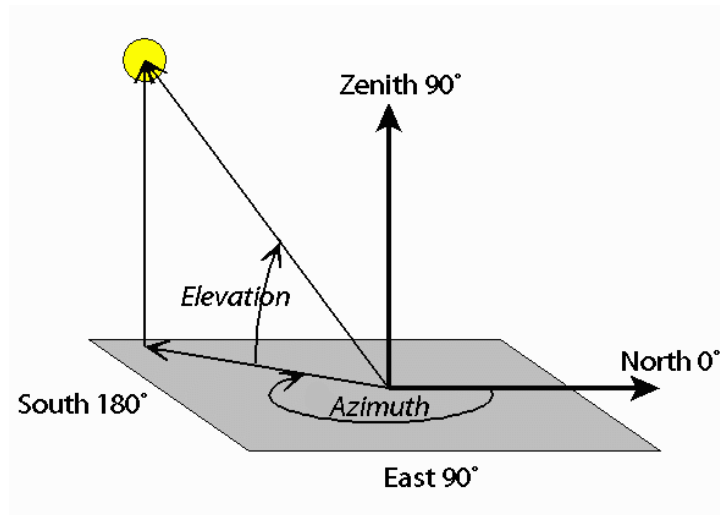


Figura 3.12: Representación gráfica de las magnitudes del sensor

Una vez sabiendo que las coordenadas vienen dadas en esféricas, para convertirlas en cartesianas se aplicarán las siguientes ecuaciones básicas de trigonometría:

$$X = \text{dist} \times \cos(\text{elev}) \times \sin(\text{azim})$$

$$Y = \text{dist} \times \cos(\text{elev}) \times \cos(\text{azim})$$

$$Z = \text{dist} \times \sin(\text{elev})$$

Una vez determinadas las ecuaciones que se emplearán para la transformación de las coordenadas esféricas a cartesianas, queda implementarlo con código de la forma que se indica a continuación:

```

1 for i in range(len(points)):
2     if not isinstance(points[i], list):
3         continue
4     elev, azim, dist = points[i][0], points[i][1], points[i]
5     if not isinstance(elev, float) or not isinstance(azim,
6     float) or not isinstance(dist, float):
7         continue
8     x = dist * (math.cos(elev) * math.sin(azim))
9     y = dist * (math.cos(elev) * math.cos(azim))
10    z = dist * math.sin(elev)
11    temp2 = [x, y, z]
    coordinates.append(temp2)

```

Listing 3.4: Transformación de coordenadas

Dicho código se puede explicar de la siguiente manera:

- Un bucle que recorre todos los arrays e identifica y almacena los valores de “elevation”, “azimuth” y “range” en variables.
- Se aplican las conversiones de coordenadas indicadas en las ecuaciones mostradas anteriormente.
- Se guardan los valores transformados de las coordenadas y se sustituyen por los valores antiguos.

Tras haber implementado la transformación los datos resultantes del primer procesado, la figura que se muestra a continuación muestra de forma esquemática lo que se ha realizado:



Figura 3.13: Esquema segundo procesado

3.3.3 Tercer procesado: Fusión de muestras obtenidas y limpieza de datos

Tras haber realizado la conversión de coordenadas esféricas a coordenadas cartesianas en tres dimensiones, el último procesado de la fase 2 de la metodología propuesta para este trabajo consta de la labor de fusión y limpieza de datos. Estas dos labores se aplicarán de forma simultánea para reducir el coste computacional del trabajo.

De este modo se conseguirá tener un conjunto de datos robusto y óptimo para la posterior aplicación del algoritmos de análisis (ver Figura 3.13).



Figura 3.14: Esquema tercer procesado

En primer lugar, la técnica de fusión se encargará de parsear en un solo archivo de datos las cuatro detecciones realizadas para cada uno de los casos explicados en la sección anterior. Para que esto sea posible se aplicarán una serie de estimaciones

estadísticas para hacer que el conjunto de datos final sea suficientemente representativo. Luego se empleará una limpieza exhaustiva de datos eliminando los casos “inservibles” para la labor de detección. Gracias a la limpieza de datos, los mismos quedarán listos para su análisis en la siguiente fase de esta metodología.

Para que el objetivo de obtener tanto unos datos representativos como limpios y listos para que la labor de análisis sea correcta, se ha desarrollado un código el cual se puede ver completo en los Anexos 6.1.2 y 6.1.3. Para poder entender de forma clara dicho código, se va a disponer de explicar las partes más importantes del mismo.

En primer lugar, antes de proceder a fusionar los datos, se procederá a limpiar los mismos de modo que cada uno de los archivos obtenidos de la fase de escaneo y obtención de datos, sean lo más óptimos posibles.

Para cumplir lo dicho se ha desarrollado una función denominada “preprocess(data)” que coge por argumento los datos legibles obtenidos gracias al primer procesado. Dicha función recorrerá línea a línea dichos datos. El algoritmo empleado en la función se muestra a continuación:

```
1 while i < len(data):
2     if skip == True and data[i] != '':
3         i += 1
4         continue
5     if skip == True and data[i] == '':
6         to_return += data[i]
7         skip = False
8         i += 1
9         continue
10    if data[i: i + 5] == "array":
11        i += 6
12        to_return += "\""
13    if data[i: i + 22] == ")", 'numDetectedPoints':
14        i += 24
15        to_return += "\", \"numDetectedPoints\": "
16        to_return += data[i]
17        while (data[i + 1] != "," and data[i + 1] != "}"):
18            to_return += data[i + 1]
19            i += 1
20        skip = True
21        continue
22    if data[i] == "'":
23        to_return += "\""
24        i += 1
25        continue
26    if data[i] == "\n":
27        i += 1
28        continue
```



```

1 def parse_all_files_in_folder(folder_path):
2     data_frames = []
3     for root, _, files in os.walk(folder_path):
4         for file_name in files:
5             if file_name.endswith('.txt'):
6                 file_path = os.path.join(root, file_name)
7                 df = parse_data(file_path)
8                 data_frames.append(df)
9     combined_df = pd.concat(data_frames, ignore_index=True)
10
11     return combined_df

```

Listing 3.6: Transformación de coordenadas

El objetivo de esta función es procesar cada archivo de una carpeta y combinar los resultados en un solo DataFrame de pandas:

- Para la ruta proporcionada que la función admite como parámetro, se emplea un bucle que procesa simplemente los archivos con la extensión “.txt”.
- Se construye una ruta que será la que se empleará para poder parsear los datos con la función “parse_data”. Dicha función se refleja completa en el Anexo 6.1.2 y esta incluye las transformaciones de datos de coordenadas esféricas a coordenadas cartesianas.
- Se finaliza la función concatenando todos los DataFrames en un solo archivo.

Dicha función tras aplicarla obtiene resultados como el que se muestra en la siguiente figura: Dicho proceso se aplicará a cada uno de los casos obteniendo como

error	frameNum	pointCloud	numDetectedPoints	coordinates	
0	0	1	[[[-2.34130753e-01, 7.84586885e-01, -3.3656180...	106	[[[0.14355405987465764, 0.1437871736612645, -0....
1	0	2	[[[-2.81729017e+00, 2.90541641e+00, -1.2518975...	108	[[[-0.10805597516600786, 0.44898407224443065, -...
2	0	4	[[[-1.85463863e+00, 1.19084983e+00, -1.9890123...	74	[[[-0.07238397026624276, -0.028906588049456217,...
3	0	6	[[[-1.41392861e+00, 3.99711971e+00, 2.2614751...	85	[[[0.008222566079162243, 0.0071432111094662, 0....
4	0	7	[[[-6.44181272e+00, 2.65393749e+00, -1.2677979...	51	[[[0.3541854199857154, -0.6677956380532489, -0....
...
1406	0	594	[[[1.29498064e-01, 1.17250306e+00, 2.2686675...	1	[[[-1.3991246840208085, -0.588727670209909, -0....
1407	0	595	[[[-2.10754489e+00, 6.81311985e+00, 4.2841285...	2	[[[0.01802072483922613, 0.030761061931030738, 0....
1408	0	596	[[[-1.78242354e+00, 6.98053643e+00, 2.1620007...	2	[[[-0.009404711911895514, -0.011225883275034317...
1409	0	598	[[[-1.97353456e+00, 6.86317133e+00, 2.1430285...	2	[[[0.014974974379286025, 0.022857375108901706, ...
1410	0	599	[[[1.76851298e-01, 1.17015282e+00, 4.4545078...	2	[[[-2.8383059831830635, -1.2021700666588058, -0....

Figura 3.17: Concatenación del archivo localizado en “./binData-1/Dark/People_moving_with_chair”

resultado simplemente 6 archivos distintos, los cuales se dividirán en 2 archivos

referentes a cada situación de luminosidad por cada uno de los 3 casos que se han estudiado.

Continuando, para asegurar la robusted en los datos hay que asegurarse de que cada uno de los frames (que son los que contienen los puntos que reflejan detecciones) tenga un número concreto en cada uno de ellos.

Para conseguir esto se va a determinar un “threshold” que indicará el número de puntos que deberá tener cada frame.

Para determinar esto se realizará una estimación muy sencilla (y de manera visual), por medio de un histograma el cual representará los puntos detectados por cada frame en cada uno de los archivos ya concatenados mediante la función que se ha explicado anteriormente.

La estimación del “threshold” se llevará a cabo estimando de manera visual donde puede encontrarse la gran mayoría de frames (encontrar la barra media).

Los 6 histogramas que representan la distribución del número de puntos de cada uno de los casos que se han propuesto para analizar en este trabajo, se pueden observar en la Figura 3.16. Ahora, con la simple observación de los histogramas mostrados en la Figura 3.16 se pueden determinar los siguientes “threshold”:

- **a)**: 120
- **b)**: 100
- **c)**: 125
- **d)**: 130
- **e)**: 150
- **f)**: 150

Una vez determinado dichos umbrales, es lógico pensar que no todos los frames creados por el sensor hayan detectado exactamente los valores marcados en los umbrales anteriores. Es por ello que hay que tener en cuenta tres casos: los frames que estén por debajo del umbral, los que tengan exactamente el umbral indicado y los que superen el umbral.

Para poder tratar cada caso de forma particular se ha diseñado el siguiente código:

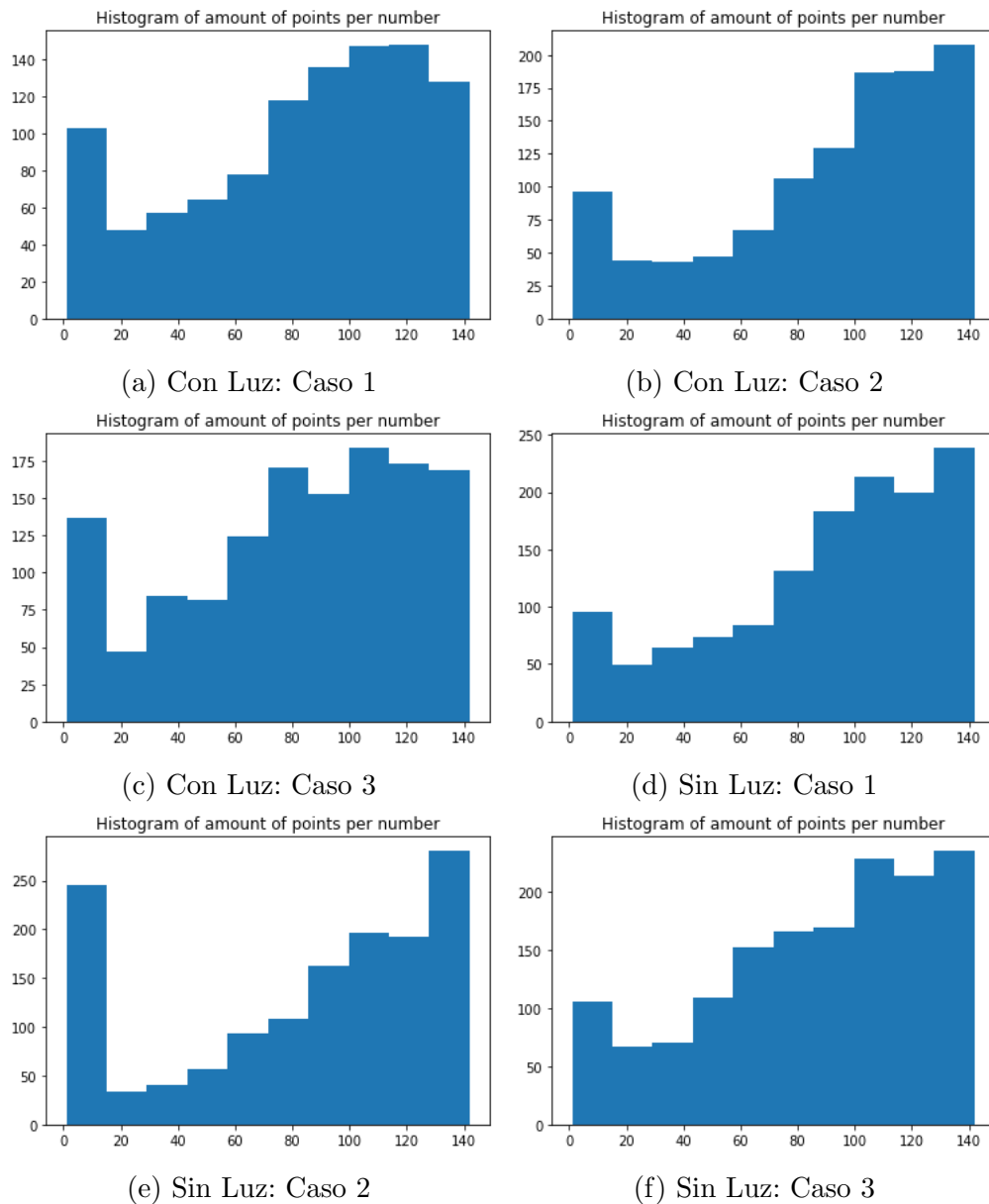


Figura 3.18: Histogramas de la distribución del número de puntos detectados por archivo

```

1
2 for i in range(num_point_clouds):
3     num_rows = len(sample_data.loc[i, "coordinates"])
4     if threshold - num_rows > 0:
5         # Duplicate points
6         sampling_indices = np.random.choice(num_rows, threshold -

```

```

num_rows)
7   new_points = sample_data.loc[i, "coordinates"][
sampling_indices, :]
8   pointnet_dataset.append(np.concatenate((sample_data.loc[i, "
coordinates"], new_points),axis=0))
9   elif threshold - num_rows < 0:
10  # sample points
11  sampling_indices = np.random.choice(num_rows, threshold)
12  pointnet_dataset.append(sample_data.loc[i, "coordinates"[
sampling_indices, :])
13  else:
14  pointnet_dataset.append(sample_data.loc[i, "coordinates"])

```

Listing 3.7: Algoritmo para procesar niveles del umbral

La explicación del método empleado para cada uno de los casos se detalla a continuación:

- **Por encima del umbral:** Esto indicaría que hay más puntos de los necesarios. Para este caso se seleccionarán al azar por medio del método “random”, “threshold” índices de los puntos existentes.
- **Por debajo del umbral:** Esto indicaría que menos puntos de los necesarios. Para este caso se propone duplicar los puntos de manera aleatoria. Este proceso se llevaría a cabo mediante la selección aleatoria de índices de cada frame.
- **Igual que el umbral:** Aquí no habrá que modificar ningún dato dado que se cumpliría con el umbral marcado.

Tras aplicar dicho algoritmo, se separarán las coordenadas en una nueva variable que será la que se empleará durante la fase 3 de este trabajo.

Tras haber aplicado los tres procesados de datos propuestos en este trabajo, a continuación se proporciona un diagrama simple que muestra el proceso realizado durante toda la fase 2 de la metodología propuesta:

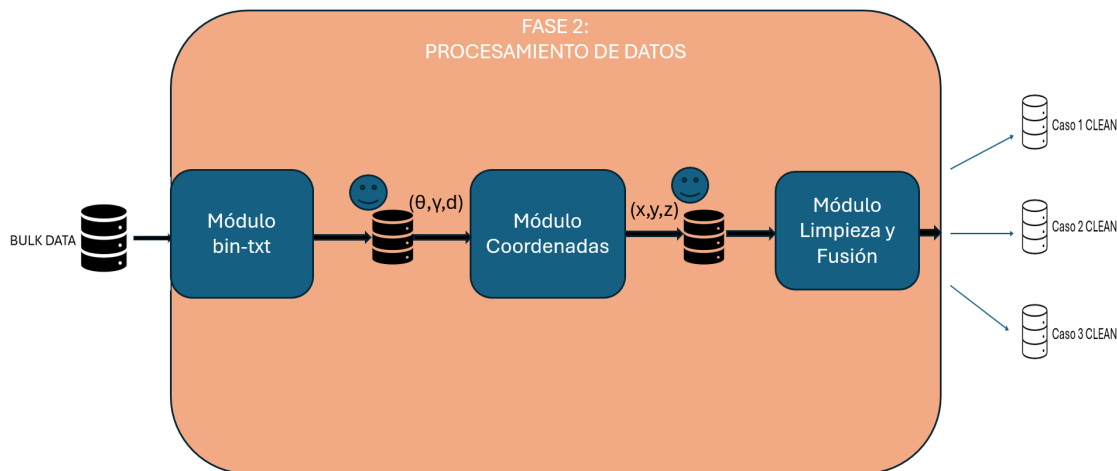


Figura 3.19: Esquema completo de la fase 2

3.4 Fase 3: Análisis

Tras haber superado la fase 2, ya se puede considerar que se tienen los archivos más robustos y representativos posibles para proceder con la fase 3 de la metodología propuesta para este trabajo: Análisis.

En esta fase se utilizarán los datos obtenidos tras el procesamiento de datos para implementar un modelo de deep learning, en concreto, PointNet.

La elección de dicho algoritmo se debe a diversos motivos: uno de ellos es su buen rendimiento en la labor de mapear nubes de puntos. Como se descubrió en el capítulo anterior, PointNet ha demostrado efectividad a la hora de obtener buenos resultados de precisión [3, 55]. Otros algoritmos explorados como SECOND o VoxelNet mostraban incluso mejores resultados que PointNet, pero como bien se comentó entonces, estos algoritmos son normalmente utilizados en estudios donde se emplean varios radares [75, 79]. Dado que en este trabajo se ha empleado un sólo radar, y como no queda totalmente demostrada la efectividad de tanto SECOND o VoxelNet en condiciones como las que se han empleado en este trabajo, PointNet parece la mejor solución.

PointNet es bastante más sencillo de implementar que sus competidores ya que esta arquitectura procesa directamente las nubes de puntos que recibe como entrada (y no como VoxelNet o SECOND, que requieren un proceso de voxelización de

las nubes de puntos, lo que los hace más costosos computacionalmente). Esto hace que se reduzca el coste computacional de la implementación. En adición, PointNet presenta una ventaja fundamental para el manejo de los Point-Clouds: el modelo es invariante a las permutaciones de los puntos obtenidos por el radar. Esto es fundamental para una evaluación precisa de los datos, dado que hay datos que es posible que se hayan detectado más tarde por el sensor y si no se manejan con efectividad podría dar lugar a problemas de sobreajuste de los datos.

Una vez explicadas las razones por las que se ha decidido implementar PointNet, se procederá con el desarrollo del análisis e implementación del algoritmo de PointNet. El desarrollo que se llevará a cabo durante esta tercera y última fase de la metodología propuesta para este trabajo, se puede esquematizar como se representa en la siguiente figura:

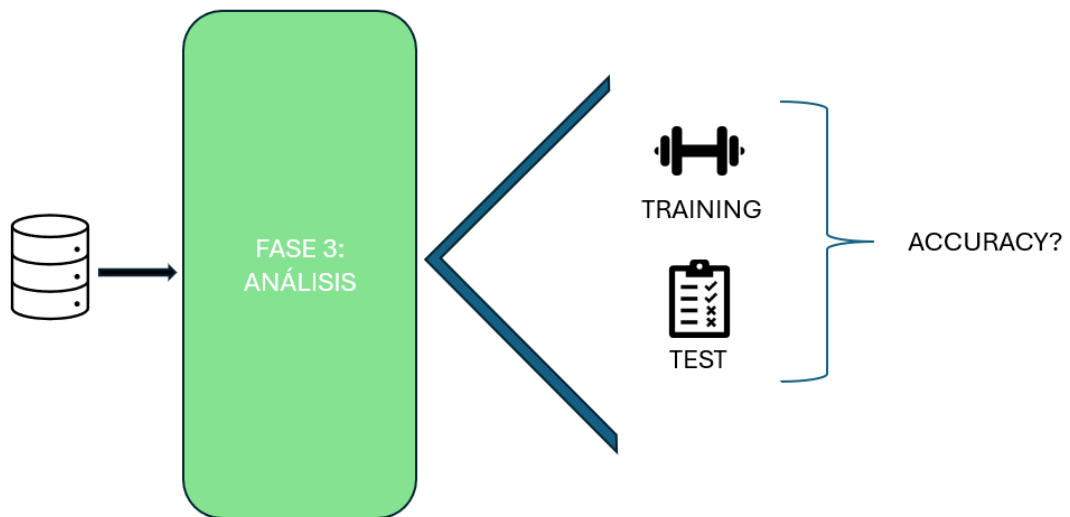


Figura 3.20: Esquema de la fase 3

3.4.1 Software: Aplicación arquitectura PointNet

Como ya se ha justificado anteriormente, PointNet será el algoritmo empleado para que se empleará para realizar la taera de análisis propuesta para este trabajo.

Gracias al procesamiento de datos, los datos ya están lo suficientemente limpios y son representativos para empezar la albor de análisis. Aún así, cabe destacar que PointNet es un modelo supervisado. Esto implica que los datos deben contener etiquetas para poder realizar la labor de análisis de forma correcta.

Como los datos obtenido vienen representados en una una columna, queda claro que los mismos no contienen etiquetas.

Es por ello que para proceder a la implementación del algoritmo, es necesario que se implementen dichas etiquetas.

Para poder proceder a una asignación de etiquetas de la manera más eficaz, se ha decidido emplear previamente a la implementación de PointNet, un modelo no supervisado que intente asignar las etiquetas de manera precisa. Dicho modelo es “K-Means Clustering”.

Este modelo es ampliamente utilizado para la segmentación de objetos en imágenes y se está empezando a proponer como solución para poder identificar con etiquetas documentos de datos que vienen sin etiquetas [45].

El funcionamiento de dicho modelo se basa principalmente en tres fases: la inicialización, la asignación de clústeres, y la actualización de los centroides. La primera fase mencionada, consta de la inicialización de los centroides. Un centroide es un punto que representa el centro dentro de un clúster. La segunda fase comentada, trata de, una vez que los centroides iniciales están ya seleccionados, el algoritmo de Kmeans va a asignar cada punto de datos al clúster cuyo centroide está más cercano (esto se realiza de forma automática dentro del algoritmo empleando la distancia euclidiana, que es la recta entre dos puntos en un espacio multidimensional). La última fase de este algoritmo, trata de ir actualizando los centroides mediante el calculo de la media de todos los puntos de datos en cada clúster. El nuevo punto resultante se convierte en el nuevo centroide del clúster.

Para evitar problemas sobreajuste, en la fase de inicialización, se implementará el modelo de forma que, de forma predeterminada, sea el propio modelo que identifique los clústeres. Esto se ha decidido para que el modelo no caiga en sobreentrenar los datos obtenidos y que así, las etiquetas obtenidas sean las más puras posibles y que se puedan considerar como datos “realistas”.

El proceso de obtención de etiquetas se ha realizado con el siguiente código:

```
1 kmeans = KMeans()  
2 labels = kmeans.fit_predict(features)
```

```
3 pointnet_df['provisional_label'] = labels
```

Listing 3.8: Añadir etiquetas a conjunto de datos

Dicha implementación creará una columna adicional con las etiquetas correspondientes para cada fila.

Aunque de esta manera, se pueden conseguir etiquetas que vayan siendo asignadas según la similaridad de los puntos obtenidos, cabe destacar que dado que PointNet es un modelo supervisado, en este modelo se debe manejar la “verdad absoluta”. En otras palabras, el modelo debe manejar unas etiquetas que están perfectamente asignadas. Dado que el radar es capaz de asignar unas etiquetas, se considerará que a partir de la aplicación del modelo no supervisado empleado, que las etiquetas asignadas son las “reales”. Esta asunción, aunque sea apropiada para la correcta implementación de PointNet, se debe tener en cuenta que es posible que dichas etiquetas no sean del todo precisas y que por tanto se asignen etiquetas a puntos que no corresponden. Esto puede llevar a problemas de sobreajuste de datos o como es comúnmente conocido, “overfitting”.

Este problema es común cuando un modelo memoriza los datos de entrenamiento en lugar de aprender patrones generalizables. Asignar, la misma etiqueta a cada uno de los datos puede dar lugar a este problema si no se gestiona correctamente. Esto se debe a que el modelo podría simplemente aprender a predecir a esa etiqueta sin realmente aprender los patrones de los datos. Esto deberá tenerse en cuenta a la hora de analizar los resultados en el posterior capítulo.

Tras haber implementado un conjunto de etiquetas en los datos, se procederá a desarrollar los conjuntos de entrenamiento, de test y de validación. El primero de los conjuntos mencionados se empleará para entrenar al modelo y servirá para ajustar los parámetros de PointNet. El conjunto de validación, es un conjunto separado que se utiliza durante el entrenamiento para evaluar el rendimiento del modelo. Por último, el conjunto de test, se emplea para evaluar el rendimiento final del modelo.

Para la creación de dichos modelos, se ha empleado el siguiente procedimiento:

```
1 from torch.utils.data import Dataset, DataLoader
2 coordinates_tensor = torch.tensor(pointnet_df["coordinates"].
   to_list(), dtype=torch.float32)
3 labels_tensor = pointnet_df["label"].to_list()
4 dstensor=PointCloudDataset(coordinates_tensor, labels_tensor)
5 dataset_size = len(dstensor)
6 train_size = int(0.8 * dataset_size)
7 val_size = int(0.1 * dataset_size)
```

```

8 test_size = dataset_size - train_size - val_size
9
10 train_dataset, val_dataset, test_dataset = random_split(dstensor,
    [train_size, val_size, test_size], generator=torch.Generator().
    manual_seed(1))
11
12 # Crear DataLoaders para cada conjunto
13 train_dataloader = DataLoader(train_dataset, batch_size=128,
    shuffle=True)
14 val_dataloader = DataLoader(val_dataset, batch_size=128, shuffle=
    True)
15 test_dataloader = DataLoader(test_dataset, batch_size=128, shuffle
    =False)

```

Listing 3.9: creación de los conjuntos de datos

Una vez creados los conjuntos necesarios para poder implementar el modelo, será empleará como base el código desarrollado por PhD. Mariona Carós [47]. El código se emplea mediante la técnica “orientación a objetos” los atributos del conjunto de datos para poder aplicar la arquitectura en su conjunto. Es por ello que para proceder a aplicar el modelo se ha tenido que crear una clase personalizada que obtenga mediante la misma técnica, los atributos necesarios para la implementación posterior del modelo. Dicha clase se ha desarrollado de la siguiente manera:

```

1 class PointCloudDataset(Dataset):
2     def __init__(self, coordinates, labels):
3         self.coordinates = coordinates
4         self.labels = labels
5
6     def __len__(self):
7         return len(self.coordinates)
8
9     def __getitem__(self, idx):
10        return (self.coordinates[idx], self.labels[idx])

```

Listing 3.10: Creación clase necesaria

Tras esto, para poder encontrar el modelo que mejor se ajuste a los datos proporcionados, es necesario encontrar los parámetros óptimos del modelo de PointNet. Para hacer dicha labor se ha empleado el siguiente código:

```

1 import torch
2 import torch.optim as optim
3 import torch.nn as nn
4 from skopt import gp_minimize
5 from skopt.space import Real
6 from skopt.utils import use_named_args
7
8 search_spaces = [
9     Real(1e-4, 1e-2, prior='log-uniform', name='lr'),

```

```

10     Real(1e-4, 1e-2, prior='log-uniform', name='weight_decay'),
11     Real(0.2, 0.5, name='dropout')
12 ]
13
14
15 @use_named_args(search_spaces)
16 def objective(lr, weight_decay, dropout):
17     model = ClassificationPointNet(num_classes=..., dropout=
18     dropout, point_dimension=3)
19     if torch.cuda.is_available():
20         model = model.cuda()
21
22     optimizer = optim.Adam(model.parameters(), lr=lr, weight_decay
23     =weight_decay)
24     criterion = nn.CrossEntropyLoss()
25
26     num_epochs = 10
27     for epoch in range(num_epochs):
28         model.train()
29         for points, targets in train_dataloader:
30             if torch.cuda.is_available():
31                 points, targets = points.cuda(), targets.cuda()
32             optimizer.zero_grad()
33             preds, feature_transform, tnet_out, ix_maxpool = model
34             (points)
35             loss = criterion(preds, targets)
36             loss.backward()
37             optimizer.step()
38
39     model.eval()
40     validation_loss = 0
41     correct = 0
42     total = 0
43     with torch.no_grad():
44         for points, targets in val_dataloader:
45             if torch.cuda.is_available():
46                 points, targets = points.cuda(), targets.cuda()
47             preds, _, _, _ = model(points)
48             loss = criterion(preds, targets)
49             validation_loss += loss.item()
50             _, predicted = preds.max(1)
51             total += targets.size(0)
52             correct += predicted.eq(targets).sum().item()
53
54     try:
55         accuracy = correct / total
56     except ZeroDivisionError:
57         accuracy = 0.0

```

```

56     print(f"Validation accuracy: {accuracy:.4f}, Validation loss:
57         {validation_loss:.4f}")
58     return -accuracy
59
60 res = gp_minimize(objective,
61                  dimensions=search_spaces,
62                  n_calls=50,
63                  random_state=42)

```

Listing 3.11: Búsqueda mejores parámetros

Este código representa en su conjunto diversas iteraciones en una red bayesiana que permite encontrar cada uno de los parámetros que maneja el modelo PointNet (weight-decay, dropout, learning rate). Las redes bayesianas, al modelar probabilísticamente el rendimiento del modelo en función de los hiperparámetros, permiten predicciones informadas y balancean inteligentemente la exploración y explotación de configuraciones. Son eficientes computacionalmente, minimizando evaluaciones costosas y reduciendo significativamente el tiempo de búsqueda en comparación con métodos tradicionales. Además, son adaptables, incorporan conocimientos previos y se ajustan dinámicamente a nueva información. Capturan relaciones complejas y no lineales entre hiperparámetros, buscando soluciones que maximizan el rendimiento global. Estas características hacen que las redes bayesianas sean ideales para optimizar hiperparámetros en modelos complejos como PointNet.

Una vez dicho lo anterior, el código quedará reflejado de forma completa en el Anexo 6.2 de este trabajo.

La implementación completa de esta fase puede verse representada en el diagrama que se muestra a continuación:

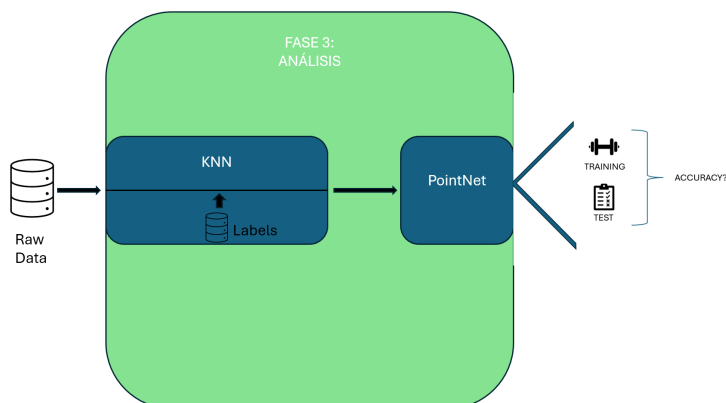


Figura 3.21: Implementación Fase 3

3.5 Implementación modular final del trabajo

Tras haber explicado la última fase de la metodología propuesta, la estructura final del desarrollo de este proyecto puede verse representada por el diagrama que se muestra a continuación:

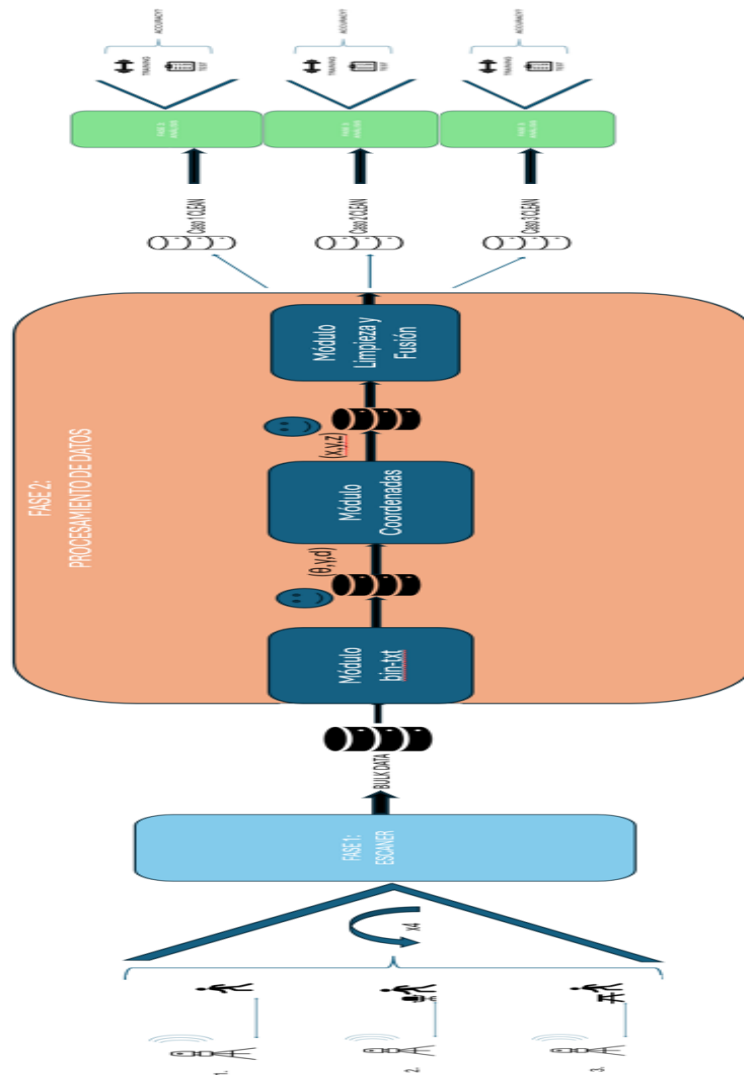


Figura 3.22: Implementación final

Tras haber implementado la metodología de forma completa, se procederá en el siguiente capítulo a analizar los resultados obtenidos.

Capítulo 4

Analisis de los resultados

En este capítulo de análisis se obtendrán los respectivos resultados de precisión de cada uno de los casos explicados en el anterior capítulo, en cada uno de los escenarios concretos.

Para proceder a la labor de análisis de una forma adecuada se emplearán distintas medidas. La primera de las medidas que se emplearán será de una gráfica 3D resultante del proceso de aplicar los modelos mencionados en el anterior capítulo. Esta gráfica servirá principalmente para ver si realmente se puede identificar los objetos y el movimiento que se pretende emplear en cada escenario.

La segunda medida que se empleará será una gráfica que explicará si bien los resultados obtenidos de la gráfica 3D proceden de un modelo suficientemente explicativo. Esto se determinará con una gráfica con las medidas de pérdidas de información de los conjuntos de entrenamiento y de validación. El modelo se considerará explicativo si bien los valores de pérdida de información se estabilizan entorno a valores cercanos a cero (lo que implicaría que los modelos pierden poca información). El comportamiento esperado de esta gráfica es la aparición de una serie de fluctuaciones que indiquen que el modelo busca su forma más óptima para poder representar la mejor forma de los datos.

La tercera medida que se empleará será el análisis de la matriz de confusión obtenida después de aplicar los modelos de Machine Learning. De esta matriz se estudiarán las siguientes medidas para cada una de las clases detectadas por PointNet:

- **Precisión:** Mide la exactitud de las predicciones positivas. Un valor alto indica que la mayoría de las predicciones positivas son correctas.
- **Recall:** Mide la capacidad del modelo para identificar todos los ejemplos

positivos. Un valor alto indica que el modelo detecta la mayoría de los verdaderos positivos.

- **F1-score:** Proporciona un equilibrio entre precisión y recall. Es especialmente útil cuando se necesita un compromiso entre ambos y cuando las clases están desbalanceadas.
- **Support:** Indica cuántos ejemplos de cada clase se tienen. Es útil para entender la distribución de las clases en el conjunto de datos.
- **Medidas Macro Average:** Son promedios no ponderados de las métricas (precisión, recall, y F1-score) calculadas para cada clase individualmente. Cada clase tiene el mismo peso, independientemente de la cantidad de ejemplos que tenga en el conjunto de datos.
- **Medidas Weighted Average:** Son promedios ponderados de las métricas (precisión, recall, y F1-score) calculadas para cada clase, donde el peso de cada clase es proporcional al número de instancias de esa clase en el conjunto de datos.

Una vez vistos los elementos que se utilizarán en este capítulo, se procederá a continuación a analizar los resultados obtenidos para cada uno de los escenarios, en cada una de las situaciones de luminosidad, propuestas para este trabajo.

4.1 Condición luminosa óptima

En esta sección se analizarán los tres escenarios (movimiento sin objetos, movimiento con una silla de por medio y movimiento con una mesa de por medio) con una condición luminosa óptima.

4.1.1 Escenario 1: Detección de objetos sin objetos

En este primer escenario, el resultado 3D de aplicar el modelo de PointNet se puede apreciar en la figura 4.1.

En esta figura se puede apreciar como el modelo ha sido capaz de clasificar 3 clases. Aunque de dicha representación no se pueden obtener conclusiones acerca de la persona que se está moviendo en dicho escenario, si que se pueden clasificar 3 etapas de movimiento que reflejan la trayectoria realizada por la persona:

En primer lugar lo que se ha clasificado como clase “0” contiene una enorme cantidad de puntos. Esta clase puede representar el inicio del movimiento y cuando la persona empieza a acelerar para realizar el recorrido. Luego la clase “1” representa algunos puntos (menos que los clasificados en la clase “0”) y puede representar el

movimiento de la persona ya cercano al sensor. Por último, la clase “2” es la que tiene menos puntos y más esparcidos que puede representar el final del movimiento cuando la persona ha llegado ya al final del recorrido para parar la simulación del radar.

Estos resultados son muy relevantes y han resultado según lo esperado. Esto se debe a que como se comentó en el anterior capítulo, en concreto en la sección de la primera fase de la metodología empleada, el sensor empleado no permite la detección de objetos estáticos. Esta limitación que tiene el radar, hace que en cuanto un objeto se paraliza después de realizar un movimiento, el radar dado que no es capaz de identificar dichos puntos, mete una función de ruido que es proporcional a los puntos recogidos previamente. Esto justifica el porque en el inicio del movimiento hay muchos más puntos concentrados que al final, donde los puntos son más esparcidos. En adición de esta gráfica se puede inducir perfectamente cual es la trayectoria de la persona, la cual es un movimiento lineal hacia el sensor.

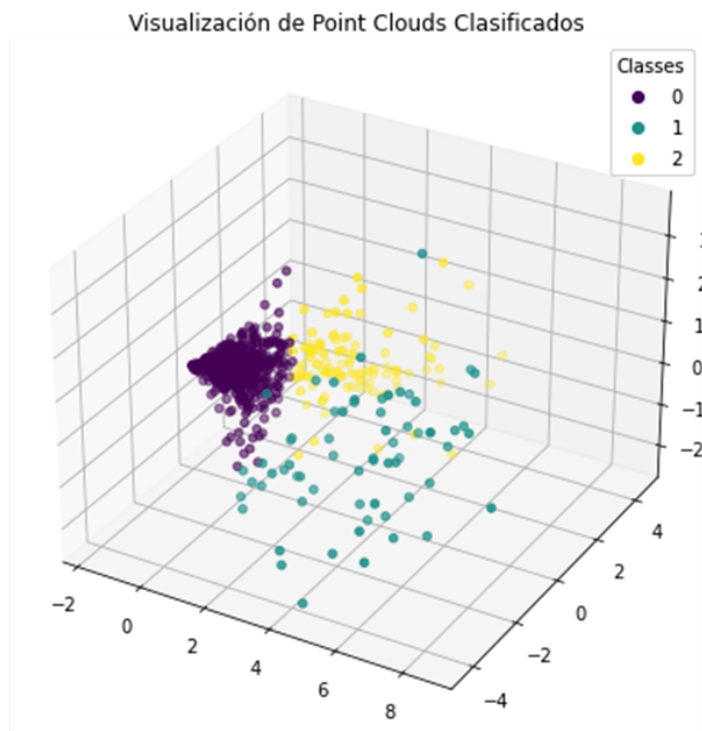


Figura 4.1: Representación 3D Escenario 1 con luz

Tras haber explicado la gráfica 3D resultante, para poder corroborar que los resultados obtenidos son relevantes, se va a examinar a continuación la gráfica de

perdida de información de los conjuntos de entrenamiento y validación empleados:

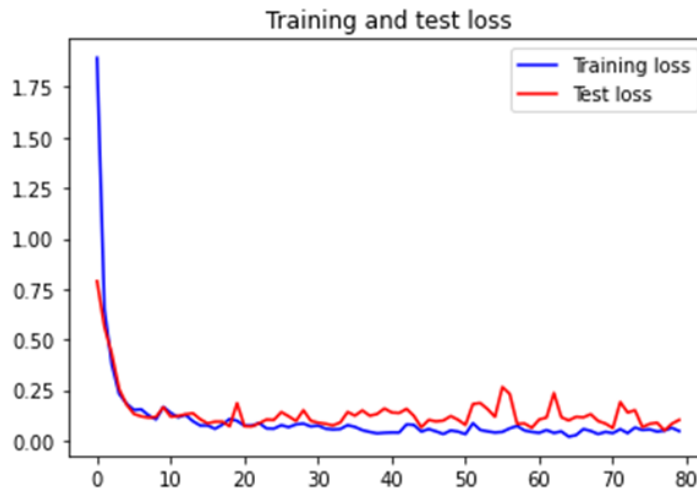


Figura 4.2: Gráfica de pérdida de información Escenario 1 con luz

Como se puede apreciar, tanto los conjuntos de entrenamiento como de validación, presentan fluctuaciones a lo largo de las iteraciones del modelo de PointNet. Se observa como al principio del modelo, el mismo no ha encontrado su forma óptima, en cambio al final del modelo se puede observar como los conjuntos tanto de validación como de entrenamiento se estabilizan en niveles de pérdida de información entorno al 0.1, lo que implica que el resultado obtenido proviene de un modelo es su mejor forma y por tanto implica que los resultados pueden ser significativos. Para determinar en profundidad si bien los resultados obtenidos son significativos, se hará un análisis en detalle de las medidas de la matriz de confusión obtenida para este escenario (ver Figura 4.3):

El modelo PointNet muestra un rendimiento excepcional en la detección de una persona en movimiento con un radar de ondas milimétricas. Para la Clase 0, el modelo presenta una precisión y recall muy altos, resultando en una F1-score de casi 1.0, lo que sugiere una identificación precisa y consistente de las instancias. En la Clase 1, aunque la precisión es perfecta, el recall ligeramente inferior (0.90) indica que algunas instancias no se detectan, pero la F1-score de 0.95 sigue siendo muy alta, indicando un buen equilibrio general. La Clase 2 muestra un rendimiento sólido con una precisión y recall de 0.91, resultando en una F1-score de 0.91, lo que sigue siendo un buen rendimiento aunque no tan alto como las otras clases. Globalmente, el modelo tiene una exactitud del 98 %, indicando alta precisión. Las medidas macro promedio muestran un rendimiento balanceado entre clases, con una ligera disminución en el recall, mientras que las medidas promedio ponderadas

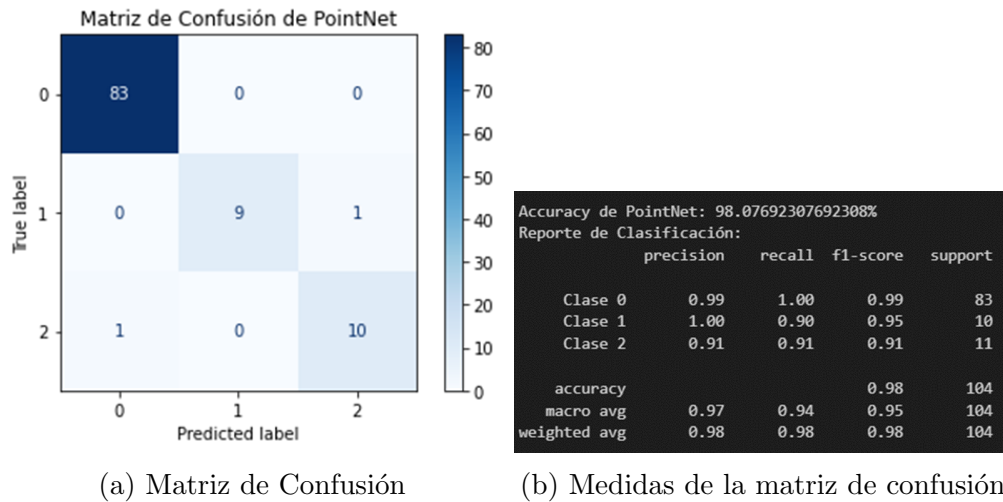


Figura 4.3: Matriz de confusión Escenario 1 con luz

reflejan una alta precisión y recall considerando la distribución de las clases. En conclusión, para este escenario concreto, PointNet es altamente preciso y balanceado para poder determinar movimientos, con posibles mejoras en el recall de la Clase 1. Las métricas sugieren que el modelo es efectivo y adecuado, con una exactitud general del 98 % y altas F1-scores, lo que sugiere que los resultados eD para este escenario concreto son significativos y robustos.

Aun así, se puede observar una clara desproporción en cuanto al número de instancias por cada clase detectada, lo que puede provocar que haya cierto sobreajuste para la clase minoritaria. Esto aunque sea relevante, no quita que los resultados encontrados para este escenario sean excelentes.

4.1.2 Escenario 2: Detección de objetos con una silla

Para este escenario, los resultados 3D son los mostrados en la Figura 4.4: Se puede observar en este caso, que el modelo identifica 2 clases claras. Aunque esto difiera de lo encontrado en el escenario anterior, cabe destacar que el resultado también muestra claramente distintas fases de movimiento: una primera fase inicial que sería el principio del movimiento realizado con la silla, donde se pasa de estar estático a estar en movimiento (donde existe una aceleración). Este fase vendría representada por la clase “1” en donde hay una cantidad compacta de puntos. La clase “0” en cambio representaría la última fase del movimiento donde cada vez hay menos aceleración (hasta detener el movimiento para terminar la simulación) y viene dado por puntos más dispersos.

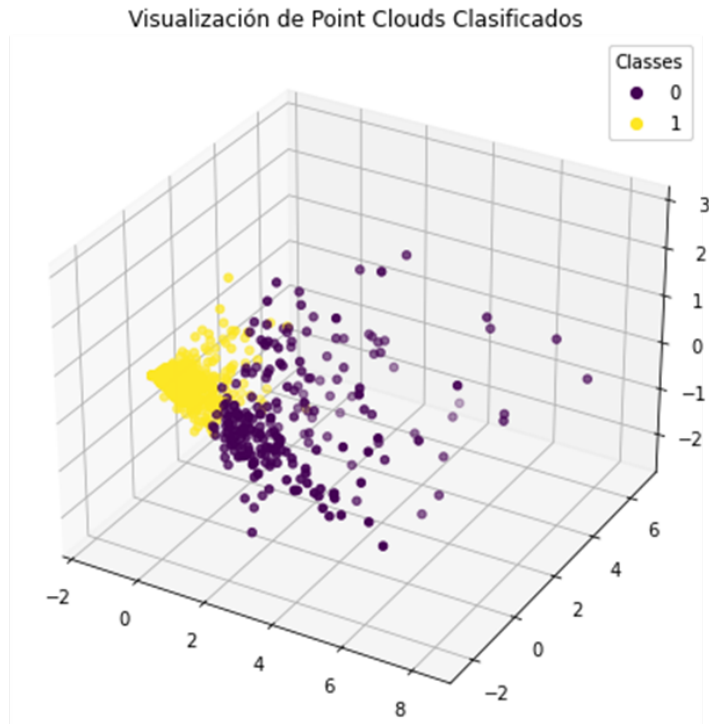


Figura 4.4: Representacion 3D Escenario 2 con luz

Como se comentó en el anterior escenario dichos resultados tienen sentido dado que el sensor cuando detecta un movimiento inminente con una cierta aceleración es capaz de mapear correctamente los puntos obtenidos (lo que explicaría la distribución compacta de puntos de la clase “1”), pero en cambio cuando el movimiento disminuye y por tanto la aceleración también, el sensor tiene más difícil mapear los datos recogidos y es por ello que introduce una variable de ruido para poder intentar inducir, donde se encontraría el objeto ya estático.

Si bien es cierto que esto justifica la distribución de los puntos, y que al mismo tiempo se puede observar claramente como la persona y el objeto sigue una trayectoria lineal, se puede apreciar al mismo tiempo que el gráfico podría haber identificado 3 clases como en el escenario anterior para calificar 3 fases de movimiento distinta. Esto puede deberse a que dado que la persona y la silla se están moviendo al mismo tiempo, el sensor haya confundido puntos. Esto es debido a que para terminar la simulación de este escenario, el movimiento se haya terminado antes de lo debido (dado que el sensor no se ha cambiado de posición con respecto al primer escenario) y que por tanto el sensor haya detectado solo 2 clases significativas dado que no se haya identificado una fase lo suficientemente

representativa como para clasificarse en una clase aparte. Esto no implica que los resultados sean erróneos, sino todo lo contrario.

Para comprobar si bien lo encontrado es significativo, se va a proceder ahora a analizar la gráfica de pérdida de información de este escenario: Se puede observar,



Figura 4.5: Gráfica de pérdida de información Escenario 2 con luz

al igual que en el escenario anterior, una serie de fluctuaciones entre los conjuntos de entrenamiento y validación, pero que al final del proceso de iterar el modelo de PointNet, ambos conjuntos de datos se estabilizan y convergen entorno a valores entorno a 0.15. Esto significa que los resultados obtenidos provienen de el modelo más óptimo y que el mismo es significativo. Aún así, para verificar esto, se va a proceder a analizar los valores obtenidos para la matriz de confusión de este escenario (Ver Figura 4.6).

El reporte de clasificación de PointNet muestra un desempeño sobresaliente con una precisión general del 96.43 %. La precisión para la Clase 0 es del 100 %, lo que indica que todos los puntos clasificados como Clase 0 son correctos, aunque el recall es del 82 %, sugiriendo un 18 % de falsos negativos. El f1-score para esta clase es del 90 %, reflejando un buen equilibrio entre precisión y recall. Para la Clase 1, la precisión es del 96 % y el recall es del 100 %, lo que significa que el modelo identifica correctamente todos los puntos de esta clase con un f1-score del 98 %. La exactitud general del modelo es del 96 %, lo que demuestra su alta capacidad de clasificación. Los promedios macro y ponderado para precisión, recall y f1-score también son altos, con valores de f1-score de 0.94 y 0.96 respectivamente, indicando un balance y rendimiento consistente en ambas clases. En resumen, el modelo PointNet tiene una

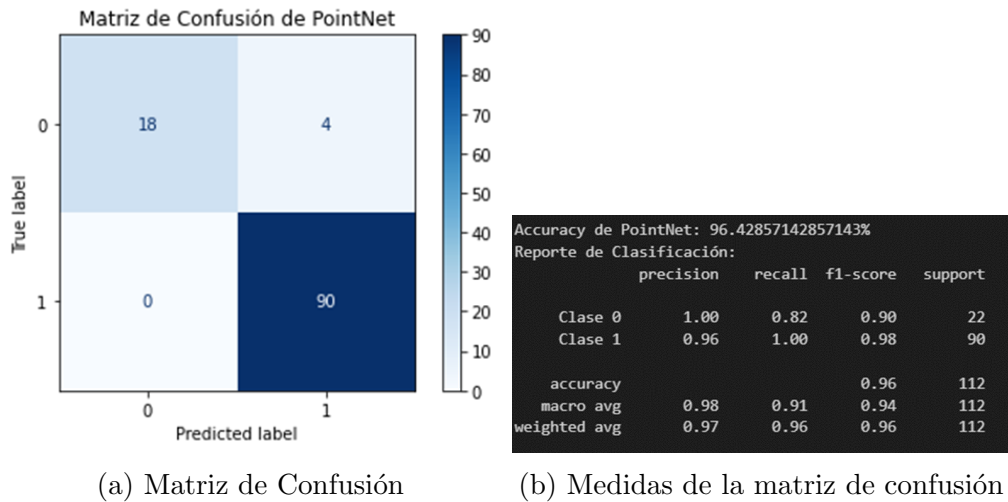


Figura 4.6: Matriz de confusión Escenario 2 con luz

alta capacidad de clasificación, especialmente para la Clase 1, aunque hay margen de mejora en el recall de la Clase 0, la cual es claramente una clase minoritaria. Esto como en el escenario anterior, puede dar lugar a problemas de sobreajuste, aunque esto no quita que el rendimiento en general del modelo es óptimo y permite inducir claramente el movimiento que se realiza en este escenario: un movimiento lineal hacia el sensor.

Gracias a estos resultados por tanto, se puede concluir que de nuevo PointNet se presenta como una técnica fiable para detectar movimiento y es capaz de representarlo de forma significativa.

4.1.3 Escenario 3: Detección de objetos con una mesa

En este último escenario la representación visual resultante de haber aplicado el modelo PointNet puede verse reflejada en la Figura 4.7. En dicha Figura se puede apreciar como al igual que en el escenario 2 se identifican 2 clases. La clase “1” que refleja los puntos amarillos más compactos y la clase “0” que refleja los puntos morados más dispersos. De nuevo, se puede observar (incluso en este escenario con más claridad) el movimiento lineal ejercido en este caso por la persona y la mesa que se empleó en este escenario. Se observa como el principio del movimiento (el que viene representado por la clase “1”), los puntos aparecen más compactos y mejor mapeados que los representados por la clase “0”, los cuales aparecen más dispersos.

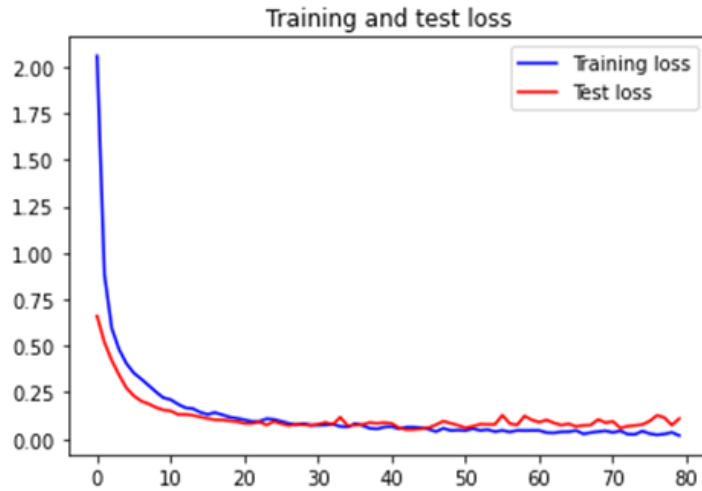
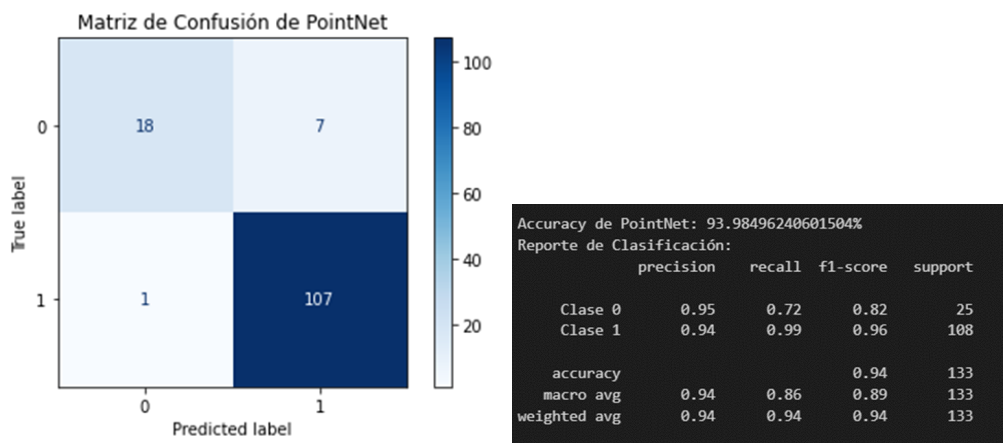


Figura 4.8: Gráfica de perdida de información Escenario 3 con luz

Se puede apreciar en dicha figura que los conjuntos de validación y entrenamiento, en las etapas iniciales del modelo ambos conjuntos difieren. En cambio, se puede apreciar al mismo tiempo como al final del modelo, ambos conjuntos se estabilizan en niveles de perdida de información entorno al 0.15. Esto muestra como el modelo PointNet ha alcanzado su forma más óptima y que por tanto los resultados obtenidos en la representación 3D provista son fiables y robustos. Aún así, para poder ver si los resultados son significativos, se va a proceder a analizar las métricas obtenidas a partir de la matriz de confusión resultante del proceso:



(a) Matriz de Confusión

(b) Medidas de la matriz de confusión

Figura 4.9: Matriz de confusión Escenario 3 con luz

El reporte de clasificación de PointNet muestra un desempeño robusto con una precisión general del 93.98 %. La precisión para la Clase 0 es del 95 %, lo que indica que la mayoría de los puntos clasificados como Clase 0 son correctos, aunque el recall es del 72 %, sugiriendo un 28 % de falsos negativos. El f1-score para esta clase es del 82 %, reflejando un equilibrio aceptable entre precisión y recall. Para la Clase 1, la precisión es del 94 % y el recall es del 99 %, lo que significa que el modelo identifica correctamente casi todos los puntos de esta clase, con un f1-score del 96 %. La exactitud general del modelo es del 94 %, lo que demuestra su alta capacidad de clasificación. Los promedios macro y ponderado para precisión, recall y f1-score también son altos, con valores de f1-score de 0.89 y 0.94 respectivamente, indicando un balance y rendimiento consistente en ambas clases. En resumen, el modelo PointNet tiene una alta capacidad de clasificación, especialmente para la Clase 1, aunque hay margen de mejora en el recall de la Clase 0.

De nuevo, tras haber analizado la matriz de confusión se puede concluir que el modelo es totalmente fiable y los resultados son asimismo significativos. De todos modos, se puede apreciar que hay un claro desbalance en cuanto al número de instancias por clase detectadas, lo que puede dar lugar a problemas de sobreajuste para la clase minoritaria. Aún así, es visible que dicho problema no parece afectar a la efectividad que presenta el modelo en este caso concreto.

Tras haber analizado los tres escenarios, se puede establecer (como en los dos escenarios anteriores) que PointNet es del todo fiable para detectar con claridad movimiento de personas y objetos en un escenario en condiciones luminosas óptimas.

4.2 Condición luminosa nula

A continuación se proceden a analizar los resultados para cada uno de los escenarios propuestos, en condiciones de visibilidad nula.

4.2.1 Escenario 1: Detección de objetos sin objetos

La representación 3D resultante de aplicar el modelo es la que se muestra a continuación en la figura 4.10.

En dicha figura, se puede apreciar como los resultados son muy similares a los encontrados en condiciones luminosas óptimas. Se puede apreciar que el modelo ha clasificado 3 clases distintas: la clase “0” que viene dada por una serie de puntos

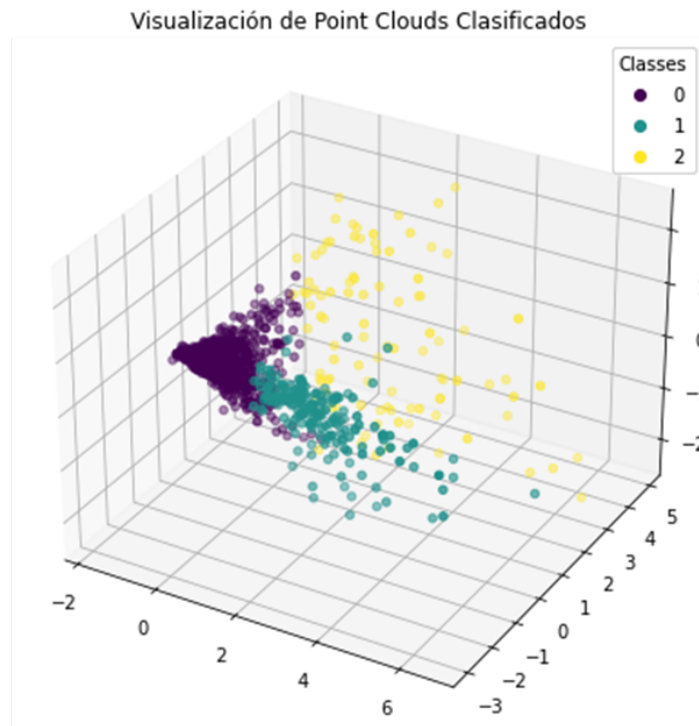


Figura 4.10: Representación 3D Escenario 1 sin luz

morados muy compactos; la clase “1” que viene dada por una serie de puntos azules algo más dispersos que los clasificados en la clase “0”; y la clase “2” que viene dada por una serie de puntos amarillos muy dispersos en comparación con las dos clases mencionadas anteriormente.

En dicha representación, se pueden apreciar perfectamente las tres fases del movimiento lineal realizado por la persona en este escenario: la primera fase de aceleración en donde la persona inicia el movimiento (que se corresponde a la clase “0”) y en donde los puntos están mejor mapeados y de forma más compacta; la segunda fase del movimiento donde ya la persona esta moviéndose a paso constante (que se corresponde con la clase “1”) y que los puntos son más dispersos dado que la persona se acerca al radar y este tiene más complicado mapear correctamente los puntos; y la clase “2” se corresponde justo con el final del movimiento, cuando se termina la simulación y que el radar tiene que introducir una variable de ruido para intentar trazar el movimiento de la persona.

Lo encontrado sigue el mismo patrón encontrado que en el escenario con luz, lo cual entre dentro de lo esperado. Ahora para confirmar que dicha representación

es robusta se analizará la gráfica de pérdida de información de este escenario:

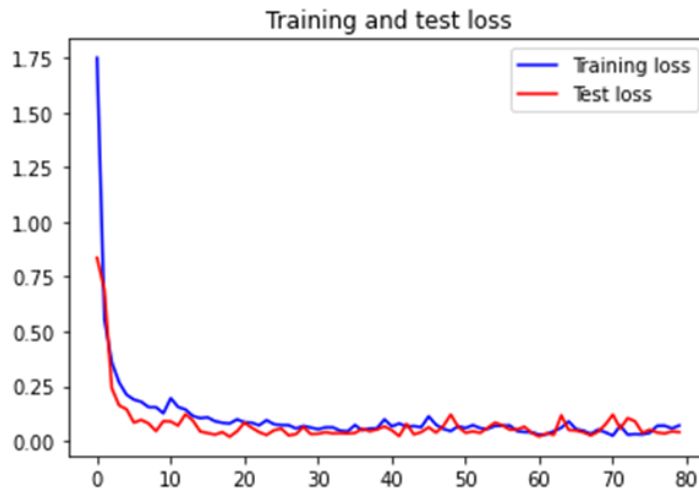


Figura 4.11: Gráfica de pérdida de información Escenario 1 sin luz

Se puede observar que los conjuntos de validación y entrenamiento al principio del proceso de iteración del modelo PointNet, difieren mostrando distintos niveles de pérdida de información, lo que indica que el modelo no ha alcanzado en esa etapa su forma más óptima. En cambio se puede observar como al final de las iteraciones del modelo, ambos conjuntos de datos se estabilizan entorno al 0.1, lo que indica que el modelo ha alcanzado su forma más óptima. Esto implica que efectivamente el modelo ha alcanzado su forma más óptima y que por tanto los resultados obtenidos son robustos.

Ahora para determinar si bien dichos resultados son significativos, se va a proceder a analizar las métricas resultantes de la matriz de confusión obtenida, la cual se puede observar en la figura 4.12.

El reporte de clasificación de PointNet muestra un desempeño excepcional con una precisión general del 98.51 %. La precisión para la Clase 0 es del 99 %, indicando que casi todos los puntos clasificados como Clase 0 son correctos, y el recall es del 100 %, lo que significa que el modelo identifica correctamente todos los puntos de esta clase. El f1-score para esta clase es del 100 %, reflejando un equilibrio perfecto entre precisión y recall. Para la Clase 1, la precisión y el recall son ambos del 95 %, resultando en un f1-score del 95 %. Esto muestra que el modelo maneja bien la

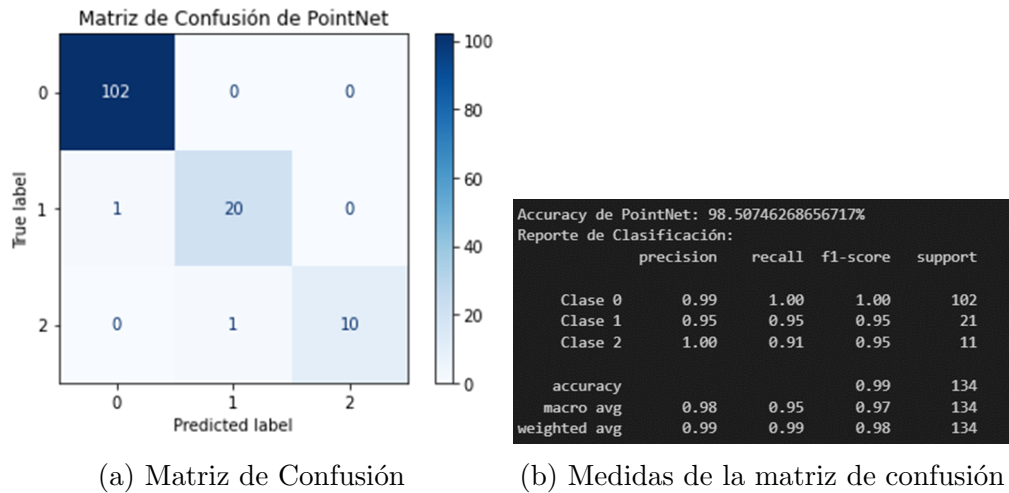


Figura 4.12: Matriz de confusión Escenario 1 sin luz

clasificación de esta clase, aunque hay una pequeña proporción de falsos positivos y falsos negativos. Para la Clase 2, la precisión es del 100 % y el recall es del 91 %, con un f1-score del 95 %, indicando una excelente identificación con una ligera proporción de falsos negativos.

La exactitud general del modelo es del 99 %, lo que demuestra su alta capacidad de clasificación. Los promedios macro y ponderado para precisión, recall y f1-score son muy altos, con valores de f1-score de 0.97 y 0.98 respectivamente, indicando un balance y rendimiento consistente en todas las clases. En resumen, el modelo PointNet tiene una alta capacidad de clasificación para todas las clases, con un rendimiento particularmente fuerte en la Clase 0, y un excelente equilibrio entre precisión y recall en las Clases 1 y 2.

Tras haber analizado la matriz de confusión, se puede concluir que en condiciones de luminosidad nula, PointNet es un modelo capaz de captar el movimiento lineal realizado por la persona de manera eficaz y significativa. De nuevo, se pueden apreciar desbalances en las clases encontradas por PointNet que puede llevar a problemas de sobreajuste para la clase minoritaria. Aún así, esto no parece afectar al rendimiento general del modelo, que es aparentemente excelente.

4.2.2 Escenario 2: Detección de objetos con una silla

Para este escenario, la representación 3D resultante es la que se muestra en la Figura 4.13.

Se puede apreciar como en este escenario se encuentran unos resultados muy

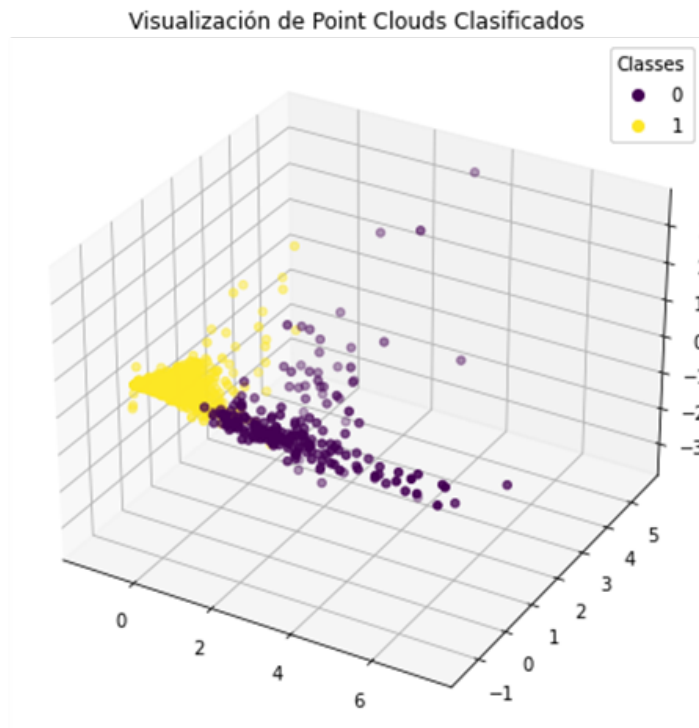


Figura 4.13: Representación 3D Escenario 2 sin luz

similares a los encontrados en condiciones luminosas óptimas: dos clases, una amarilla que conforma una serie de puntos más compacto (la clase “1”) y otra morada que contiene una serie de puntos más dispersa (la clase “0”). Al igual que en condiciones luminosas óptimas, las clases identificadas representan distintas fases de movimiento: la primera fase en donde se inicia el movimiento y que viene dada por una serie de puntos más compactos y mejor mapeados, y luego una segunda fase que viene dada con puntos más dispersos que representa la fase final del movimiento con la silla, en donde se para la simulación de este escenario. En adición, se puede observar que en esta fase hay unos puntos más dispersos que el resto que representan la variable de ruido que introduce el sensor ya cuando se termina el movimiento.

De nuevo este resultado no indica 3 fases de movimiento como en el escenario que no implica objetos, y esto se debe a que el movimiento realizado no ha sido completo y se haya detenido antes la simulación, motivo por el que se ha introducido esa variable de ruido mientras el persona se movía con la silla. Aún así se puede apreciar de nuevo de forma clara como se realiza un movimiento lineal hacia el radar, lo cual entra dentro de lo esperado.

Para determinar si dichos resultados son robustos, se va a proceder a analizar la gráfica de pérdida de información de este escenario, que viene representada en la figura 4.14: Se puede apreciar como (al igual que en los casos ya analizados), los



Figura 4.14: Gráfica de pérdida de información Escenario 2 sin luz

conjuntos de entrenamiento y validación difieren al principio del proceso del modelo, pero al final de las iteraciones, ambos conjuntos se estabilizan entorno a valores entre 0.25 y 0.47. Aunque estos valores de pérdida de información son más altos de lo esperado, lo cierto es que aún así indican que el modelo ya ha encontrado su forma más óptima y que por tanto los resultados son ciertamente robustos.

Aún así, para poder determinar si bien los resultados obtenidos son significativos, se procederá al análisis de la matriz de confusión obtenida para este escenario: El reporte de clasificación de PointNet muestra un desempeño sólido con una precisión general del 93.66 %. La precisión para la Clase 0 es del 92 %, indicando que la mayoría de los puntos clasificados como Clase 0 son correctos, aunque el recall es del 60 %, sugiriendo un 40 % de falsos negativos. El f1-score para esta clase es del 73 %, reflejando un equilibrio aceptable entre precisión y recall. Para la Clase 1, la precisión es del 94 % y el recall es del 99 %, lo que significa que el modelo identifica correctamente casi todos los puntos de esta clase, con un f1-score del 96 %. La exactitud general del modelo es del 94 %, lo que demuestra su alta capacidad de clasificación. Los promedios macro y ponderado para precisión, recall y f1-score son altos, con un f1-score de 0.85 y 0.93 respectivamente, indicando un balance y rendimiento consistente en ambas clases. En resumen, el modelo PointNet tiene una alta capacidad de clasificación, especialmente para la Clase 1, aunque hay margen de mejora en el recall de la Clase 0.

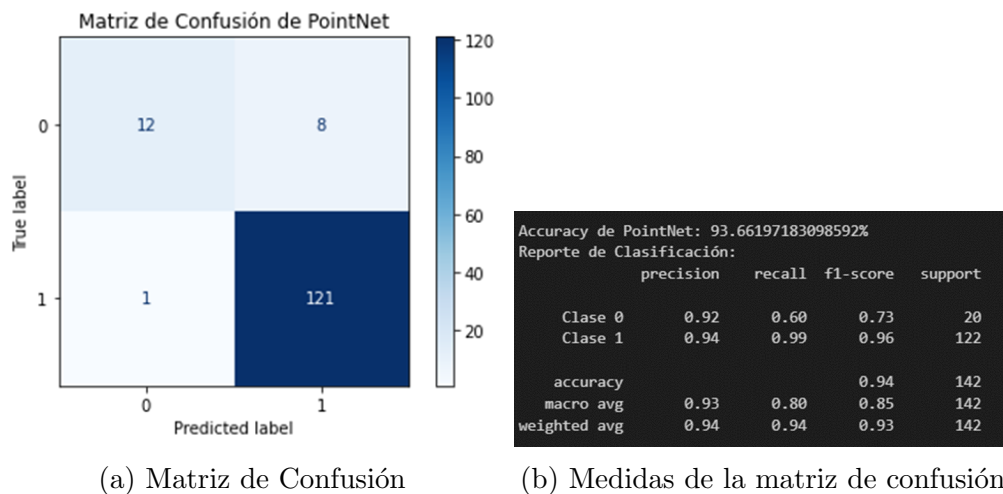


Figura 4.15: Matriz de confusión Escenario 2 sin luz

Tras este análisis, se puede apreciar que el modelo PointNet es aún con valores de pérdida de información más alto que en escenarios anteriores, óptimo y eficaz para la detección de movimiento de personas de forma significativa. Aún así es cierto que como en escenarios anteriores, PointNet presenta un claro desbalanceo en las clases lo que puede dar lugar a problemas de overfitting, pero que siguen sin afectar a la calidad de los resultados (que como se ha dicho, son muy óptimos).

4.2.3 Escenario 3: Detección de objetos con una mesa

Para este último caso la representación visual 3D de los resultados obtenidos por el modelo PointNet, es la dada en la figura 4.16:

Al igual que en el escenario con condiciones luminosas óptimas, este escenario presenta dos clases que representan dos fases distintas de movimiento: una primera fase inicial que representa el inicio de la simulación donde la persona se empieza a mover con la mesa y donde los puntos están más compactos y mejor mapeados (clase “0”), mientras que la otra clase identificada por el modelo PointNet (clase “1”) contiene una serie de puntos más dispersa y que representa la fase final de la simulación, incluida también la variable de ruido que ha introducido el radar para poder mapear de mejor manera los detectados en dicha fase. De nuevo, se observa que el modelo PointNet no ha identificado 3 fases de movimiento (como en todos los escenarios que involucran algún objeto de por medio) debido a que el movimiento no se ha realizado complementamente y por tanto la simulación haya sido mucho más corta de lo que debiera ser (razón por la que se hayan mezclado en

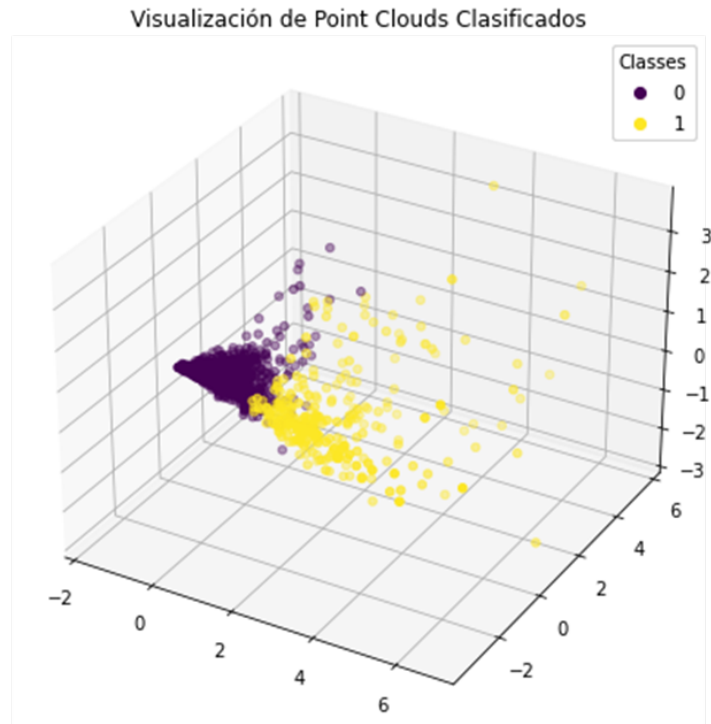


Figura 4.16: Representación 3D Escenario 3 sin luz

una sola clase, dos fases de movimiento). Aún así se puede observar con claridad el movimiento realizado por la persona: un movimiento lineal hacia el radar. De nuevo, se encuentra una representación visual clara del movimiento como en los casos ya estudiados en este capítulo.

Para verificar este resultado, se va a proceder a analizar la gráfica de pérdida de información de este escenario, que viene dada en la figura 4.17:

Se puede observar que al igual que en el resto de casos anteriores, el modelo alcanza su forma más óptima al final del proceso de iterar el modelo PointNet, donde los conjuntos de entrenamiento y validación convergen entorno al valor de 0.13. Esto indica que el resultado obtenido proviene de un modelo robusto, por lo que el resultado obtenido puede considerarse robusto.

Para poder analizar si bien los resultados son significativos, se va a proceder a analizar la matriz de confusión de este escenario, y que puede verse en la figura 4.18.

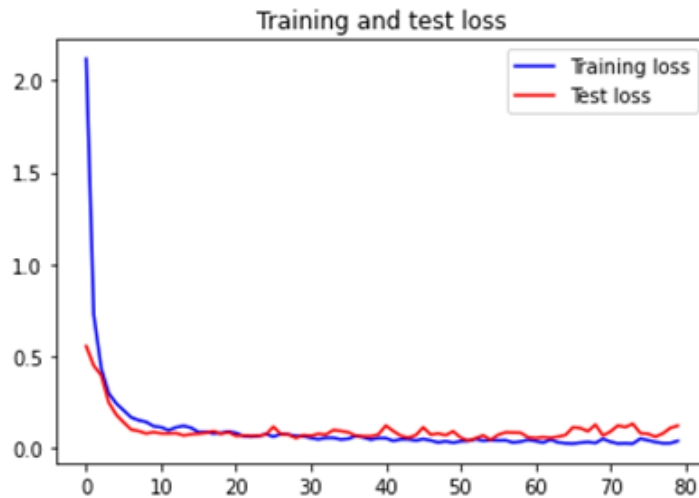


Figura 4.17: Gráfica de pérdida de información Escenario 3 sin luz

El reporte de clasificación de PointNet muestra un desempeño robusto con una precisión general del 94.74 %. La precisión para la Clase 0 es del 94 %, indicando que la mayoría de los puntos clasificados como Clase 0 son correctos, con un recall del 100 %, lo que significa que el modelo identifica correctamente todos los puntos de esta clase. El f1-score para esta clase es del 97 %, reflejando un equilibrio excelente entre precisión y recall. Para la Clase 1, la precisión es del 100 % y el recall es del 73 %, lo que sugiere un 27 % de falsos negativos. El f1-score para esta clase es del 85 %, mostrando un buen equilibrio, aunque con margen de mejora en la exhaustividad. La exactitud general del modelo es del 95 %, lo que demuestra su alta capacidad de clasificación. Los promedios macro y ponderado para precisión, recall y f1-score son altos, con un f1-score de 0.91 y 0.94 respectivamente, indicando un balance y rendimiento consistente en ambas clases. En resumen, el modelo PointNet tiene una alta capacidad de clasificación, especialmente para la Clase 0, aunque hay margen de mejora en el recall de la Clase 1.

Tras estos resultados, se puede concluir que de nuevo, el modelo PointNet es capaz de identificar el movimiento realizado y pretendido para este escenario de forma eficaz y óptima. De nuevo, como a lo largo de todo el análisis, se encuentra claramente un desbalanceo de clases, donde se aprecia una clase minoritaria (lo que puede dar lugar a problemas de sobreajuste del modelo). Aún así, este problema no parece afectar a la capacidad explicativa que ha obtenido el modelo para este caso: capaz de identificar el movimiento realizado por la persona.

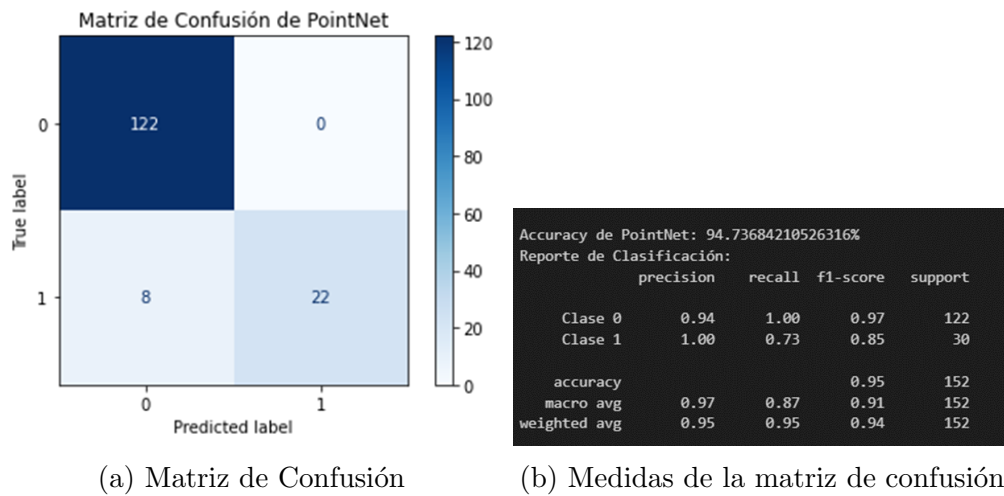


Figura 4.18: Matriz de confusión Escenario 3 sin luz

Tras este capítulo de análisis, no solo se ha podido determinar que PointNet es un modelo eficaz para poder identificar un movimiento realizado por una persona con éxito, si no que también se ha podido comprobar que las tecnologías de ondas milimétricas presentan una clara ventaja sobre las tecnologías de detección que emplean ondas del espectro visible: mediante el uso de tecnologías de detección que emplean ondas milimétricas, se han encontrado resultados sino idénticos, pues muy similares, en condiciones de visibilidad muy dispares, lo que lo hace (como se ha indicado a lo largo de este trabajo) robusto y fiable frente a condiciones de luminosidad variable.

4.3 El problema de la privacidad de datos

Independientemente de los resultados obtenidos en el anterior capítulo (y por tanto dejando a un lado si bien se han cumplido los objetivos principales de este trabajo o no), dada la importancia de este tipo de tecnologías, y dado que se van a seguir desarrollando constantemente en el futuro para mejorar la seguridad y fiabilidad en distintos ámbitos, es de vital importancia analizar las implicaciones éticas que conlleva el desarrollo de este tipo de tecnologías.

Tal y como se comentó en el capítulo de introducción de este trabajo, en el

ámbito de detección de datos se pueden encontrar principalmente dos tipos de tecnologías: las que emplean ondas del espectro visible para dar lugar a cámaras de visión, y las que emplean sensores que miden ondas milimétricas. Lo cierto es que aunque las tecnologías de ondas milimétricas permiten dar más privacidad dado que no tienen tanta resolución de los detalles de los entornos que se analizan (como es el caso de las cámaras de visión, que aportan precisión a la hora de saber todo tipo de detalles del entorno que se analiza), pueden dar lugar a dilemas éticos que no solo requieren su identificación, sino que también formas de abordarlos para poder proporcionar a la sociedad no solo tecnologías más precisas y fiables, sino que también protejan a los individuos de la misma.

El principal problema que surge a raíz del desarrollo de dichas tecnologías es la privacidad de datos. Este problema se refiere a la complejidad que existe a la hora de proteger la información personal y sensible en el manejo sobre grandes volúmenes de datos. En el contexto de la detección de datos, la protección de datos implica garantizar que tanto las imágenes como los videos capturados y construidos a través de las tecnologías de detección, no infrinjan los derechos de privacidad de las personas presentes en ellos.

Para poder estudiar en profundidad las posibles implicaciones éticas que pueden surgir tanto del desarrollo de tecnologías de detección como del uso de las mismas, se explorarán en profundidad tres dilemas éticos clave y se propondrán soluciones para abordarlos de manera correcta.

4.3.1 Vigilancia Masiva y Derecho a la Privacidad

Uno de los dilemas más identificables en la detección de objetos es el potencial que tiene esta labor para la vigilancia masiva. Los sistemas de detección de objetos pueden desplegarse tanto en espacios públicos como privados (como se ha visto en distintas aplicaciones de dichas tecnologías en diversos estudios a lo largo de este trabajo). Esto puede llegar a permitir que las entidades gubernamentales y corporativas puedan monitorear las actividades de las personas en tiempo real sin su consentimiento explícito. Es por ello que, este tipo de vigilancia puede erosionar el derecho a la privacidad [63]. Esto puede dar lugar a la creación de una sociedad donde los individuos de la misma se sientan bajo vigilancia. Esto puede dar lugar a efectos adversos en el comportamiento y la libertad individual. Este tipo de “sociedad bajo vigilancia” se puede ver similar a la descrita por George Orwell en su novela “1984” [13].

En la sociedad descrita por Orwell se pueden identificar claramente varias preocupaciones éticas que surgen a raíz de una sociedad vigilada de forma masiva:

- **Violación del derecho a la privacidad:** Un constante monitoreo de datos visuales sin el consentimiento explícito de las personas puede ser una violación directa de su derecho a la privacidad. Esto es especialmente preocupante en espacios públicos donde las personas pueden no ser conscientes de que están siendo vigiladas.
- **Impacto en la libertad individual:** Dentro de una sociedad vigilada de forma masiva, puede tener consecuencias en términos de libertad de expresión y de movimiento de las personas que viven en la misma. Esto se debe a que los individuos pueden sentirse cohibidos para participar en ciertas actividades o expresar ciertas opiniones.
- **Abuso de poder:** La capacidades que dan las tecnologías de última generación a entidades tan poderosas como gobiernos o incluso a corporaciones multinacionales, puede llegar a propiciar este problema. La vigilancia de activistas, periodistas y opositores políticos, puede dar lugar a situaciones de represión o incluso violación de derechos humanos.

Aunque estos problemas son preocupantes, para poder abordarlos de forma correcta, sería esencial aplicar el concepto de "privacidad por diseño" propuesto por Ann Cavokian [10]. Esta teoría sugiere que la privacidad debe integrarse en el desarrollo de sistemas y tecnologías desde el inicio. Para poder implementar la privacidad por diseño en la detección de objetos, se deben seguir unos principios básicos:

- **Minimizar la Recolección de Datos:** Para conseguir esto, se debe recolectar solo la información necesaria para el propósito específico del sistema de detección de objetos. Por ejemplo, en lugar de almacenar todas las imágenes capturadas, los sistemas podrían procesar las imágenes en tiempo real y descartar aquellas que no son relevantes para el objetivo del sistema.
- **Anonimización y Encriptación:** Cavoukian destaca que los datos recolectados deben ser adecuadamente anonimizados y encriptados para así proteger la identidad de los individuos. Podrían servir técnicas como el desenfoque de rostros o la eliminación de metadatos que pudiesen identificar a las personas.
- **Transparencia y Consentimiento:** Para lograr dicho principio, es crucial informar a las personas sobre la recolección de datos y obtener su consentimiento explícito. Esto se lograría mediante la implementación de señales claras en áreas donde se utilizan sistemas de detección de objetos. También la implementación de políticas de privacidad accesibles y comprensibles abierto a todos los públicos.
- **Auditorías y Monitoreo:** Realizar auditorías periódicas y monitorear el uso de los sistemas de detección de objetos podría asegurar que se cumplen con

las políticas de privacidad establecidas. Aquí se incluyen revisiones exhaustivas de las prácticas llevadas a cabo para la recolección de datos, almacenamiento y su uso para poder detectar y corregir posibles violaciones de privacidad.

- **Legislación y Regulación:** Por último, se debería tanto promover como adherirse a leyes y regulaciones que protejan la información sensible de los individuos. La implementación de leyes que permitan el uso de la vigilancia masiva y que protejan los derechos de privacidad de las personas, podría ser una manera óptima y efectiva de abordar este problema.

Aunque aplicando el concepto introducido por Ann Cavokian puede ser útil para poder reducir los riesgos en privacidad que las tecnologías de detección de datos ofrecen, se debe destacar que dado que la labor de detección requiere de una precisión extremadamente alta (para poder realizar su función con éxito), siempre se recopilaran datos que puedan infringir alguno de estos principios. Esto implica que para abordar el dilema de la vigilancia de forma práctica y por tanto, reducir al máximo sus riesgos, se debe revisar de forma exhaustiva si bien se cumplen los principios mencionados para así poder introducir nuevos métodos que hagan por lograr el objetivo de una sociedad que no este masivamente vigilada.

4.3.2 Seguridad de los Datos y Acceso No Autorizado

Otro dilema ético que se debe tener en cuenta (y que es crucial) en la detección de detección de objetos es el se refiere a la seguridad de datos. Como se ha visto en el anterior apartado de este análisis ético, la recopilación de datos de una forma descontrolada puede llegar a escenarios de vigilancia masiva y provocar cambios en el comportamiento de las personas. Ciertamente, un problema (que lamentablemente ocurre de forma muy recurrente) que ocurre de un descontrol en la recopilación de datos es la realización de actividades delictivas como el acoso, el robo de identidad o el chantaje [41].

Se puede inferir por tanto que sin la una protección adecuada de los datos recopilados, se puede poner en riesgo no solo la privacidad de las personas, sino que tambien el principal motivo por el que la detección de objetos y personas se sigue desarrollando: la seguridad.

Algunos de los desafíos éticos que conciernen la seguridad de datos se muestran a continuación:

- **Acceso no autorizado:** Las imgenes o videos construidos por las tecnologías de detección, pueden ser vulnerables a accesos no autorizados, tanto por por actores externos como internos [41].

- **Uso malicioso de datos:** Las datos recolectados por los radares o sensores pueden obtener información sensible que puede a su vez ser utilizada de forma maliciosa. Un ejemplo de actividades maliciosas pueden ser el acoso a individuos, el robo de identidad y el chantaje utilizando imágenes o videos comprometedores [41].
- **Responsabilidad y custodia de los datos:** Las organizaciones que recolectan y almacenan datos visuales (como imágenes o videos) deben responsabilizarse de proteger dichos datos. Si no se protegen de forma adecuada, se puede propiciar filtraciones de datos que expongan los datos a malhechores [41].

Para poder abordar de forma correcta este dilema y los problemas derivados que pueden surgir a partir de este, sería útil aplicar el principio de “seguridad de los datos” el cual proviene de la teoría de protección de datos [24]. Este principio recalca la importancia de proteger los datos contra accesos no autorizados y su uso indebido. Para lograr su cumplimiento se proponen varias estrategias:

- **Implementación de medidas de seguridad robustas:** Técnicas avanzadas de cifrado (tanto en tránsito como en reposo), autenticación (de múltiples factores) y control de acceso (como los que están basados en roles), podrían ser algunas de las técnicas que se podrían aplicar para lograr una seguridad más robusta en la gestión de los datos obtenidos por los sistemas de detección de objetos y personas [24].
- **Capacitación y concienciación:** Proporcionar educación y concienciar tanto a desarrolladores como usuarios, sobre manejar datos de forma segura, sobre la importancia de implementar políticas de seguridad de la información podría contribuir a cumplir este principio [24].
- **Monitoreo y auditoría continua:** Como en el anterior dilema comentado, la realización de auditorías y la monitorización de los datos, contribuye a la detección de posibles brechas de seguridad y a responder de forma eficaz a las mismas [24].
- **Planes de respuesta a incidentes:** Para poder responder a incidentes de seguridad tales como brechas o filtraciones de datos es fundamental desarrollar y mantener planes de respuesta a incidentes. Esto serviría para identificar, contener y mitigar dichos incidentes de seguridad [24].
- **Colaboración con reguladores:** También, trabajar en colaboración con agencias reguladoras y de cumplimiento puede asegurar que las prácticas de seguridad de datos estén alineados con la legislación vigente [24].

Tras haber mencionado posibles soluciones al dilema comentado, se debe tener en cuenta que aunque se hayan proporcionado soluciones para minimizar los riesgos que pueden surgir a partir de este dilema ético (la seguridad de datos y el acceso no autorizado), se deben aplicar al mismo tiempo las soluciones y prácticas propuestas para el dilema de la vigilancia masiva de modo que así se evite caer en los problemas derivados de la vigilancia masiva mientras que se combate el problema de la seguridad de datos y el acceso no autorizado a los mismos.

4.3.3 Discriminación algorítmica

En último lugar, la discriminación algorítmica podría aparecer como un dilema ética en labor de detección de objetos y personas. Esto se debe a que dichos sistemas se entrenan con grandes volúmenes de datos que pueden llegar a tener sesgos inherentes. El problema reside en que si los datos de entrenamiento no son representativos o están sesgados, los algoritmos pueden llegar a ignorar las desigualdades sociales, discriminando a ciertos grupos de personas en función de su raza, género u otras características. Un ejemplo que representa lo dicho, es un sistema de detección de objetos entrenando predominante de una etnia particular puede tener dificultades para reconocer correctamente a personas de otras etnias [22].

Lo cierto es que aunque este problema no aparece en tecnologías como las cámaras de visión (dado que son capaces de captar características como los colores presentes en las imágenes, y por lo tanto se verían las diferencias entre distintas etnias o razas), las ondas milimétricas no son capaces de captar dichas diferencias de forma directa. Es por ello que en estos casos, la discriminación algorítmica podría manifestarse de varias formas:

- **Sesgos en los datos de entrenamiento:** Se puede encontrar sesgo en los datos de entrenamiento si los mismos no consiguen representar de forma adecuada la diversidad de la población. Esto podría resultar en algoritmos que funcionan bien para ciertos grupos pero no para otros [22].
- **Prejuicios en el diseño del algoritmo:** De forma inconsciente, los diseñadores de algoritmos pueden incorporar sus propios prejuicios en los modelos que ellos mismos han desarrollado. Lo comentado por tanto podría ocurrir a través de decisiones sobre qué características incluir y como ponderarlas en dichos algoritmos [22].
- **Impacto desproporcionado:** Los algoritmos pueden llegar a tener un impacto desproporcionado en ciertos grupos etnicos si los datos que se emplean en los algoritmos estan sesgados. Esto podría pasar por ejemplo en un sistema de vigilancia que identifica más fácilmente a personas de ciertos orígenes

éticos que a otras distintas. Esto podría dar lugar no solo a problemas de discriminación, sino que también abre las puertas a problemas relacionados con el racismo [22].

Para poder mitigar dicho problema, el marco teórico de la justicia algorítmica, la cual busca asegurar que los algoritmos sean justos y equitativos, puede ser clave. Dicho enfoque implica que se sigan varias estrategias clave [64]:

- **Diversidad en los Datos de Entrenamiento:** Con el fin de que el entrenamiento de algoritmos sea considerado diverso y representativo de la población general es fundamental asegurarse de que la recolección de datos provenga de múltiples fuentes. En adición, añadir información extra como la inclusión de datos relativos a distintas etnias, géneros, edades u otras características, podría contribuir a mejorar en este aspecto.
- **Evaluación Continua de Sesgos:** La implementación de procedimientos de evaluación y corrección de sesgos en los algoritmos que se empleen para la detección de objetos puede ayudar a analizar en profundidad los datos obtenidos y determinar si bien hay sesgos y en que medida pueden llegar a influir en los resultados finales. Emplear técnicas de aprendizaje automático puede ayudar en esta labor.
- **Inclusión de Perspectivas Diversas:** La inclusión de individuos de diversos antecedentes y experiencias en el desarrollo y evaluación de sistemas de detección de objetos puede ayudar a identificar y abordar posibles sesgos que de otra manera podrían pasarse por alto. En adición, la colaboración con organizaciones que defienden los derechos de grupos minoritarios puede proporcionar perspectivas valiosas sobre cómo estos sistemas pueden llegar a impactar a distintas comunidades.
- **Transparencia y Rendición de Cuentas:** La divulgación de datos y metodologías utilizadas para entrenar los algoritmos mediante la intervención de mecanismos para rendir cuentas en caso de que se detecten casos de sesgos o incluso discriminación, puede contribuir a que el proceso de detección no solo sea inclusivo sino que también transparente y seguro.
- **Regulaciones y Normativas:** Por último, promover el desarrollo de regulaciones y normativas que exijan la evaluación de equidad en los algoritmos es fundamental para que no ocurran los problemas comentados. Aquí se incluyen regulaciones que incluyan requisitos para pruebas de sesgos, informes de impact y auditorías independientes que analicen si bien los resultados obtenidos por parte de los algoritmos no propician problemas si bien de sesgo o de discriminación.

Este problema aunque no tenga relación directa con la privacidad de datos, se podrían encontrar implicaciones éticas sobre el tema a la hora de hacer que dichos algoritmos no caigan en la discriminación algorítmica. Esto se debe principalmente a que para que estos algoritmos no caigan en el problema descrito deben contener una cantidad de información que sea lo suficientemente grande y diversa para que represente de la mejor forma a la población general. Debido a esto, se podría caer en alguno de los dilemas mencionados anteriormente dado que si se proporcionan demasiados datos, podría propiciar escenarios bien de vigilancia masiva o de acceso a datos no autorizados. Es por ello que al mismo tiempo, este problema debería revisar las soluciones propuestas para los dilemas mencionados con el fin de que no solo de minimizen los riesgos de discriminación algorítmica, sino que al mismo tiempo evitar problemas de vigilancia masiva y seguridad en los datos.

Queda claro que aunque la detección de objetos ofrezca beneficios significativos, hay una serie de dilemas éticos derivados de esta labor que deben ser abordados con un enfoque multidimensional que combine principios teóricos con prácticas concretas. Es por ello que si se implementan la privacidad por diseño, la seguridad de los datos y la justicia algorítmica en el desarrollo de estas tecnologías, podría contribuir a maximizar los beneficios de estas tecnologías mientras se minimizan los riesgos y se protegen los derechos fundamentales de las personas (como es el caso del derecho a la privacidad).

Además de la implementación de colaboraciones entre los desarrolladores, legisladores y usuarios, que permitan un desarrollo seguro de este tipo de tecnologías y que no caigan en los problemas comentados de discriminación algorítmica, es fundamental la educación y la concienciación por parte de la sociedad sobre estos dilemas éticos para que el futuro de estas tecnologías proteja de verdad a las personas de la misma.

Capítulo 5

Discusiones Finales

A lo largo de esta memoria se ha investigado sobre la detección de objetos empleando sensores de ondas milimétricas. Al principio de la memoria se hizo un estudio evolutivo que profundizó sobre cómo las tecnologías de detección de objetos han ido evolucionando a medida que la tecnología ha evolucionado. Se vio cómo en dicho desarrollo de estas tecnologías, las que estaban basadas en ondas del espectro visible obtenían buenos resultados de detección en condiciones de visibilidad óptimas. En cambio, cuando dichas tecnologías eran implementadas en entornos donde dichas condiciones no eran óptimas, la capacidad de detección se veía drásticamente reducida. Es por ello que desde entonces, el estudio de nuevas tecnologías que puedan superar principalmente esta limitación que las tecnologías basadas en el espectro visible ha ido incrementando. Gracias al estudio comparativo mostrado en el capítulo de introducción de esta memoria, se comprobó cómo los sistemas que implementaban tecnologías de detección de ondas milimétricas superaban a las tecnologías convencionales (las basadas en el espectro visible).

En adición, en el estudio del arte se encontraron las tecnologías que se han estado utilizando en este ámbito (como sensores que han sido habitualmente utilizados en el ámbito de detección de ondas milimétricas) con efectividad (alcanzando como se bien en dicho capítulo, niveles de precisión esperanzadores). En adición, se descubrió que la técnica de Vision-Fusion es la técnica empleada que mejores resultados da a la hora de detectar objetos, mediante el empleo de ondas milimétricas. Dicha técnica emplea la combinación de cámaras de fusión con sensores de ondas milimétricas. Los datos provenientes de dichos sensores se fusionan en un proceso avanzado de procesamiento de datos (como se vio en la figura 2.1). En adición, hizo una exploración sobre los modelos de detección más avanzados y con los que se obtenían resultados prometedores: PointNet, SECOND y VoxelNet. Dicha exploración puede verse resumida en la tabla 2.6.

Posteriormente, gracias al estudio exhaustivo del arte mostrado en el segundo capítulo de esta memoria, se consiguió desarrollar una metodología basada en la técnica de Vision-Fusion (ver Figura 3.1). Esta metodología se componía de tres fases:

La primera fase constaba del escaneo y obtención de datos para cada uno de los escenarios propuestos para este trabajo: en condiciones de visibilidad óptima y nula, detección de movimiento de personas y objetos (movimiento de persona sin objetos de por medio, movimiento de persona con una silla de por medio y movimiento de persona con una mesa de por medio). Para dicha fase, se empleó el sensor IWR6438OPEVM, el cual se eligió debido a que tenía características similares a los sensores encontrados en el capítulo del estado del arte, y además era óptimo para aplicaciones de detección portables (debido a su bajo consumo).

La segunda fase consistió en realizar un procesamiento de datos exhaustivo para poder obtener un conjunto de datos tanto robusto como representativo. La primera técnica consistió en conseguir hacer legibles los datos obtenidos por el sensor (ya que dichos datos venían formateados en binario). La segunda técnica consistió en hacer una conversión de los datos para que pudiesen representar puntos en el espacio (x,y,z) . La última técnica empleada, consistió en limpiar los datos de aquellos que no fuesen representativos y útiles para el estudio. Para ello, se aplicaron las explicaciones provistas por el manual de especificaciones del sensor empleado y se recorrieron fila a fila los datos obtenidos para así poder evitar la corrupción de los mismos. En adición, se fusionaron los datos obtenidos en cada uno de los escenarios para así poder hacer el conjunto de datos lo más representativo posible.

La tercera y última fase, consistió en aplicar el algoritmo PointNet, el cual se empleó gracias al estudio comparativo hecho en el capítulo del estado del arte (que demostró que era más útil en aplicaciones que involucraban el empleo de un solo sensor de ondas milimétricas). En adición, en esta fase se preparó con el código necesario la implementación del modelo PointNet (con la creación de los conjuntos de entrenamiento, validación y test). Dado que dicho modelo requiere que los datos obtenidos tengan etiquetas identificativas para cada uno de los Point-Clouds, se aplicó en primer lugar un modelo no supervisado (KNN), para que dicho modelo identificase según la tipología de los datos obtenidos distintas etiquetas. Una vez aplicado el modelo KNN se procedería a la aplicación del mejor modelo PointNet (que se obtuvo mediante la búsqueda de los mejores parámetros del mismo).

En el capítulo de análisis, se estudiaron los resultados obtenidos en cada uno de los escenarios provistos. Para dicha labor se emplearon tres elementos: una representación 3D resultante de la aplicación del modelo PointNet, la gráfica de pérdida de información del modelo y la matriz de confusión resultante del proceso de clasificación del modelo empleado. Aquí se descubrió que de forma generalizada la metodología empleada ha sido óptima y se han podido obtener resultados excelentes a la hora de identificar el movimiento de personas en distintos escenarios, en distintas condiciones de luminosidad. De esta forma se ha podido probar la robustez de las tecnologías de detección que emplean ondas milimétricas en distintas condiciones.

En adición, en dicho capítulo se analizaron las implicaciones éticas del desarrollo de tecnologías de detección, encontrando aquí tres problemas fundamentales: Vigilancia masiva, seguridad de los datos y acceso no autorizado, y el problema de discriminación algorítmica. Para poder minimizar los riesgos de caer en las tres situaciones, se propusieron soluciones conjuntas de transparencia y minimización de los datos, o colaboración con organismos reguladores para poder establecer políticas que regulen el desarrollo de estas tecnologías con el objetivo de no solo hacerlas más seguras, si no que también hacerlas más inclusivas y representativas de la población general.

Tras haber resumido el recorrido de este trabajo, a continuación se van a establecer las conclusiones finales del estudio realizado.

5.1 Conclusiones del estudio

Este trabajo tiene como objetivos principales: Reconocimiento de movimiento de personas sin ningún objeto de por medio, reconocimiento de movimiento de personas con una silla de por medio y reconocimiento de movimiento de personas con una mesa de por medio, en condiciones de luminosidad óptimas y nulas y con cierto nivel de precisión; y el diseño de una metodología óptima para realizar la labor de detección de objetos con éxito.

Tras el capítulo de análisis por tanto se puede concluir lo siguiente:

- **Reconocimiento de movimiento de personas sin ningún objeto de por medio:** Para este escenario en condiciones luminosas óptimas se obtuvo una representación clara del movimiento lineal realizado por la persona con una precisión global del 98%. En adición, en condiciones luminosas nulas se

obtuvo un resultado muy similar tanto en la representación visual (que se observa claramente el movimiento lineal realizado por la persona) con una precisión global del 98.5 %. Es por ello por lo que con los resultados obtenidos, este primer objetivo propuesto para este trabajo se puede dar por satisfecho en ambas condiciones de luminosidad.

- **Reconocimiento de movimiento de personas con una silla de por medio:** Para este escenario tanto en condiciones de luminosidad óptima como en condiciones nulas, se encontraron unos resultados visuales muy similares que explicaban de forma clara el movimiento lineal realizado por la persona con la silla hacia el radar, alcanzando valores de precisión global del 96.4 % y del 93.7 % respectivamente. Es por ello por lo que con los resultados obtenidos, este segundo objetivo también puede darse por satisfecho en ambas condiciones de luminosidad.
- **Reconocimiento de movimiento de personas con una mesa de por medio:** Para este escenario se encontraron excelentes resultados visuales (que mostraban con claridad el movimiento lineal realizado por la persona con la mesa hacia el radar) alcanzando valores de precisión en condiciones luminosas óptimas de 93.9 %, y en condiciones de luminosidad nulas se obtuvo una precisión del 95.73 %. Por este motivo, este tercer objetivo principal se puede dar por satisfecho con éxito.
- **Diseño de una metodología óptima para la detección de objetos:** Como bien se descubrió en el capítulo de estado del arte, hay una metodología basada en la fusión de datos que se emplea universalmente para la labor de detección de objetos mediante el uso complementario de tecnologías de detección que emplean ondas milimétricas y las que emplean ondas del espectro visible como cámaras de visión [44]. Gracias a esta metodología, se pudo diseñar la planteada en el capítulo de metodología. En adición a lo anterior, dado que los tres objetivos de detección planteados se han satisfecho con éxito, se puede concluir que la metodología diseñada para este trabajo ha sido efectiva y por lo tanto óptima, por lo que este cuarto y último objetivo principal planteado para este trabajo, se puede dar por satisfecho.

Además de haberse cumplido los cuatro objetivos principales propuestos para este trabajo, gracias a la investigación realizada se ha podido verificar la efectividad que tienen las ondas milimétricas a la hora de ser capaces de dar resultados consistentes y robustos no solo en condiciones de visibilidad óptimas, sino que también en condiciones de luminosidad nulas. Como se ha podido observar, los resultados de los tres escenarios propuestos en ambas condiciones de visibilidad, no solo son muy similares si no que muestran un rendimiento excelente y una capacidad explicativa

muy alta mediante el uso del sistema diseñado. Es por ello que este trabajo también ha servido para demostrar la ventaja que ofrece el uso de tecnologías de detección que empleen ondas milimétricas (que funcionan de forma consistente y eficaz en condiciones de visibilidad variables), frente a otras tecnologías tradicionales que se basan en el empleo de ondas del espectro visible (y que no son capaces de dar buenos resultados en ambientes de luminosidad nula).

En adición, en este estudio se consiguió identificar con éxito las implicaciones éticas y la importancia que tiene la ética en el desarrollo de tecnologías de detección de objetos. Esto se consiguió mediante la identificación de técnicas que deben ser empleadas para minimizar los riesgos en esta labor: aplicar la denominada “seguridad por diseño” desarrollada por Ann Cavokin, la aplicación del principio de “seguridad de los datos” y la propuesta de estrategias para abordar el problema de la discriminación algorítmica.

De todos modos, aún siendo los resultados técnicos excelentes y aunque se hayan cumplido los objetivos principales propuestos para este trabajo, hay una serie de limitaciones que deben ser comentadas:

- **Capacidad nula de detectar objetos estaticos:** Esta limitación con la que contaba este trabajo, hizo que se hiciera la asunción para darle simplicidad a la labor de detección, que a la hora de testear los escenarios que involucraban la detección de personas con objetos de por medio, se decidió mover al mismo tiempo que la persona realizaba el desplazamiento provisto, mover los objetos en cuestión. Esto ha podido causar que (como ya se ha comentado en el capítulo de análisis) que no se hayan podido identificar como en el escenario 1, tres fases distintas de movimiento sino que dos.
- **Inconsistencia a la hora de asignar clases:** Como se ha podido observar, las representaciones visuales obtenidos tras el proceso de detección, identifican una serie de clases que representan distintas fases del movimiento realizado por la persona (bien sea con objetos o sin ningún objeto de por medio). Si bien esto es cierto, y ayuda a comprender el movimiento realizado en cada uno de los escenarios, se puede apreciar como en cada representación visual, no hay consistencia a la hora de asignar una clase a cada fase del movimiento. Aunque esto no quite capacidad explicativa del modelo, es cierto que el modelo aunque identifique que existen distintos tipos de clases para distintas fases del movimiento, no es capaz de ser consistente a la hora de asignar una etiqueta consistente para datos que pertenezcan a una fase concreta del movimiento.
- **Incapacidad de identificación de objetos/ personas:** Aunque los

objetivos principales de este trabajo se hayan cumplido, es cierto que el modelo desarrollado aún no es capaz de identificar y clasificar los objetos y personas que se estaban moviendo en los escenarios probados en este caso. Esto, aunque no fuese uno de los objetivos principales del trabajo, es cierto que a la hora de aplicar dicho modelo a un escenario real como de conducción autónoma, o en aplicaciones en cámaras de seguridad, no solo es fundamental saber que movimientos se realizan en un entorno concreto, sino que también ser capaz de identificar que objetos o personas están presentes en el mismo.

- **Introducción de una variable de ruido por parte del sensor:** Como ya se ha comentado a lo largo del capítulo de metodología, el radar introduce una variable de ruido para intentar mapear de forma aproximada e intuir donde se puede encontrar un objeto o persona después de haber parado el movimiento. Si bien es cierto que en el escenario 1, esta variable de ruido que introduce el radar, sirve para poder identificar claramente el momento cuando la persona se para para detener la simulación. En cambio, en el resto de escenarios, aunque se puede intuir que los puntos más dispersos corresponden a esta última fase del movimiento que aparece reflejada en las representaciones del primer escenario, es posible que dicho ruido que introduce el radar, confunda al modelo empleado y que por tanto se pierda capacidad explicativa.
- **Desbalanceo en las instancias de cada clase identificada:** Como se ha podido apreciar en cada uno de los resultados de la matriz de confusión, claramente aparece una clase minoritaria y otra que ocupa una gran parte de las instancias detectadas por clase. Esto puede dar lugar a problemas de sobreajuste que afecten a la capacidad explicativa del modelo empleado. Esto se debe a que si una gran parte de las instancias detectadas por clase están focalizadas en una sola clase, esto puede sesgar los resultados ya que podrían basarse en mayor medida en esta clase mayoritaria. De todos modos, como bien se explico en la parte de análisis, las métricas tanto macro como ponderadas son igual de similares como de excelentes, por lo que aparentemente esto no ha afectado a la capacidad explicativa del momento. Aún así, este es un factor que debe tenerse en cuenta para poder evitar que el modelo clasifique de forma correcta instancias en una clase que no le corresponde. En otras palabras, esta es una limitación que puede hacer que los falsos positivos incrementen y que por tanto, aunque la precisión no cambie, esto podría dañar la robustez del modelo empleado.

5.1.1 Posibles mejoras y siguientes pasos

Tras haber determinado que todos los objetivos principales de este trabajo se han cumplido y tras haber comentado algunas de las limitaciones que presenta este

trabajo, es fundamental detallar los siguientes pasos a seguir para no solo superar las limitaciones comentadas, sino para poder contribuir de forma lógica y coherente al desarrollo de un sistema eficaz y óptimo tomando como base todo lo realizado en este trabajo:

Implementar Metodología SAR [9]

Una de las limitaciones del estudio actual es la incapacidad de detectar objetos estáticos. Debido a esto, se decidió que en los escenarios que implicaban la detección a través de objetos (en este caso a través de una silla y a través de una mesa), mover los objetos al mismo tiempo que se movía la persona entorno al radar. Esto ha podido causar que se mezclen los puntos detectados y que por tanto a la hora de mapear los dichos, no se consiga una representación precisa. Una técnica que ha sido de utilidad para poder detectar objetos tanto dinámicos como estáticos es la denominada “Synthetic Aperture Radar (SAR)”.

Esta técnica emplea integraciones de metodologías multi-ángulo y amplio ángulo que mejoran la caracterización de los objetivos tanto en movimiento como estáticos. Estos enfoques permiten compensar de manera más sencilla los errores de movimiento y fusionar los datos de manera más rápida y precisa, proporcionando una alta resolución y sin deformaciones.

Además, la técnica SAR aplica algoritmos avanzados de estimación y compensación de parámetros Doppler, lo que permitiría un enfoque más preciso de los datos más precisos obtenidos por el radar, incluso en entornos que tengan una geometría más compleja.

En adición, la aplicación de SAR podría optimizar el procesamiento de los datos adquiridos utilizando métodos que no requieren suposiciones de dispersión isotrópica [40], como los residuos de la compresión de mínimos cuadrados (LS-CS), que mejoran la estimación de la dispersión dependiente del ángulo al que se encuentran los objetivos que capta el radar.

Aplicación de técnicas de “Computer-Vision” para el mapeo de datos

Alguna de las limitaciones que se han comentado en la anterior sección era la inconsistencia que presenta el sistema diseñado en este trabajo a la hora de asignar clases y la incapacidad del mismo para identificar de forma precisa objetos y personas en un entorno. Lo cierto es que esta labor se puede llegar a conseguir aplicando técnicas de “Computer-Vision”.

“Computer-Vision” es un campo interdisciplinario de la informática que se centra en permitir que las máquinas y los sistemas comprendan e interpreten el contenido del mundo visual, como imágenes y videos, de una manera similar a la percepción

humana. Esto implica el desarrollo de algoritmos y modelos que puedan analizar y extraer información útil de datos visuales. Lo cierto es que tradicionalmente estas técnicas se han aplicado sobre imágenes que se han obtenido de tecnologías que emplean el espectro visible como cámaras RGB, dado que es más sencillo extraer características de ellas.

Aún así, dado que se ha probado en este trabajo la efectividad del sistema diseñado basado en ondas milimétricas, en condiciones de luminosidad variable, y pudiendo establecer que dicho sistema presenta innumerables oportunidades para el desarrollo de tecnologías basadas en ondas milimétricas en el futuro, si se pudiesen aplicar técnicas de “Computer-Vision” en este tipo de sistemas, no solo se lograría una consistencia a la hora de asignar clases (dado que se obtendrían características concretas de los puntos recogidos por el radar y se encontraría por tanto un patrón) sino que se podrían obtener formas concretas sobre la persona y los objetos que se estén analizando, mediante el uso de las técnicas adecuadas.

Algún ejemplo de técnica puede ser la técnica de reconstrucción de superficies de Poisson [38]. La reconstrucción de superficies Poisson convierte nubes de puntos en superficies 3D suaves y continuas utilizando ecuaciones de Poisson. El proceso incluye estimar las normales de los puntos, construir un campo de gradiente, resolver la ecuación de Poisson para obtener una función implícita, y extraer la superficie con algoritmos como Marching Cubes. Es robusto frente al ruido y maneja topologías complejas (lo que podría ser útil para reconstruir formas irregulares como las que pueden conformar a una persona).

Otro ejemplo de técnica puede ser NUBRS (Non-Uniform Rational B-Splines) [17]. Esta técnica se aplica en gráficos 3D para representar curvas y superficies con precisión y flexibilidad. Utiliza puntos de control y pesos para definir formas complejas, combinados mediante funciones B-Spline. Ofrece flexibilidad, precisión y es ampliamente utilizada en construcciones 3D.

Lo cierto es que ninguna de estas técnicas han sido provadas en un ámbito similar como el que ha planteado todo este trabajo, pero lo cierto es que en el ámbito de reconstrucción de superficies, los resultados son excelentes y en futuros trabajos se podría investigar sobre la aplicación de alguna de estas técnicas para poder reconstruir los escenarios en donde se aplique un sensor de ondas milimétricas.

Introducción de otro sensor en la metodología diseñada

Otra de las limitaciones comentadas en la anterior sección, es que el sensor empleado implementa una variable de ruido proporcional a los puntos recogidos previamente

por el mismo, con el fin de poder mapear los objetos o personas que se encuentren en el rango de radar, una vez estos no estén en movimiento. Lo cierto es que hay estudios que para poder utilizar ese ruido inherente que introduce el radar en los datos a su favor, se emplea un radar adicional en una posición estratégica, con el fin de luego, aplicar técnicas de fusión de datos que permitan mapear de incluso mejor forma ese ruido. De esta forma se podría evitar la dispersión de algunos puntos, y contribuir a que el sistema mejore su capacidad de detección [48].

Aplicación de la técnica SMOTE [4]

SMOTE (Synthetic Minority Over-sampling Technique) es una técnica utilizada para reducir el desbalanceo de clases en conjuntos de datos de aprendizaje automático. Funciona generando ejemplos sintéticos de la clase minoritaria en lugar de simplemente duplicar instancias existentes. Esto se logra mediante la creación de nuevos puntos de datos que son interpolaciones entre los puntos de datos de la clase minoritaria seleccionados aleatoriamente. El proceso comienza eligiendo una instancia de la clase minoritaria y sus vecinos más cercanos. Luego, se selecciona uno de estos vecinos y se genera un nuevo punto de datos en el espacio entre la instancia original y el vecino, determinado por una interpolación aleatoria. Este proceso se repite hasta que se alcanza el número deseado de ejemplos sintéticos, equilibrando así el número de instancias entre las clases.

Aplicar SMOTE puede mejorar significativamente el rendimiento del modelo desarrollado en varias formas. Primero, al equilibrar el conjunto de datos, el modelo no estará sesgado hacia la clase mayoritaria, lo que puede resultar en una mejor capacidad de generalización y una mayor precisión para la clase minoritaria. Además, la técnica de SMOTE puede ayudar a prevenir problemas de sobreajuste que podrían ocurrir cuando se duplican las instancias de la clase minoritaria, proporcionando en su lugar ejemplos más variados y representativos. Esto, a su vez, mejora las métricas de rendimiento como el recall y el F1-score para la clase minoritaria, asegurando que el modelo sea más robusto y confiable al identificar todas las clases presentes en los datos (que como se ha observado, en todos los escenarios provados, ambas métricas deben ser mejoradas en las clases minoritarias detectadas por el radar).

5.2 Trabajos futuros

Para poder terminar de entender las implicaciones de este trabajo en otras áreas (y también para poder satisfacer el segundo de los objetivos secundarios propuestos para este trabajo), en esta sección se tratará de introducir tres áreas donde el trabajo

de detección de objetos con ondas milimétricas podría ser de utilidad: el ámbito de la medicina, del medioambiente, y para el desarrollo de tecnologías biométricas.

En primer lugar, en el ámbito de la medicina el uso de ondas milimétricas cada vez parece ser una solución para múltiples ámbitos debido a la capacidad de las ondas milimétricas de penetrar materiales no metálicos y al mismo tiempo permitir una visualización clara y detallada de estructuras internas dado que el empleo de este tipo de ondas no implica el uso de radiación ionizante, lo que aumenta la seguridad de los pacientes. Un ejemplo puede ser la aplicación prometedora de este tipo de ondas en el campo de las imágenes médicas avanzadas, dado que estas ondas podrían producir imágenes de alta resolución de tejidos blancos y estructuras internas. Esto mejoraría significativamente la precisión del diagnóstico médico [27]. Además, hoy en día las tecnologías de detección de objetos que emplean ondas milimétricas se están aplicando en la detección no invasiva de cáncer de piel, donde las ondas milimétricas son capaces de identificar irregularidades en la piel indicativas de cáncer [11].

En adición, en el futuro las implicaciones que tiene este tipo de tecnología son vastas. Estas aplicaciones podrían ir desde el monitoreo de pacientes en tiempo real; donde las ondas milimétricas pueden proporcionar datos continuos, lo que eventualmente permitiría una vigilancia más precisa y una respuesta rápida a cambios en el estado de salud del paciente [2]; hasta para dirigir terapias específicas a áreas afectadas del cuerpo, minimizando el daño a los tejidos circundantes y aumentando la eficacia del tratamiento [21].

En segundo lugar, en el ámbito del medioambiente, dada la capacidad de las tecnologías que emplean ondas milimétricas para el monitoreo preciso y en tiempo real de datos, dichas tecnologías se podrían emplear para el monitoreo de la calidad del aire y la cantidad de contaminantes atmosféricos con alta precisión [15]. Esto sería crucial para evaluar el impacto de la contaminación en la salud pública y el medio ambiente. Además, estas ondas se utilizan para detectar y mapear derrames de petróleo en el agua, proporcionando datos en tiempo real esenciales para una respuesta rápida y mitigación de daños ambientales [7].

En el seguimiento del cambio climático, las ondas milimétricas podrían ser útiles para detectar variables atmosféricas clave como temperatura, humedad y concentración de gases de efecto invernadero, lo que ayudaría a que los modelos climáticos actuales fuesen mucho más precisos [14]. En adición, este tipo de tecnologías se está estudiando para aplicarlas para monitorear la salud de los ecosistemas, detectando cambios en la biomasa forestal y la humedad del suelo, vitales para la conservación y gestión sostenible de los recursos naturales [51].

En último lugar, la implicación de tecnologías que empleen ondas milimétricas podría mejorar la industria de las tecnologías biométricas. En el ámbito de la seguridad, las ondas milimétricas se utilizan en escáneres para detectar objetos ocultos en aeropuertos y otro tipo de instalaciones, ofreciendo una detección precisa sin comprometer la privacidad [1]. También, la aplicación de este tipo de tecnologías pueden mejorar los sistemas de reconocimiento facial al proporcionar datos tridimensionales precisos del rostro, incluso en condiciones de baja visibilidad [74]. En el futuro, las ondas milimétricas podrían facilitar la verificación de identidad sin contacto en accesos a edificios, pagos electrónicos y dispositivos personales, mejorando la seguridad y conveniencia [78]. También permitirán medir signos vitales como la respiración y el ritmo cardíaco sin contacto, proporcionando datos esenciales para aplicaciones de salud y seguridad [56].

Tras esta investigación se ha podido comprobar las posibles implicaciones de un desarrollo correcto de tecnologías que empleen ondas milimétricas y por tanto, se puede dar por cumplido el último de los objetivos secundarios propuestos para este trabajo. De todos modos, cabe destacar que aunque el desarrollo de este tipo de tecnologías presenta una gran cantidad de oportunidades en ámbitos como los que se han comentado, es necesario tener en cuenta que para que estas tecnologías sean un éxito, deben regirse a regulaciones y principios éticos para que no se caiga en los problemas descubiertos en el análisis ético de este trabajo.

5.3 Últimas palabras

Tras la realización de este trabajo, se pretende que el mismo sirva para futuros investigadores como una base sólida para explorar y expandir el uso de tecnologías basadas en ondas milimétricas. Los hallazgos y las metodologías desarrolladas aquí pueden ser utilizados como puntos de referencia y orientación en investigaciones posteriores. Además, se espera que los objetivos alcanzados y las limitaciones identificadas proporcionen una comprensión más clara de las capacidades y desafíos asociados con estas tecnologías, fomentando así innovaciones y mejoras continuas en áreas como la medicina, el medioambiente y la biometría. En definitiva, este trabajo aspira a contribuir significativamente al avance del conocimiento y a la aplicación práctica de las tecnologías de ondas milimétricas, impulsando su desarrollo y su integración en diversas disciplinas científicas y técnicas.

Bibliografía

- [1] Abdullah G. Alharbi et al. "Design and Study of a Miniaturized Millimeter Wave Array Antenna for Wireless Body Area Network". En: International Journal of Antennas and Propagation 2022 (2022). ISSN: 16875877. DOI: 10.1155/2022/1736377.
- [2] Abdullah K. Alhazmi et al. "Intelligent Millimeter-Wave System for Human Activity Monitoring for Telemedicine". En: Sensors 24.1 (2024). ISSN: 14248220. DOI: 10.3390/s24010268.
- [3] Tejas Anvekar et al. "VG-VAE: A Venatus Geometry Point-Cloud Variational Auto-Encoder". En: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. Vol. 2022-June. 2022. DOI: 10.1109/CVPRW56347.2022.00336.
- [4] Ahmed Arafa et al. "RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification". En: Journal of King Saud University - Computer and Information Sciences 34.8 (2022). ISSN: 22131248. DOI: 10.1016/j.jksuci.2022.06.005.
- [5] ARS 408-21 Long Range Radar Sensor 77 GHz Continental Engineering Services. Inf. téc. URL: www.conti-engineering.com.
- [6] AWR1642 Single-Chip 77-and 79-GHz FMCW Radar sensor. Inf. téc. 2023. URL: www.ti.com.
- [7] P.P. Beskid et al. "Results of research in the field of remote sensing methods for detecting oil pollution on the water surface conducted at the RSHU". En: HYDROMETEOROLOGY AND ECOLOGY. PROCEEDINGS OF THE RUSSIAN STATE HYDROMETEOROLOGICAL UNIVERSITY 60 (2020). DOI: 10.33933/2074-2762-2020-60-371-391.
- [8] Angona Biswas, Sabrina Abedin y Md. Ahasan Kabir. "Moving Object Detection Using Ultrasonic Radar with Proper Distance, Direction, and Object Shape Analysis". En: Journal of Information Systems Engineering and Business Intelligence 6.2 (2020). ISSN: 2598-6333. DOI: 10.20473/jisebi.6.2.99-111.
- [9] Fabio Bovenga. Special issue "synthetic aperture radar (SAR) techniques and applications". 2020. DOI: 10.3390/s20071851.
- [10] Ann Cavoukian. "Privacy by design: the definitive workshop. A foreword by Ann Cavoukian, Ph.D". En: Identity in the Information Society 3.2 (2010). DOI: 10.1007/s12394-010-0062-y.
- [11] N. Chahat et al. "Human skin-equivalent phantom for on-body antenna measurements in 60GHz band". En: Electronics Letters 48.2 (2012). ISSN: 00135194. DOI: 10.1049/el.2011.2619.
- [12] Jun Chen et al. "Integrating topographic features and patch matching into point cloud restoration for terrain modelling". En: International Journal of Digital Earth 16.2 (2023). ISSN: 17538955. DOI: 10.1080/17538947.2023.2277797.
- [13] Caio Luis Chiariello. "Brasil pandêmico visitado na distopia 1984 de George Orwell". En: Revista Fim do Mundo 03 (2020). ISSN: 2675-3812. DOI: 10.36311/2675-3871.2020.v1n03.p50-64.
- [14] Domenico Cimini, Ed R. Westwater y Albin J. Gasiewski. "Temperature and humidity profiling in the arctic using ground-based millimeter-wave radiometry and 1DVAR". En: IEEE Transactions on Geoscience and Remote Sensing 48.3 PART2 (2010). ISSN: 01962892. DOI: 10.1109/TGRS.2009.2030500.
- [15] David Cuadrado-Calle, Petri Piironen y Natanael Ayllon. "Solid-State Diode Technology for Millimeter and Submillimeter-Wave Remote Sensing Applications: Current Status and Future Trends". En: IEEE Microwave Magazine 23.6 (2022). ISSN: 15579581. DOI: 10.1109/MMM.2022.3155031.
- [16] Yaodong Cui et al. Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review. 2022. DOI: 10.1109/TITS.2020.3023541.
- [17] Anjana Deva Prasad et al. "NURBS-Diff: A Differentiable Programming Module for NURBS". En: CAD Computer Aided Design 146 (2022). ISSN: 00104485. DOI: 10.1016/j.cad.2022.103199.
- [18] Analog Devices. ADAR1000 8 GHz to 16 GHz, 4-Channel, X Band and Ku Band Beamformer (Rev.B). Inf. téc.
- [19] Petr Dvořák y Petr Hurtik. "What Is the Cost of Privacy?" En: Communications in Computer and Information Science. Vol. 1602 CCIS. 2022. DOI: 10.1007/978-3-031-08974-9(_)55.
- [20] Andreas Ellmauthaler et al. "A visible-light and infrared video database for performance evaluation of video/image fusion methods". En: Multidimensional Systems and Signal Processing 30.1 (2019). ISSN: 15730824. DOI: 10.1007/s11045-017-0548-y.
- [21] Katherina P. Farr et al. Imaging Modalities for Early Detection of Pancreatic Cancer: Current State and Future Research Opportunities. 2022. DOI: 10.3390/cancers14102539.
- [22] Maddalena Favaretto, Eva De Clercq y Bernice Simone Elger. "Big Data and discrimination: perils, promises and solutions. A systematic review". En: Journal of Big Data 6.1 (2019). ISSN: 21961115. DOI: 10.1186/s40537-019-0177-4.

- [23] Fermi National Accelerator Laboratory. Why can radio waves pass through a wall but light cannot? Inf. téc. 2014.
- [24] Gina Fisk et al. "Privacy principles for sharing cyber security data". En: Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015. 2015. doi: 10.1109/SPW.2015.23.
- [25] Valéria Garlet et al. "Objetivos de desenvolvimento sustentável - ODS". En: Journal on Innovation and Sustainability RISUS 13.2 (2022). doi: 10.23925/2179-3565.2022v13i2p14-26.
- [26] Yusen Geng et al. "A novel welding path planning method based on point cloud for robotic welding of impeller blades". En: International Journal of Advanced Manufacturing Technology 119.11-12 (2022). issn: 14333015. doi: 10.1007/s00170-021-08573-3.
- [27] M. Ghavami, L. B. Michael y R. Kohno. Ultra Wideband Signals and Systems in Communication Engineering: Second Edition. 2007. doi: 10.1002/9780470060490.
- [28] Bowen Gong et al. "Louvain-Based Traffic Object Detection for Roadside 4D Millimeter-Wave Radar". En: Remote Sensing 16.2 (2024). issn: 20724292. doi: 10.3390/rs16020366.
- [29] Yulan Guo et al. Deep Learning for 3D Point Clouds: A Survey. 2021. doi: 10.1109/TPAMI.2020.3005434.
- [30] Patrick Herbers y Markus König. "Indoor localization for augmented reality devices using BIM, point clouds, and template matching". En: Applied Sciences (Switzerland) 9.20 (2019). issn: 20763417. doi: 10.3390/app9204260.
- [31] Kaixin Hu et al. "Semi-automated Generation of Geometric Digital Twin for Bridge Based on Terrestrial Laser Scanning Data". En: Advances in Civil Engineering 2023 (2023). issn: 16878094. doi: 10.1155/2023/6192001.
- [32] Xu Huang, Joseph K.P. Tsoi y Nitish Patel. "mmWave Radar Sensors Fusion for Indoor Object Detection and Tracking". En: Electronics (Switzerland) 11.14 (2022). issn: 20799292. doi: 10.3390/electronics11142209.
- [33] A. Courville I. Goodfellow Y. Bengio. Deep Learning. The MIT Press. 2016.
- [34] "Infineon-BGT60ATR24C-DataSheet-v01_60-EN". En: ().
- [35] Cesar Iovescu y Sandeep Rao. "The Fundamentals of Millimeter Wave Sensors". En: Texas Instruments (2017). issn: 0019-5022.
- [36] Yuhao Jin et al. "A Hybrid Model for Object Detection Based on Feature-Level Camera-Radar Fusion in Autonomous Driving". En: 2023 8th International Conference on Intelligent Computing and Signal Processing, ICSP 2023. 2023. doi: 10.1109/ICSP58490.2023.10248746.
- [37] Ebi Jose et al. "Predicting millimeter wave radar spectra for autonomous navigation". En: IEEE Sensors Journal 10.5 (2010). issn: 1530437X. doi: 10.1109/JSEN.2009.2037013.
- [38] Michael Kazhdan y Hugues Hoppe. "Screened poisson surface reconstruction". En: ACM Transactions on Graphics 32.3 (2013). issn: 07300301. doi: 10.1145/2487228.2487237.
- [39] Imran Khan Mirani et al. "Object Recognition in Different Lighting Conditions at Various Angles by Deep Learning Method". En: Journal of Xi'an University of Architecture & Technology XIV.1 (2022).
- [40] Woo Tae Kim, Il Suck Koh y Jong Gwan Yook. "3D isotropic dispersion (ID)-FDTD algorithm: Update equation and characteristics analysis". En: IEEE Transactions on Antennas and Propagation 58.4 (2010). issn: 0018926X. doi: 10.1109/TAP.2010.2041311.
- [41] Janet L. Lauritsen. "The future of crime data". En: Criminology 61.2 (2023). issn: 17459125. doi: 10.1111/1745-9125.12330.
- [42] Hao Li et al. "Pedestrian Liveness Detection Based on mmWave Radar and Camera Fusion". En: Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks workshops. Vol. 2022-September. 2022. doi: 10.1109/SECON55815.2022.9918553.
- [43] Shenglin Li y Hwan Sik Yoon. "Sensor Fusion-Based Vehicle Detection and Tracking Using a Single Camera and Radar at a Traffic Intersection". En: Sensors 23.10 (2023). issn: 14248220. doi: 10.3390/s23104888.
- [44] Martin E. Liggins, David L. Hall y James Llinas. Handbook of Multisensor Data Fusion: Theory and Practice: Second Edition. 2008. doi: 10.1201/9781420053098.
- [45] Chien Liang Liu, Tao Hsing Chang y Hsuan Hsun Li. "Clustering documents with labeled and unlabeled documents using fuzzy semi-Kmeans". En: Fuzzy Sets and Systems 221 (2013). issn: 01650114. doi: 10.1016/j.fss.2013.01.004.
- [46] M Livshitz. Tracking radar targets with multiple reflection points. Inf. téc. TI Intern, 2017, págs. 5-39.
- [47] Mariona Carós. PointNet Explained Visually. Abr. de 2022.
- [48] Ana Beatriz Rodrigues Costa De Mattos et al. "Human and Small Animal Detection Using Multiple Millimeter-Wave Radars and Data Fusion: Enabling Safe Applications". En: Sensors 24.6 (mar. de 2024). issn: 1424-8220. doi: 10.3390/s24061901.
- [49] Muhammad Jehanzeb Mirza et al. "Robustness of Object Detectors in Degrading Weather Conditions". En: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. Vol. 2021-September. 2021. doi: 10.1109/ITSC48978.2021.9564505.
- [50] MR3003 High-Performance 77 GHz Radar Transceiver. Inf. téc.
- [51] S. Paloscia, E. Santi y P. Pampaloni. "Radiometric features of land surfaces at 37GHz". En: 2014. doi: 10.1109/ursigass.2014.6929599.
- [52] Lei Pang et al. "Real-time concealed object detection from passive millimeter wave images based on the YOLOV3 algorithm". En: Sensors (Switzerland) 20.6 (2020). issn: 14248220. doi: 10.3390/s20061678.
- [53] Adela Rueda Márquez de la Plata et al. "Protocol development for point clouds, triangulated meshes and parametric model acquisition and integration in an hbim workflow for change control and management in a unesco's world heritage site". En: Sensors (Switzerland) 21.4 (2021). issn: 14248220. doi: 10.3390/s21041083.
- [54] J Ponce y D Forsyth. Computer vision: a modern approach. 2012.

- [55] Charles R. Qi et al. "PointNet: Deep learning on point sets for 3D classification and segmentation". En: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Vol. 2017-January. 2017. DOI: 10.1109/CVPR.2017.16.
- [56] J. Ramkumar et al. "Wearable biometric authentication for health monitoring system using redtacton". En: *International Journal of Advanced Science and Technology* 29.4 (2020). ISSN: 22076360.
- [57] T. S Rappaport et al. *Milimeter Wave Wireless Communications*. Pearson Education, 2013.
- [58] Hermann Rohling y Marc Michael Meinecke. "Waveform design principles for automotive radar systems". En: *CIE International Conference of Radar Proceedings*. 2001. DOI: 10.1109/icr.2001.984612.
- [59] Pascal Housam Salmane et al. "3D Object Detection for Self-Driving Cars Using Video and LiDAR: An Ablation Study". En: *Sensors* 23.6 (2023). ISSN: 14248220. DOI: 10.3390/s23063223.
- [60] Avik Santra, Raghavendran Vagarappan Ulaganathan y Thomas Finke. "Short-range millimetric-wave radar system for occupancy sensing application". En: *IEEE Sensors Letters* 2.3 (2018). ISSN: 24751472. DOI: 10.1109/LSENS.2018.2852263.
- [61] Nicolas Scheiner et al. "Object detection for automotive radar point clouds – a comparison". En: *AI Perspectives* 3.1 (2021). DOI: 10.1186/s42467-021-00012-z.
- [62] Merrill I Skolnik. "Introduction to Radar Systems McGraw-Hill". En: *New York* 19622 (2001).
- [63] Daniel J. Solove. "THE LIMITATIONS OF PRIVACY RIGHTS". En: *Notre Dame Law Review* 98.3 (2023). ISSN: 07453515. DOI: 10.2139/ssrn.4024790.
- [64] Luping Sun y Yanfei Tang. "Data-Driven Discrimination, Perceived Fairness, and Consumer Trust—The Perspective of Consumer Attribution". En: *Frontiers in Psychology* 12 (2021). ISSN: 16641078. DOI: 10.3389/fpsyg.2021.748765.
- [65] Richard Szelisk. *Computer Vision: Algorithms and Applications*. Vol. 42. 2020.
- [66] Chris Urmson et al. "Autonomous driving in urban environments: Boss and the urban challenge". En: *Journal of Field Robotics* 25.8 (2008). ISSN: 15564959. DOI: 10.1002/rob.20255.
- [67] *User's Guide 60GHz mmWave Sensor EVMs*. Inf. téc. 2022. URL: www.ti.com.
- [68] Christian Versloot. "What are max pooling, average pooling, global max pooling and global average pooling, " en: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR (2022)*. ISSN: 15205363.
- [69] Aparna Vyas, Soohwan Yu y Joonki Paik. "Fundamentals of digital image processing". En: *Signals and Communication Technology*. 2018. DOI: 10.1007/978-981-10-7272-7(_).1.
- [70] Safat B. Wali et al. "An Automatic Traffic Sign Detection and Recognition System Based on Colour Segmentation, Shape Matching, and SVM". En: *Mathematical Problems in Engineering* 2015 (2015). ISSN: 15635147. DOI: 10.1155/2015/250461.
- [71] Zhangjing Wang et al. "Research of target detection and classification techniques using millimeter-wave radar and vision sensors". En: *Remote Sensing* 13.6 (mar. de 2021). ISSN: 20724292. DOI: 10.3390/rs13061064.
- [72] Zhiqing Wei et al. *MmWave Radar and Vision Fusion for Object Detection in Autonomous Driving: A Review*. 2022. DOI: 10.3390/s22072542.
- [73] J. Wenger. "Automotive mm-wave radar: Status and trends in system design and technology". En: *IEE Colloquium (Digest)* 230 (1998). ISSN: 09633308. DOI: 10.1049/ic:19980188.
- [74] Weiye Xu et al. "Mask does not matter: Anti-spoofing face authentication using mmwave without on-site registration". En: *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*. 2022. DOI: 10.1145/3495243.3560515.
- [75] Yan Yan, Yuxing Mao y Bo Li. "Second: Sparsely embedded convolutional detection". En: *Sensors (Switzerland)* 18.10 (2018). ISSN: 14248220. DOI: 10.3390/s18103337.
- [76] Zixun Ye et al. "YOLOv7-3D: A Monocular 3D Traffic Object Detection Method from a Roadside Perspective". En: *Applied Sciences (Switzerland)* 13.20 (2023). ISSN: 20763417. DOI: 10.3390/app132011402.
- [77] H Zhang et al. "Real-Time Surveillance Video Synopsis Based on Object Detection and Tracking". En: *Journal of Real-Time Image Processing* (2020), págs. 929-941.
- [78] Jia Zhang et al. "A Survey of mmWave-Based Human Sensing: Technology, Platforms and Applications". En: *IEEE Communications Surveys and Tutorials* 25.4 (2023). ISSN: 1553877X. DOI: 10.1109/COMST.2023.3298300.
- [79] Yin Zhou y Oncel Tuzel. "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection". En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2018. DOI: 10.1109/CVPR.2018.00472.
- [80] Jiannan Zhu, Yixin Yang y Yuwei Cheng. "A Millimeter-Wave Radar-Aided Vision Detection Method for Water Surface Small Object Detection". En: *Journal of Marine Science and Engineering* 11.9 (sep. de 2023). ISSN: 20771312. DOI: 10.3390/jmse11091794.
- [81] David Zimmerman y Robert Buder. "The Invention that Changed the World: How a Small Group of Radar Pioneers Won the Second World War and Launched a Technological Revolution." En: *The Journal of Military History* 61.4 (1997). ISSN: 08993718. DOI: 10.2307/2954124.

Capítulo 6

Apéndices

6.1 Código relevante para el procesamiento de datos

6.1.1 Código para la poder hacer legibles los datos obtenidos por el radar: de .bin a .txt

```
1 from gui_parser import uartParser
2 import sys
3 import os
4
5 def decode_uart(file_path):
6     # Open the binary file in read mode
7     with open(file_path, 'rb') as f:
8         # Read the entire file
9         parser = uartParser()
10        parser.dataCom = f
11        output_path = file_path + ".txt"
12        with open(output_path, 'w') as output_file:
13            while (1):
14                try:
15                    frameData = parser.
readAndParseUartDoubleCOMPortBreak()
16                except:
17                    print("Error reading data.")
18                    break
19                if (len(frameData) > 1):
20                    # print(frameData)
21                    print(frameData, file=output_file)
22                else:
23                    print("End.")
24                    break
```

```

25         output_file.close()
26         f.close()
27
28
29 if __name__ == "__main__":
30     if len(sys.argv) != 2:
31         print("Usage: python decoder.py <source_folder>")
32         sys.exit(1)
33
34     folder_path = sys.argv[1]
35     for x in os.walk(folder_path):
36         print(x[2])
37         for file in x[2]:
38             if file.endswith(".bin"):
39                 file_path = os.path.join(x[0], file)
40                 decode_uart(file_path)
41 # decode_uart(file_path)

```

Listing 6.1: Función completa decode_uart(file_path)

6.1.2 Código empleado para el primer procesado de los datos obtenidos por el radar: Solo coordenadas (x,y,z)

```

1 def parse_data(path):
2     data = []
3     f = open(path)
4     lines = f.readlines()
5     temp = ""
6     invalid = False
7     for line in lines:
8         temp += line
9         if "shape" in line:
10            temp = ""
11            invalid = True
12            continue
13        if '}' in line:
14            if invalid == True:
15                temp = ""
16                invalid = False
17            continue
18        fixed = preprocess(temp)
19        # print(fixed)
20        json_object = json.loads(fixed)
21        points = ast.literal_eval(json_object["pointCloud"])
22        numCoordinates = int(json_object["numDetectedPoints"])
23        coordinates = []
24        count = 0

```

```

25         for i in range(len(points)):
26             if not isinstance(points[i], list):
27                 continue
28             elev, azim, dist = points[i][0], points[i][1],
points[i][3]
29             if not isinstance(elev, float) or not isinstance(
azim, float) or not isinstance(dist, float):
30                 continue
31             x = dist * (math.cos(elev) * math.sin(azim))
32             y = dist * (math.cos(elev) * math.cos(azim))
33             z = dist * math.sin(elev)
34             temp2 = [x, y, z]
35             coordinates.append(temp2)
36             count += 1
37             json_object["coordinates"] = np.array(coordinates)
38             json_object["numDetectedPoints"] = count
39             # print(json_object)
40             if count > 0:
41                 data.append(json_object)
42             temp = ""
43
44     return pd.DataFrame(data)

```

Listing 6.2: Función completa transformación coordenadas

6.1.3 Código empleado para el segundo procesado de los datos obtenidos por el radar: Eliminación de datos erróneos

```

1
2 def preprocess(data):
3     to_return = ""
4     i = 0
5     skip = False
6     while i < len(data):
7         if skip == True and data[i] != '}'':
8             i += 1
9             continue
10        if skip == True and data[i] == '}'':
11            to_return += data[i]
12            skip = False
13            i += 1
14            continue
15        if data[i: i + 5] == "array":
16            i += 6
17            to_return += "\""
18        if data[i: i + 22] == ")", 'numDetectedPoints':

```

```

19         i += 24
20         to_return += "\", \"numDetectedPoints\": "
21         to_return += data[i]
22         while (data[i + 1] != "," and data[i + 1] != "}"):
23             to_return += data[i + 1]
24             i += 1
25         skip = True
26         continue
27     if data[i] == "'":
28         to_return += "\""
29         i += 1
30         continue
31     if data[i] == "\n":
32         i += 1
33         continue
34     to_return += data[i]
35     i += 1
36     return to_return
37
38
39 def parse_all_files_in_folder(folder_path):
40     data_frames = []
41     for root, _, files in os.walk(folder_path):
42         for file_name in files:
43             if file_name.endswith('.txt'):
44                 file_path = os.path.join(root, file_name)
45                 df = parse_data(file_path)
46                 data_frames.append(df)
47     combined_df = pd.concat(data_frames, ignore_index=True)
48
49     return combined_df
50
51 h = plt.hist(sample_data["numDetectedPoints"])
52 plt.title('Histogram of amount of points per number')
53
54
55 #Ensuring all point clouds have same number of points and get a np
56   array
57 num_point_clouds = len(sample_data)
58 pointnet_dataset = []
59
60 #Set Threshold Here
61 threshold = 100
62
63 for i in range(num_point_clouds):
64     num_rows = len(sample_data.loc[i, "coordinates"])
65
66     if threshold - num_rows > 0:
67         # Duplicate points

```

```

67     sampling_indices = np.random.choice(num_rows, threshold -
68     num_rows)
69     new_points = sample_data.loc[i, "coordinates"][
70     sampling_indices, :]
71     pointnet_dataset.append(np.concatenate((sample_data.loc[i,
72     "coordinates"], new_points),axis=0))
73     elif threshold - num_rows < 0:
74         # sample points
75         sampling_indices = np.random.choice(num_rows, threshold)
76         pointnet_dataset.append(sample_data.loc[i, "coordinates")[
77     sampling_indices, :])
78     else:
79         pointnet_dataset.append(sample_data.loc[i, "coordinates"])
80
81 pointnet_df = pd.DataFrame({"coordinates": pointnet_dataset})

```

Listing 6.3: Limpieza y Parseado completo

6.2 Código completo de PointNet

6.2.1 Modelo PointNet y estimación del rendimiento

```

1 import torch
2 import torch.nn as nn
3 import torch.nn.functional as F
4 class TransformationNet(nn.Module):
5
6     def __init__(self, input_dim, output_dim):
7         super(TransformationNet, self).__init__()
8         self.output_dim = output_dim
9
10        self.conv_1 = nn.Conv1d(input_dim, 64, 1)
11        self.conv_2 = nn.Conv1d(64, 128, 1)
12        self.conv_3 = nn.Conv1d(128, 256, 1)
13
14        self.bn_1 = nn.BatchNorm1d(64)
15        self.bn_2 = nn.BatchNorm1d(128)
16        self.bn_3 = nn.BatchNorm1d(256)
17        self.bn_4 = nn.BatchNorm1d(256)
18        self.bn_5 = nn.BatchNorm1d(128)
19
20        self.fc_1 = nn.Linear(256, 256)
21        self.fc_2 = nn.Linear(256, 128)
22        self.fc_3 = nn.Linear(128, self.output_dim*self.output_dim
)

```

```

23
24     def forward(self, x):
25         num_points = x.shape[1]
26         x = x.transpose(2, 1)
27         x = F.relu(self.bn_1(self.conv_1(x)))
28         x = F.relu(self.bn_2(self.conv_2(x)))
29         x = F.relu(self.bn_3(self.conv_3(x)))
30
31         x = nn.MaxPool1d(num_points)(x)
32         x = x.view(-1, 256)
33
34         x = F.relu(self.bn_4(self.fc_1(x)))
35         x = F.relu(self.bn_5(self.fc_2(x)))
36         x = self.fc_3(x)
37
38         identity_matrix = torch.eye(self.output_dim)
39         if torch.cuda.is_available():
40             identity_matrix = identity_matrix.cuda()
41         x = x.view(-1, self.output_dim, self.output_dim) +
identity_matrix
42         return x
43
44
45 class BasePointNet(nn.Module):
46
47     def __init__(self, point_dimension):
48         super(BasePointNet, self).__init__()
49         self.input_transform = TransformationNet(input_dim=
point_dimension, output_dim=point_dimension)
50         self.feature_transform = TransformationNet(input_dim=64,
output_dim=64)
51
52         self.conv_1 = nn.Conv1d(point_dimension, 64, 1)
53         self.conv_2 = nn.Conv1d(64, 64, 1)
54         self.conv_3 = nn.Conv1d(64, 64, 1)
55         self.conv_4 = nn.Conv1d(64, 128, 1)
56         self.conv_5 = nn.Conv1d(128, 256, 1)
57
58         self.bn_1 = nn.BatchNorm1d(64)
59         self.bn_2 = nn.BatchNorm1d(64)
60         self.bn_3 = nn.BatchNorm1d(64)
61         self.bn_4 = nn.BatchNorm1d(128)
62         self.bn_5 = nn.BatchNorm1d(256)
63
64
65     def forward(self, x, plot=False):
66         num_points = x.shape[1]
67
68         input_transform = self.input_transform(x) # T-Net tensor [

```



```

batch, 3, 3]
69     x = torch.bmm(x, input_transform) # Batch matrix-matrix
product
70     x = x.transpose(2, 1)
71     tnet_out=x.cpu().detach().numpy()
72
73     x = F.relu(self.bn_1(self.conv_1(x)))
74     x = F.relu(self.bn_2(self.conv_2(x)))
75     x = x.transpose(2, 1)
76
77     feature_transform = self.feature_transform(x) # T-Net
tensor [batch, 64, 64]
78     x = torch.bmm(x, feature_transform)
79     x = x.transpose(2, 1)
80     x = F.relu(self.bn_3(self.conv_3(x)))
81     x = F.relu(self.bn_4(self.conv_4(x)))
82     x = F.relu(self.bn_5(self.conv_5(x)))
83     x, ix = nn.MaxPool1d(num_points, return_indices=True)(x)
# max-pooling
84     x = x.view(-1, 256) # global feature vector
85
86     return x, feature_transform, tnet_out, ix
87
88
89 class ClassificationPointNet(nn.Module):
90
91     def __init__(self, num_classes, dropout=0.3, point_dimension
=3):
92         super(ClassificationPointNet, self).__init__()
93         self.base_pointnet = BasePointNet(point_dimension=
point_dimension)
94
95         self.fc_1 = nn.Linear(256, 128)
96         self.fc_2 = nn.Linear(128, 64)
97         self.fc_3 = nn.Linear(64, num_classes)
98
99         self.bn_1 = nn.BatchNorm1d(128)
100        self.bn_2 = nn.BatchNorm1d(64)
101
102        self.dropout_1 = nn.Dropout(dropout)
103
104        def forward(self, x):
105            x, feature_transform, tnet_out, ix_maxpool = self.
base_pointnet(x)
106
107            x = F.relu(self.bn_1(self.fc_1(x)))
108            x = F.relu(self.bn_2(self.fc_2(x)))
109            x = self.dropout_1(x)
110

```

```

111         return F.log_softmax(self.fc_3(x), dim=1),
           feature_transform, tnet_out, ix_maxpool
112
113 import torch.optim as optim
114 import torch.nn.functional as F
115 from tqdm.notebook import tqdm
116 import os
117
118 model = ClassificationPointNet(num_classes=1,
119                               point_dimension=3)
120 if torch.cuda.is_available():
121     model.cuda()
122     device = 'cuda'
123 else:
124     device = 'cpu'
125
126
127 optimizer = optim.Adam(model.parameters(), lr=0.001)
128
129 !pip install prettytable
130 from prettytable import PrettyTable
131
132 table = PrettyTable(["Modules", "Parameters"])
133 total_params = 0
134 for name, parameter in model.named_parameters():
135     if not parameter.requires_grad: continue
136     params = parameter.numel()
137     table.add_row([name, params])
138     total_params+=params
139 print(table)
140 print(f"Total Trainable Params: {total_params}")
141
142 import torch.optim as optim
143 import torch.nn.functional as F
144 from tqdm.notebook import tqdm
145 import os
146 # training model
147 epochs=80
148 train_loss = []
149 test_loss = []
150 train_acc = []
151 test_acc = []
152 best_loss= np.inf
153
154 for epoch in tqdm(range(epochs)):
155     epoch_train_loss = []
156     epoch_train_acc = []
157
158     # training loop

```

```

159     for data in train_dataloader:
160         points, targets = data # [batch, num_points, 3] [batch]
161
162         if torch.cuda.is_available():
163             points, targets = points.cuda(), targets.cuda()
164         if points.shape[0] <= 1:
165             continue
166         optimizer.zero_grad()
167         model = model.train()
168         preds, feature_transform, tnet_out, ix_maxpool = model(
points)
169
170         identity = torch.eye(feature_transform.shape[-1])
171         if torch.cuda.is_available():
172             identity = identity.cuda()
173         regularization_loss = torch.norm(
174             identity - torch.bmm(feature_transform,
feature_transform.transpose(2, 1)))
175         # Loss
176         loss = F.nll_loss(preds, targets) + 0.001 *
regularization_loss
177         epoch_train_loss.append(loss.cpu().item())
178         loss.backward()
179         optimizer.step()
180         preds = preds.data.max(1)[1]
181         corrects = preds.eq(targets.data).cpu().sum()
182
183         accuracy = corrects.item() / float(128)
184         epoch_train_acc.append(accuracy)
185
186     epoch_test_loss = []
187     epoch_test_acc = []
188
189     # validation loop
190     for batch_number, data in enumerate(test_dataloader):
191         points, targets = data
192         if torch.cuda.is_available():
193             points, targets = points.cuda(), targets.cuda()
194         model = model.eval()
195         preds, feature_transform, tnet_out, ix = model(points)
196         loss = F.nll_loss(preds, targets)
197         epoch_test_loss.append(loss.cpu().item())
198         preds = preds.data.max(1)[1]
199         corrects = preds.eq(targets.data).cpu().sum()
200         accuracy = corrects.item() / float(128)
201         epoch_test_acc.append(accuracy)
202
203     print('Epoch %s: train loss: %s, val loss: %f, train accuracy:
%s, val accuracy: %f'
```

```

204         % (epoch,
205             round(np.mean(epoch_train_loss), 4),
206             round(np.mean(epoch_test_loss), 4),
207             round(np.mean(epoch_train_acc), 4),
208             round(np.mean(epoch_test_acc), 4)))
209
210     if np.mean(test_loss) < best_loss:
211         state = {
212             'model':model.state_dict(),
213             'optimizer':optimizer.state_dict()
214         }
215         os.makedirs('checkpoints', exist_ok=True)
216
217         torch.save(state, os.path.join('checkpoints', '
pointcloud_checkpoint.pth'))
218         best_loss=np.mean(test_loss)
219
220         train_loss.append(np.mean(epoch_train_loss))
221         test_loss.append(np.mean(epoch_test_loss))
222         train_acc.append(np.mean(epoch_train_acc))
223         test_acc.append(np.mean(epoch_test_acc))
224
225     def plot_losses(train_loss, test_loss):
226         fig = plt.figure()
227         epochs = len(train_loss)
228         plt.plot(range(epochs), train_loss, 'b', label='Training loss'
229 )
230         plt.plot(range(epochs), test_loss, 'r', label='Validation loss
231 ')
232         plt.title('Training and validation loss')
233         plt.legend()
234
235     def plot_accuracies(train_acc, test_acc):
236         fig = plt.figure()
237         epochs = len(train_acc)
238         plt.plot(range(epochs), train_acc, 'b', label='Training
239 accuracy')
240         plt.plot(range(epochs), test_acc, 'r', label='Validation
241 accuracy')
242         plt.title('Training and validation accuracy')
243         plt.legend()
244
245 plot_losses(train_loss, test_loss)
246 plot_accuracies(train_acc, test_acc)

```

Listing 6.4: Implementación modelo

6.2.2 Visualizaciones 3D

```

1 fig = plt.figure(figsize=(10, 7))
2 ax = fig.add_subplot(111, projection='3d')
3 scatter = ax.scatter(reduced_features[:, 0], reduced_features[:,
    1], reduced_features[:, 2], c=pred_labels, cmap='viridis')
4 legend1 = ax.legend(*scatter.legend_elements(), title="Classes")
5 ax.add_artist(legend1)
6 plt.title("Visualizaci n de Point Clouds Clasificados")
7 plt.show()

```

Listing 6.5: Visualizaciones 3D

6.2.3 Matriz de confusión y métricas asociadas

```

1 import torch
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.metrics import confusion_matrix,
    ConfusionMatrixDisplay, classification_report
5 from mpl_toolkits.mplot3d import Axes3D
6
7 def infer(dataset, model, point_cloud_file, shuffle_points=False,
    plot_tNet_out=True, return_indices_maxpool=False):
8     points, label = point_cloud_file
9
10    if torch.cuda.is_available():
11        points = points.cuda()
12        model.cuda()
13
14    points = points.unsqueeze(dim=0)
15    model.eval()
16    preds, feature_transform, tnet_out, ix = model(points)
17    preds = preds.data.max(1)[1]
18
19    points = points.cpu().numpy().squeeze()
20    preds = preds.cpu().numpy()
21
22    if return_indices_maxpool:
23        return preds, tnet_out, ix
24
25    return preds, tnet_out
26
27 from tqdm.notebook import tqdm
28 corrects = 0
29
30 # load checkpoint if available

```

```

31 # model_checkpoint = '/content/checkpoints/3Dmnist_checkpoint_200.
    pth'
32 # if model_checkpoint:
33 #     state = torch.load(model_checkpoint, map_location=torch.
    device(device))
34 #     model.load_state_dict(state['model'])
35
36 for i in tqdm(range(len(test_dataset))):
37     target = test_dataset[i][1]
38     pred, tnet_out = infer(dstensor, model, test_dataset[i],
    shuffle_points=False, plot_tNet_out=False)
39     if target == pred:
40         corrects += 1
41
42 print(f'Accuracy = {corrects/len(test_dataset)}')
43
44 y_true = []
45 y_pred = []
46 with torch.no_grad():
47     for points, targets in test_dataloader:
48         if torch.cuda.is_available():
49             points, targets = points.cuda(), targets.cuda()
50             outputs, _, _, _ = model(points)
51             _, predicted = torch.max(outputs.data, 1)
52             y_true.extend(targets.cpu().numpy())
53             y_pred.extend(predicted.cpu().numpy())
54
55 cm_pointnet = confusion_matrix(y_true, y_pred)
56 disp_pointnet = ConfusionMatrixDisplay(confusion_matrix=
    cm_pointnet, display_labels=np.unique(y_true))
57 disp_pointnet.plot(cmap=plt.cm.Blues)
58 plt.title("Matriz de Confusi n de PointNet")
59 plt.show()
60
61 model.eval()
62 correct = 0
63 total = 0
64 all_points = []
65 all_labels = []
66 all_predicted = []
67
68 with torch.no_grad():
69     for points, targets in test_dataloader:
70         if torch.cuda.is_available():
71             points, targets = points.cuda(), targets.cuda()
72             outputs, _, _, _ = model(points)
73             _, predicted = torch.max(outputs.data, 1)
74             total += targets.size(0)
75             correct += (predicted == targets).sum().item()

```

```

76     all_points.append(points.cpu().numpy())
77     all_labels.append(targets.cpu().numpy())
78     all_predicted.append(predicted.cpu().numpy())
79
80 accuracy_pointnet = 100 * correct / total
81 print(f"Accuracy de PointNet: {accuracy_pointnet}%")
82
83 # Calcular el reporte de clasificaci n
84 report = classification_report(y_true, y_pred, target_names=[f'
      Clase {i}' for i in np.unique(y_true)])
85 print("Reporte de Clasificaci n:")
86 print(report)
87
88 # Convertir los resultados a numpy arrays
89 all_points = np.concatenate(all_points, axis=0)
90 all_labels = np.concatenate(all_labels, axis=0)
91 all_predicted = np.concatenate(all_predicted, axis=0)
92
93 # Dibujar los resultados en 3D por clases predichas
94 fig = plt.figure(figsize=(10, 7))
95 ax = fig.add_subplot(111, projection='3d')
96
97 # Colores diferentes para cada clase
98 colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']
99
100 for class_label in np.unique(all_predicted):
101     class_points = all_points[all_predicted == class_label]
102     ax.scatter(class_points[:, 0, 0], class_points[:, 0, 1],
103               class_points[:, 0, 2], c=colors[class_label % len(colors)],
104               label=f'Clase {class_label}')
105
106 ax.set_xlabel('X')
107 ax.set_ylabel('Y')
108 ax.set_zlabel('Z')
109 plt.title("Resultados de PointNet por Clases en 3D")
110 plt.legend()
111 plt.show()

```

Listing 6.6: Código para representar matrices de confusión

6.3 Objetivos de Desarrollo Sostenible (ODS)

Aunque esta investigación quiere contribuir al contexto evolutivo tecnológico en el que nuestra sociedad se encuentra actualmente, también se pretende contribuir significativamente a los Objetivos de Desarrollo Sostenible (ODS) establecidos por la Naciones Unidas [25]. En particular, se busca alinear los esperables avances

tecnológicos que surjan a partir de esta investigación con las siguientes metas globales:

- Objetivo 9: Industria, Innovación e Infraestructura. Ya que de esta investigación se pretenden mejorar las tecnologías convencionales que emplean ondas visibles, por nuevas que empleen ondas milimétricas.
- Objetivo 11: Ciudades y Comunidades Sostenibles. Dado que esta investigación pretende mejorar la precisión de las tecnologías de detección de objetos, tecnologías como aquellas que gestionan bien sea el tráfico o la seguridad urbana, pueden ser mejoradas y contribuir a la creación de entornos urbanos tanto más seguros como eficientes.
- Objetivo 13: Acción por el clima. Ya que las ondas milimétricas son menos invasivas para el medio ambiente en comparación con las visibles.
- Objetivo 16: Paz, Justicia e Instituciones Sólidas. Como ya se ha comentado, los avances en la detección de objetos pueden aplicaciones como en seguridad o incluso en justicia, los cuales pueden contribuir a la seguridad pública.



Figura 6.1: Objetivos del Desarrollo Sostenible