



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Creación de demos técnica y
documentación asociada, para
la detección/clasificación de
materiales, utilizando un
sensor radar a 60GHz**



Presentado por Martín Encabo Contreras
en Universidad de Burgos — 10 de diciembre
de 2021

Tutores:

Pedro Latorre Carmona
José Francisco Díez Pastor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Martín Encabo Contreras, con DNI 72897369L, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 10 de diciembre de 2021

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Pedro Latorre Carmona

D. José Francisco Díez Pastor

Resumen

En este proyecto se va a desarrollar una aplicación mediante código Python que permita, mediante la lectura de materiales de un radar de 60 GHz, clasificar e indicar que tipo de material es. Además, se indica con que probabilidad es el material en cuestión.

Para realizar la parte del reconocimiento de materiales se aplicarán técnicas de aprendizaje automático, en concreto el clasificador Random Forest junto con la validación cruzada.

A la hora de recopilar la información necesaria, se usará la librería proporcionada por el fabricante del radar para conseguir una serie de lecturas con información suficiente capaz de identificar a cada objeto. Esta parte se realizará creando un registro de treinta materiales y diez lecturas de cada uno.

El objetivo del proyecto es el de demostrar el funcionamiento del sensor radar mediante una herramienta capaz de comparar diferentes tipos de materiales.

Descriptores

Aprendizaje Automático, Clasificador, Random Forest, Reconocimiento, Python, Sensor, Radar, Acconeer

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
Conceptos teóricos	5
3.1. Teledetección	5
3.2. Radar Acconeer	6
3.3. Servicio In-phase and Quadrature	9
3.4. Aprendizaje automático	9
3.5. Algoritmos machine learning	10
3.6. Características de los materiales	12
Técnicas y herramientas	15
4.1. Técnicas y metodologías	15
4.2. Lenguajes y bibliotecas	16
4.3. Herramientas de desarrollo	18
4.4. Herramientas de documentación	18
Aspectos relevantes del desarrollo del proyecto	19
5.1. Propuesta del Proyecto	19

5.2. Metodologías aplicadas	20
5.3. Formación	20
5.4. Montaje del radar	20
5.5. Lectura de los objetos	22
5.6. Modelo clasificador	24
Trabajos relacionados	25
6.1. RadarCat	25
Conclusiones y Líneas de trabajo futuras	27
7.1. Conclusiones	27
7.2. Líneas de trabajo futuras	27
Bibliografía	29

Índice de figuras

3.1. Sistemas de observación global.	6
3.2. Radar A111.	7
3.3. Diagrama de bloques del sensor A111.	8
3.4. Random Forest	11
3.5. K-nearest neighbors	12
3.6. Herramienta Acconeer	13
5.1. Regresión Logística	24
6.1. RadarCat.	26

Índice de tablas

3.1. Servicios del radar A111	8
---	---

Introducción

La tecnología de radar existe desde la década de 1930 de la mano de Watson-Watt. El término radar proviene del acrónimo inglés RAdio Detection And Ranging.

El uso típico de los radares de radiofrecuencia se basa en medir distancias, altitudes, direcciones y velocidades de objetos. Un uso muy reconocido es el mapa de navegación de los barcos.

Durante los últimos años han surgido nuevas áreas de aplicación que plantean diferentes desafíos. Una aplicación es monitorizar los signos vitales, reconocimiento de gestos, entre otros.

Los radares últimamente se han vuelto más baratos, en gran parte debido a su adopción en la industria automotriz, lo que convierte a estos dispositivos en una opción atractiva en una amplia gama de aplicaciones de bajo costo.

Este trabajo se centra en documentar y demostrar el uso de un radar de 60GHz fabricado por Acconeer. Usando un procedimiento de extracción de características y aprendizaje automático. Para ello se realizará un registro de tres tipos de materiales (cartón, plástico y cristal) y se creará un modelo de IA para conseguir un correcto funcionamiento del radar.

Objetivos del proyecto

En este apartado se explican los objetivos marcados en el proyecto, diferenciando entre los objetivos generales para llevar a cabo en este proyecto y los objetivos técnicos a lo largo de su ejecución.

2.1. Objetivos generales

A continuación se muestran los principales objetivos generales del proyecto:

- El objetivo principal del proyecto se centra en documentar y demostrar el uso de un radar de 60GHz fabricado por Acconeer capaz de identificar, mediante un clasificador, el tipo de material al que pertenece un objeto.
- Crear un procedimiento de extracción de características y aprendizaje automático.
- Otro objetivo es el de mostrar un gráfico de probabilidades de la clasificación del material.
- Se intentará desarrollar una interfaz amigable para Windows.

2.2. Objetivos técnicos

Conceptos teóricos

Para construir un modelo capaz de capturar con precisión las características distintivas de las superficies objetivo, primero se debe comprender el origen y la estructura de la señal recibida. En este apartado se introducen algunos conceptos fundamentales del sistema de radar.

3.1. Teledetección

Teledetección o detección a distancia (detección remota) es la técnica que permite recopilar información a distancia de objetos sin que exista un contacto material. Para que esto sea posible tendremos una interacción entre los objetos y un sensor situado en una plataforma.

La distancia a que debe estar situado un sensor para ser remoto puede variar desde pocos decímetros hasta miles de kilómetros.

La teledetección es un flujo de radiación emitido por los objetos o materiales hacia un radar o sensor. El origen del flujo puede venir de:

- Radiación solar reflejada por los objetos - Teledetección pasiva
- Radiación terrestre emitida por los objetos - Teledetección pasiva
- Radiación emitida por el sensor y reflejada por los objetos (radar) - Teledetección activa

Se llama teledetección pasiva a la técnica aplicada por los sensores que miden las variaciones de la energía procedente de los objetos sin intervenir en el campo natural y se denomina teledetección activa a aquellos que generan

un campo de energía artificial, utilizado para registrar y medir el efecto que en él producen los objetos.

Las características técnicas del sensor influye en la calidad de los datos y en la posibilidad de recibir información en distintas longitudes de onda.

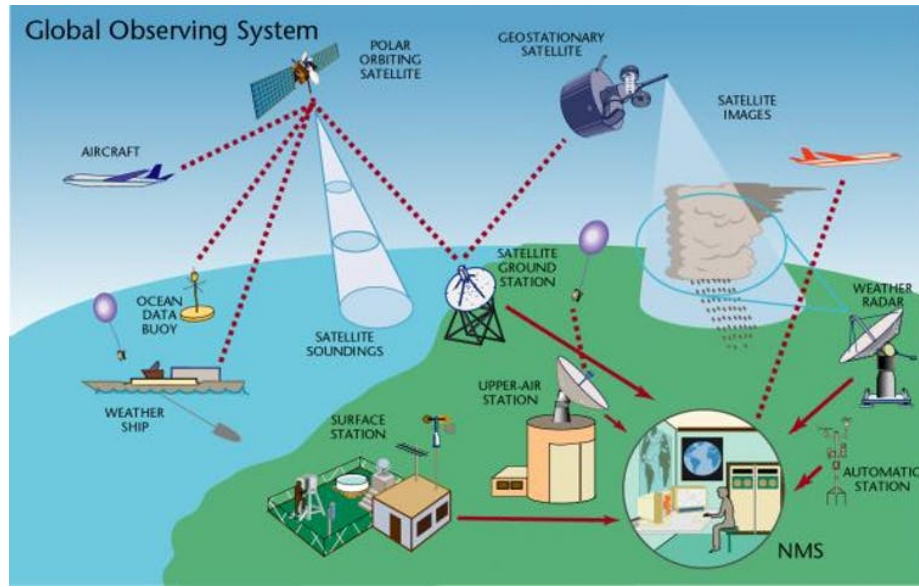


Figura 3.1: Sistemas de observación global.

En la figura 3.1 vemos la globalización actual de la teledetección utilizada para obtener información precisa, actualizada y de fácil acceso empleada para diversas utilidades y finalidades como pueden ser:

- Imágenes territoriales
- Motorización de las mareas
- Análisis de materiales terrestres

3.2. Radar Acconeer

El radar utilizado en el proyecto está fabricado por Acconeer llamado A111. Es un radar de 60GHz basado en impulsos tecnología de radar coherente (PCR¹) totalmente integrado en un pequeño chip de 29 mm². Esto permitirá una fácil integración en cualquier dispositivo portátil impulsado por batería.

¹Pulsed Coherent Radar

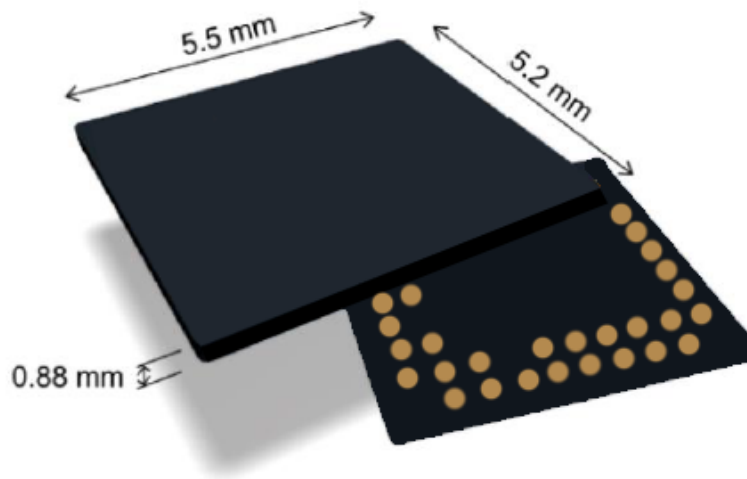


Figura 3.2: Radar A111.

Aplicaciones:

- Mediciones de distancia de alta precisión con mm de precisión y alta tasa de actualización.
- Detección de movimiento.
- Detección de velocidad.
- Detección de material.
- Seguimiento de objetos de alta precisión como el control de gestos.
- Seguimiento de alta precisión de objetos 3D.
- Control de los signos vitales, como la respiración y la frecuencia del pulso.

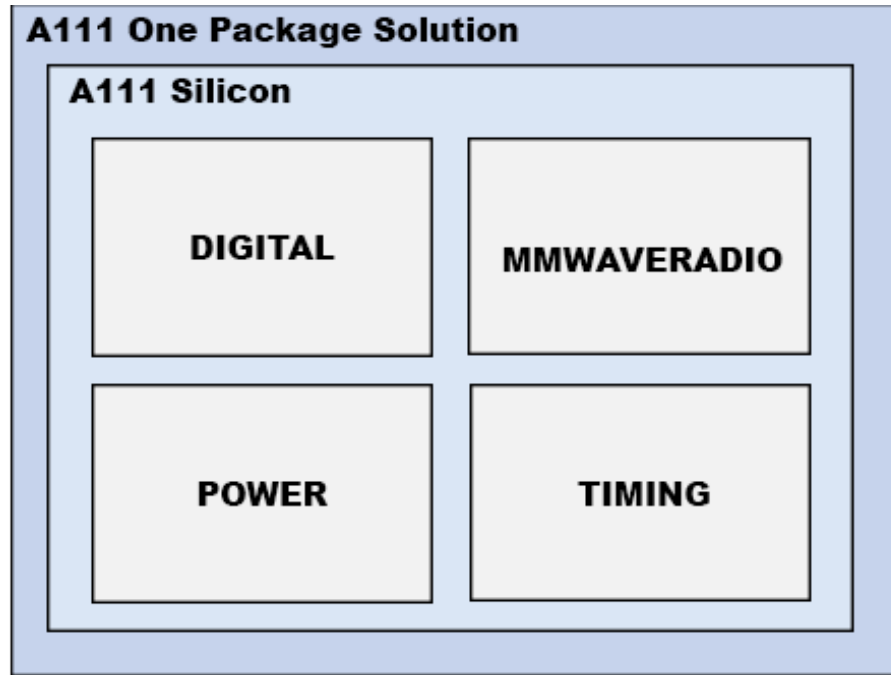


Figura 3.3: Diagrama de bloques del sensor A111.

La figura 3.2 muestra un diagrama de bloques del sensor A111. La señal se transmite desde la antena Tx y es recibida por la antena Rx, ambas integradas en la capa superior del sustrato del paquete A111. Además de la radio mmWave, el sensor consta de administración de energía y control digital, cuantificación de señales, memoria y un circuito de temporización.

El sensor se puede ejecutar en uno de los siguientes servicios básicos de la tabla 3.1.

Servicio	Tipo de dato	Ejemplo de uso
Envelope	Amplitud	Distancia absoluta y presencia estática
IQ	Amplitud y fase	Detección de obstáculos, respiración y distancia relativa
Sparse	Amplitud instantánea	Velocidad, detección de presencia y detección de gestos

Tabla 3.1: Servicios del radar A111

3.3. Servicio In-phase and Quadrature

El servicio In-phase and Quadrature (IQ) utiliza la coherencia de fase del radar pulsado Acconeer para producir componentes estables en fase y en cuadratura. Este servicio se puede utilizar para la detección de presencia frente al sensor, la detección de la frecuencia respiratoria, la detección de obstáculos y , en nuestro caso, para diferenciar materiales.

Los componentes en fase y en cuadratura se representan como valores complejos, lo que genera un conjunto complejo de N_D muestras representadas como $x[d]$, donde d es el índice de demora de la muestra.

Los datos obtenidos a través del servicio IQ proporcionan un método para examinar la reflectividad a diferentes distancias del sensor de radar.

3.4. Aprendizaje automático

El aprendizaje automático o machine learning es un tipo de inteligencia artificial (AI), consiste en programar una computadora para que mejore en la realización de una tarea a partir de datos de ejemplo o de la experiencia.

La inteligencia artificial es un concepto de creación de máquinas inteligentes que estimula el comportamiento humano, mientras que el aprendizaje automático es un subconjunto de la inteligencia artificial que permite que la máquina aprenda de los datos sin ser programada.

La diferencia entre el software informático normal y el aprendizaje automático es que un desarrollador humano no ha dado códigos que le indiquen al sistema cómo reaccionar ante la situación, sino que está siendo entrenado por una gran cantidad de datos.

Tipos de aprendizaje automático

Aprendizaje supervisado

El aprendizaje supervisado es una técnica en la que el programa recibe datos de entrada etiquetados y los datos de salida esperados. Obtiene los datos de los datos de entrenamiento que contienen conjuntos de ejemplos.

Generan dos tipos de resultados:

- Clasificación: Notifican la clase de los datos que se presentan.
- Regresión: esperan que el producto produzca un valor numérico.

Aprendizaje sin supervisión

Este tipo de algoritmo consta de datos de entrada sin etiquetar. Con una intervención humana menor. Se utiliza principalmente en análisis exploratorios, ya que puede identificar automáticamente la estructura en los datos.

Aprendizaje reforzado

Este modelo interactúa con el entorno y se utiliza para tomar una secuencia de decisiones. Se puede decir que es un método de seguimiento y error para encontrar el mejor resultado basado en la experiencia.

3.5. Algoritmos machine learning

Algoritmos que se han utilizado y planteado para el desarrollo del proyecto:

- Random Forest
- Regresión Logística
- KNN

El algoritmo RandomForest ha sido el seleccionado para realizar el proyecto.

Random Forest

Random Forest conocido en castellano como "Bosques Aleatorios"^{es} una combinación de árboles predictores, de modo que cada árbol depende de los valores de un vector aleatorio muestreado de forma independiente y con la misma distribución para todos los árboles del bosque.

Los árboles se crean siguiendo el algoritmo:

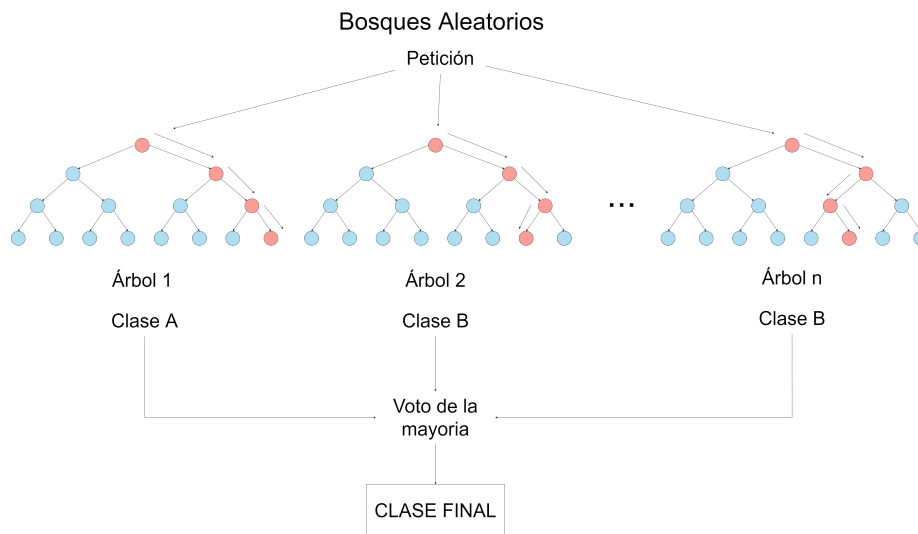


Figura 3.4: Random Forest

- Sea N el número de casos de prueba, M es el número de variables en el clasificador.
- Sea m el número de variables de entrada, m menor que M
- Se elige un conjunto de datos para el entrenamiento del árbol y el resto de los casos se utilizará para estimar el error.
- Para cada nodo del árbol, elegir aleatoriamente m variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las m variables.

Regresión Logística

El método de regresión logística es uno de los métodos estadísticos más utilizados para resolver problemas de clasificación binaria (clasificación de dos clases), donde el resultado solo puede ser de naturaleza dicotómica, o sea, solo puede tomar dos valores posibles.

Este algoritmo se puede utilizar para varios problemas de clasificación, detección de spam, predicción de la diabetes, si un cliente determinado comprará o no un producto en particular.

KNN

K-nearest neighbors traducido en castellano como "k vecinos más próximos" su funcionalidad es encontrar un número predefinido de muestras de entrenamiento más cercanas en distancia al nuevo punto y predecir la etiqueta a partir de ellas.

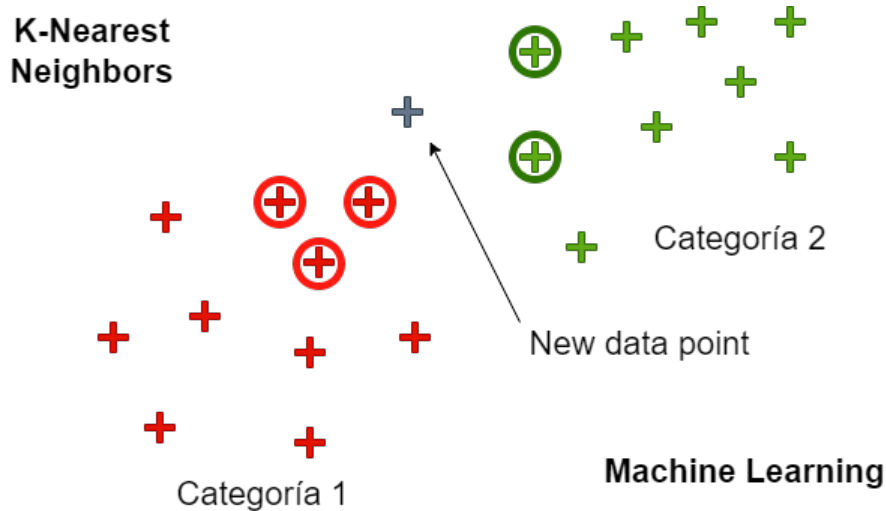


Figura 3.5: K-nearest neighbors

El número de muestras puede ser una constante definida por el usuario (aprendizaje del vecino más cercano k) o variar según la densidad local de puntos (aprendizaje del vecino basado en el radio). La distancia puede ser cualquier medida, la distancia euclidiana estándar es la opción más común.

3.6. Características de los materiales

Se ha realizado una extracción de características de los materiales utilizados para el entrenamiento del modelo de clasificación. Las características o atributos obtenidos son una serie de datos que identifican el material, estos datos están representados por números complejos. Estos números serán extraídos de las lecturas y separados en módulo y fase para poder realizar cálculos correctos.

Cada extracción se ha realizado mediante la herramienta facilitada por Acconeer.

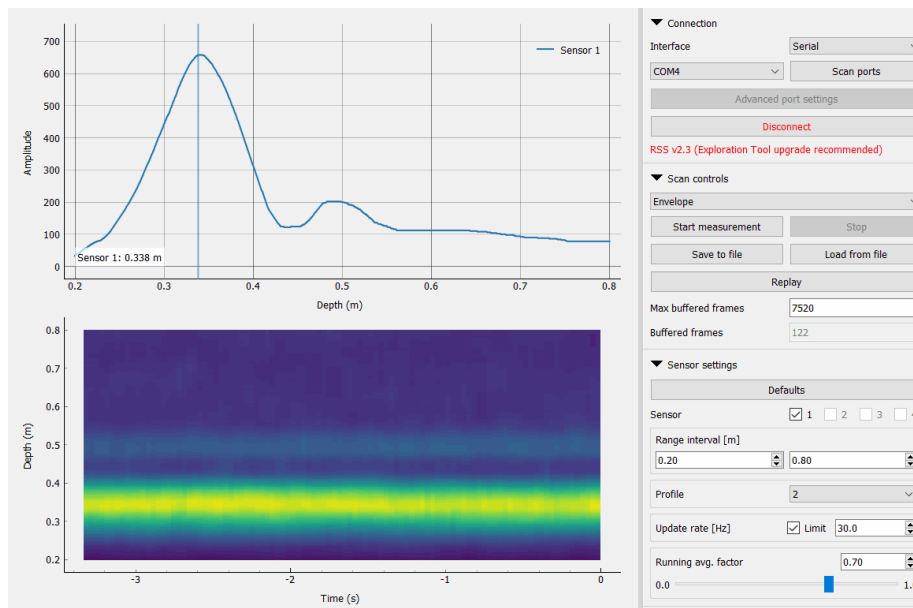


Figura 3.6: Herramienta Acconeer

Se necesita la aplicación Acconeer Exploration Tool que se puede descargar desde el repositorio de este TFG https://github.com/mecyc/TFG_RADAR_60GHZ/tree/main/acconeer-python-exploration. En esta interfaz se conecta utilizando el modo socket a la IP de la Raspberry Pi 4.

Una vez dentro hay cuatro servicios y varias funcionalidades. Los servicios son:

- Power Bins: según <https://acconeer-python-exploration.readthedocs.io/en/latest/services/pb.html>, calcula la energía de diferentes distancias, su objetivo es la medición de objetos grandes a distancias cortas como un sensor de parking. Al ser un método muy sencillo con poca cantidad de datos y enfocado a grandes objetos como una pared no se ha contemplado para el estudio
- Envelope: según <https://acconeer-python-exploration.readthedocs.io/en/latest/services/envelope.html>, es igual que Power Bins pero utilizando un espectro continuo de los datos. Su caso de uso típico es la detección estática, por esto se ha contemplado para el estudio ya que la basura estará quieta.
- IQ: según <https://acconeer-python-exploration.readthedocs.io/en/latest/services/iq.html>, utiliza la coherencia de fase del radar

que detecta movimiento a nivel fino. Tiene cinco modos, el primero es el que ha sido relevante al detectar mejores reflejos de los datos y estar optimizado para distancias muy cortas.

- Sparce: según <https://aconeer-python-exploration.readthedocs.io/en/latest/services/sparse.html>, se base en la señal está más cuantizada y su uso principal es el análisis del movimiento y no de situaciones estáticas, por ello no se ha contemplado su uso para el estudio.

Para la creación del modelo se han seleccionado 30 materiales divididos en, 10 de plástico, 10 de cristal y 10 de cartón. De cada material se han realizado 10 lecturas, de varias caras, girando el objeto.

Llegamos a una colección de 300 lecturas exportadas cada una en ficheros con formato numpy (.npy) donde están almacenadas las características en vectores y matrices. Un porcentaje de estos datos conformaran la red de entrenamiento y otro porcentaje servirán para testear la red.

Cada instante de tiempo de la lectura comprende 291 atributos, de cada fichero obtenemos del orden de 300 instancias. Una instancia son estadísticas (medias, stds, etc) de los 291 atributos calculadas a partir de los aproximadamente 300 instantes de tiempo.

Tenemos los siguientes datos:

- N° Experimentos: Material - vista
- M instantes de tiempo
- Atributos

Técnicas y herramientas

4.1. Técnicas y metodologías

Metodología SCRUM

Se trata de una metodología de trabajo ágil que tiene como finalidad dividir en periodos de tiempo el flujo de trabajo. Estos periodos son conocidos como *sprints*, al finalizar cada *sprint* se realizan revisiones y reuniones donde se deciden las tareas de los próximos *sprints*.

Cliente de control de versiones

- Herramientas consideradas: GitHub Desktop y Gitkraken
- Herramienta elegida: GitHub Desktop

GitHub Desktop utilizado para la gestión ágil del proyecto.

Hosting del repositorio

- Herramientas consideradas: GitLab y GitHub.
- Herramienta elegida: GitHub

GitHub ofrece una gran cantidad de facilidades para mantener el proyecto en la nube y debido a que no cobra por sus servicios lo convierte en la mejor opción posible. Nos permite alojar nuestro repositorio central del proyecto usando el control de versiones Git.

Git es un sistema de control de versiones distribuido de código abierto. El control de versiones nos permitirá retornar a algún punto anterior del desarrollo de nuestra aplicación en caso de sufrir errores.

Url de la herramienta: <https://github.com/>

4.2. Lenguajes y bibliotecas

Python

Python es un lenguaje de programación que destaca por su código limpio y legible esto hace que sea uno de los lenguajes de iniciación de muchos programadores. Además se trata de un lenguaje de multiparadigma y multiplataforma muy utilizado en la técnica del BigData.

Url de la herramienta: <https://www.python.org/>

TensorFlow

TensorFlow es una biblioteca de software de código abierto para computación numérica. Desarrollada por Google capaz de construir y entrenar redes neuronales.

Tensorflow es capaz de realizar cálculos numéricos usando grafos de flujo de datos. Los nodos hacen referencia a las operaciones matemáticas y los enlaces representan conjuntos de datos multidimensionales que relacionan los nodos.

La arquitectura flexible de Tensorflow permite realizar los cálculos en una o varias CPUs o GPUs mediante una sola API.

Url de la herramienta: <https://www.tensorflow.org/>

NumPy

NumPy es una biblioteca utilizada en la programación con Python para crear vectores y matrices grandes multidimensionales.

La funcionalidad principal de NumPy es su estructura de datos "ndarray", para una matriz de n dimensiones.

El uso de NumPy en Python brinda una funcionalidad comparable a MATLAB [?].

Url de la herramienta: <https://numpy.org/>

Pandas

Pandas es una biblioteca de código abierto especializada en el manejo y análisis de estructuras de datos flexible y fácil de usar, construida sobre el lenguaje de programación Python.

Su objetivo es ser el bloque de construcción fundamental de alto nivel para realizar análisis de datos prácticos del mundo real en Python.

Pandas está construido sobre NumPy y está destinado a integrarse bien dentro de un entorno informático científico con muchas otras bibliotecas de terceros.

Url de la herramienta: <https://pandas.pydata.org/>

acconeer.exptool

acconeer.exptool es la biblioteca facilitada por la empresa Acconeer. Con ella pondremos en funcionamiento el radar de 60GHz además de recopilar los datos necesarios.

Url de la herramienta: <https://github.com/acconeer/acconeer-python-exploration>

Sklearn

Scikit-learn es una biblioteca para aprendizaje automático de software libre para el lenguaje de programación de Python. En esta biblioteca encontramos varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, bosques aleatorios, Gradient boosting, K-means y el utilizado en este proyecto RandomForest.

Url de la herramienta: <https://scikit-learn.org/>

Tkinter

Tkinter es una adaptación de la biblioteca gráfica Tcl/Tk [1] para el lenguaje de programación Python. Mediante Tkinter se desarrollará la interfaz gráfica para comparar distintos tipos de materiales.

Url de la herramienta: <https://docs.python.org/es/3/library/tkinter.html>

4.3. Herramientas de desarrollo

Jupyter Notebook

Jupyter Notebook es el entorno de trabajo utilizado en el proyecto que permite desarrollar código en Python de manera dinámica. Nos ofrece integrar en un mismo archivo bloques de código, texto, gráficas o imágenes. Utilizado ampliamente en análisis numéricos y estadísticos.

Jupyter admite más de 40 lenguajes de programación, incluidos Python, R, Julia y Scala.

Url de la herramienta: <https://jupyter.org/>

4.4. Herramientas de documentación

\LaTeX

\LaTeX es un sistema de elaboración de documentos basado en comandos. El proyecto está desarrollado mediante una plantilla creada en \LaTeX .

Url de la herramienta: <https://jupyter.org/>

MiKTeX

MiKTeX es una distribución de \LaTeX encargada de gestionar los componentes y paquetes. Tiene la capacidad de actualizarse así mismo descargando nuevas versiones de componentes.

Url de la herramienta: <https://miktex.org/>

Texmaker

Texmaker es el editor de documentos \LaTeX utilizado para funcionar necesita MiKTeX.

Url de la herramienta: <https://www.xm1math.net/texmaker/>

BibItNow!

BibItNow! es la extensión del navegador que utilizamos para citar páginas web en el formato Bibtex, formato soportado por \LaTeX .

Url de la herramienta: <https://chrome.google.com/webstore/detail/bibitnow/bmnfikjlonhkoojjfddnlbinkkapmldg>

Aspectos relevantes del desarrollo del proyecto

En este apartado se va a recoger el ciclo de vida del proyecto, detallando los aspectos más relevantes que se han tratado y como se han resuelto las diferentes dificultades encontradas a lo largo de su desarrollo. Se irán presentando diferentes secciones que concuerdan con el orden cronológico seguido en el proyecto y muestran la justificación de las decisiones tomadas.

5.1. Propuesta del Proyecto

La propuesta de este proyecto consistía en crear un modelo clasificador integrado en una interfaz para el reconocimiento de materiales que se dividía en las siguientes 3 tareas principales:

- Montaje del sensor: Puesta en funcionamiento del radar junto con sus componentes. Siendo capaces de extraer las características de las lecturas realizadas.
- Lecturas de los materiales: Conseguir información a través de las características de los materiales para crear un modelo capaz de diferenciar materiales como son el cartón, el plástico y el cristal.
- Clasificador de materiales: La tarea principal pedida para realizar este proyecto era la de construir un clasificador de materiales a través de un radar de 60GHz devolviendo la identificación correcta del material.

5.2. Metodologías aplicadas

En el desarrollo de este proyecto se ha intentado emplear en la medida de lo posible la metodología ágil de Scrum.

Debido a que el proyecto se comenzó en abril de 2021 en la modalidad presencial, no se ha podido seguir de forma estricta todas las pautas de la metodología, como las reuniones diarias con todo el equipo.

Las pautas que se han seguido durante el proyecto han sido:

- Desarrollo incremental del proyecto mediante sprints.
- Sprints de 1 a 2 semanas.
- Al finalizar el sprint, reuniones para evaluar el proyecto y plantear los pasos a seguir en el siguiente sprint.

5.3. Formación

Durante las primeras fases del proyecto, fue necesario aprender el funcionamiento del Radar Sensor. Para esto, se consultó la documentación facilitada por Acconeer.

A parte de esto se investigó información adicional sobre el uso de estos radares así como ejemplos de su uso, la información extraída de la red fue escasa o casi nula. únicamente fue válida la documentación facilitada por Acconeer² en el repositorio de GitHub.

5.4. Montaje del radar

En esta sección se indican tanto las necesidades hardware como software para el montaje y puesta en funcionamiento del radar.

Necesidades hardware

Los creadores de Acconner crearon un video ([EVK 2](#)) con las instrucciones del montaje del sensor. En el se explica el montaje de los diferentes componentes a la Raspberry Pi 4 (a partir de ahora Raspberry").

En un comienzo será necesario disponer de:

²<https://acconeer-python-exploration.readthedocs.io/>

- Raspberry Pi 4
- Radar A111
- Placa XR112
- Cable flexible para XR112
- Tarjeta SD
- Teclado USB
- Ratón USB
- Monitor con HDMI
- Cable HDMI

FOTO

La placa se conecta a la raspi por el puerto GPIO quedando por encima del resto.

FOTO

Se ha realizado un espacio o cubículo de lectura de objetos de $30 \times 22 \times 25 \text{ cm}$ (ancho, largo y alto) con una incisión en la parte inferior en el medio del tamaño del sensor. El sensor fue bloqueado a su posición solamente colocándolo en la posición de la incisión. Esta caja de madera se utiliza poniendo la abertura en la parte inferior, mirando al suelo.

El sensor tiene sensibilidad a partir de los 10 cms. La sensibilidad del sensor se limitó al rango de 10 a 24 cm, el mínimo de sensibilidad hasta el suelo.

FOTOPROTOTIPO

Necesidades software

Una vez se había procedido con el montaje del dispositivo se pasó a la instalación y configuración del sistema operativo. Además, se instalaron los diversos programas necesarios tanto para recoger los datos como para poder interpretarlos.

Se necesitarán las siguientes aplicaciones para completar la instalación. Además, serán muy útiles a la hora de trabajar con el Radar. Hay que descargar e instalar:

- **Acconeer SW**: disponible para todos los usuarios (Windows, Linux, IOS)
- **Raspbian OS**
- **Etcher**: se usará para instalar el sistema operativo Raspbian en la SD
- **PuTTY**: se utiliza para conectarse a Raspberry Pi
- **WinSCP**: utilizado para transferir archivos a Raspberry Pi

5.5. Lectura de los objetos

Para la creación del modelo se han seleccionado 30 materiales divididos en, 10 de plástico, 10 de cristal y 10 de cartón. De cada material se han realizado 10 lecturas, de varias caras, girando el objeto.

Llegamos a una colección de 300 lecturas exportadas cada una en ficheros con formato numpy (.npz) donde están almacenadas las características en vectores y matrices. Un porcentaje de estos datos conformarán la red de entrenamiento y otro porcentaje servirán para testear la red.

Cada instante de tiempo de la lectura comprende 291 atributos, de cada fichero obtenemos del orden de 300 instancias. Una instancia son estadísticas (medias, stds, etc) de los 291 atributos calculadas a partir de los aproximadamente 300 instantes de tiempo.

Tenemos los siguientes datos:

- N° Experimentos: Material - vista
- M instantes de tiempo
- Atributos

Se creará la siguiente matriz:

$$Matriz1 = (N \cdot M) \cdot A \quad (5.1)$$

Es una matriz 2D ($N \cdot M, A$)

La extracción y creación de la matriz con los datos necesarios se realizará mediante Python:

```
diccionario = np.load('C01_V01.npy',allow_pickle=True).item()
data = diccionario.get('sweep_data').get('data')
data = data.reshape(data.shape[0],data.shape[2])
```

Los datos de la lectura tienen una parte real y otra imaginaria, a partir del array 2D anterior, se obtiene un array 2D, con el doble de anchura. Por ejemplo, si Matriz1 es de 300×291 , se obtendrá otra matriz que comprende el módulo y la fase de 600×291 . Así se obtiene el módulo del número complejo y la fase del número complejo. El módulo será una matriz de 300×291 y la fase también.

```
modulo = abs(data)
fase = []
for i in data:
    fila = []
    for j in i:
        fila.append(phase(j)) #Fase
    fase.append(fila)
    fase = np.asarray(fase)
    modulo_fase = np.concatenate((modulo, fase), axis=0)
```

A partir de los datos del módulo y la fase se obtiene la media.

```
traspuesta = np.transpose(modulo_fase)
#Obtenemos la matriz traspuesta(291x600) de modulo_fase(600x291)

#Separamos la traspuesta en modulo y fase
t_modulo = traspuesta[:,int(traspuesta.shape[1]/2)]
t_fase = traspuesta[:,int(traspuesta.shape[1]/2):]
media = []
for i in t_modulo:
```

```

        media.append(i.mean())
for j in t_fase:
    media.append(j.mean())
#media -> medias modulo y fase

```

5.6. Modelo clasificador

Para elegir el modelo a usar se tuvieron que tener en cuenta diferentes factores como el peso del modelo, su carga de computo y su precisión.

Se realizaron pruebas con los diferentes modelos (KNN, RamdonFores y Logistic Regression) y al final opté por usar el modelo RandomFores debido a que el promedio hace que este modelo sea mejor que un solo árbol de decisión, por lo que mejora su precisión y reduce el sobreajuste.

La Regresión Logística obtuvo un bajo porcentaje de coincidencia un 59,33 %

```

from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_test2, y_prds)
accuracy2 = accuracy_score(y_train, y_prds_train)

accuracy, accuracy2

(0.5933333333333334, 0.8851851851851852)

```

Figura 5.1: Regresión Logística

Trabajos relacionados

En este apartado se muestran las herramientas relacionadas con el proyecto. Hoy en día hay poca información y documentación de este tipo de tecnología de radares en la red, por ello solo muestro una única herramienta.

6.1. RadarCat

RadarCat (Radar Categorization for Input and Interaction) permite que un dispositivo electrónico pueda reconocer y clasificar distintos materiales y objetos a tiempo real con alta precisión. Además da mucha más información de los objetos, como su estructura interna.

El escaner/radar utilizado envía ondas electromagnéticas al objeto, estas rebotan y vuelven al punto de partida para su procesamiento con aprendizaje automático o machine learning.

La figura 6.1 muestra un esquema de cómo está estructurado RadarCar. Podemos ver que es necesario un ordenador para inicializar el software de RadarCat, a este equipo se conecta una placa base donde se encuentran las antenas/radares capaces de realizar lecturas de objetos. Por último encontramos una placa que protege el sensor y soporta el objeto a reconocer.

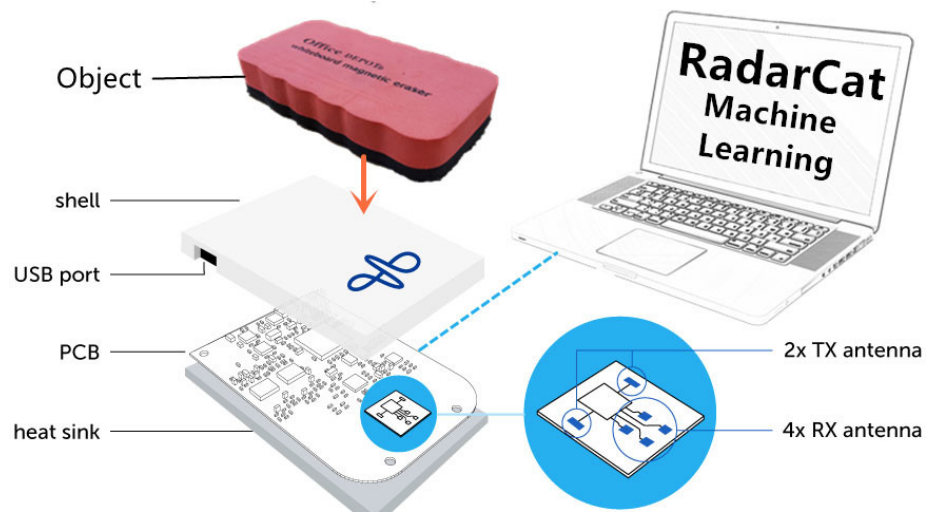


Figura 6.1: RadarCat.

Más allá de la interacción humana con la computadora, RadarCat también abre nuevas oportunidades en áreas como la navegación y el conocimiento mundial (por ejemplo, usuarios con baja visión).

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

En este trabajo se exploraron las posibilidades de utilizar un radar de ondas de 60GHz para la clasificación de diferentes tipos de materiales. Se descubrió que extrayendo características mediante el radar y combinando esto con un clasificador Random Forest es posible realizar una clasificación para distinguir tres tipos de materiales comunes en la vida diaria como son PLÁSTICO, CRISTAL y CARTÓN con una precisión del 99,89 % usando validación cruzada.

Mediante una aplicación en fase Beta se ha conseguido realizar la lectura de las características de varios materiales y su correcta identificación.

Una posible aplicación sería automatizar la diferenciación de los tipos de embalajes en una empresa de envíos para su correcto reparto evitando los golpes.

Respecto a las dificultades encontradas, algunas han surgido por intentar abarcar una gran cantidad de lecturas de distintos objetos tanto en su estructura como en su forma, lo que ha significado tener un gran número de características dificultando la tarea de implementar un modelo clasificador que diese una gran exactitud a la hora de indicar el tipo de material.

7.2. Líneas de trabajo futuras

Posibles líneas de trabajo futuras relativas al uso del radar:

- Combinar el radar utilizado de Acconeer con un sensor radar del fabricante Infineon Technologies, posiblemente colocados en dos posiciones del habitáculo de lectura (un radar superior y otro lateral).
- Refactorización del código actual para su uso en otros sensores.
- Aplicar otros clasificadores.
- Añadir otros estadísticos.
- Automatizar el proceso de creación y entrenamiento del clasificador mediante una base de datos, así como la ampliación de la capacidad de reconocer materiales nuevos.

Bibliografía

- [1] Colaboradores de los proyectos Wikimedia, “Tcl - Wikipedia, la enciclopedia libre,” Apr 2020. [Online; accessed 24. Jun. 2021].