Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Segmentación semántica RGB-D para navegación en interiores

Trabajo Fin de Máster

Presentado por: Aldalur Corta, Mikel

Director/a: Cobos Guzman, Salvador

Ciudad: Bilbao
Fecha:

Resumen

En este apartado se introducirá un breve resumen en español del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Palabras Clave: Se deben incluir de 3 a 5 palabras claves en español

Abstract

En este apartado se introducirá un breve resumen en inglés del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Keywords: Se deben incluir de 3 a 5 palabras claves en inglés

Índice de contenidos

Índice de tablas

Índice de figuras

1. Introducción

La navegación autónoma es cada vez más popular entre los robots y los vehículos móviles. De esta manera estos robots o vehículos pueden moverse en base a los sensores de captación que integran, siendo necesario la efectividad y la correcta limitación de navegación. Las disciplinas de visión por computador y aprendizaje automático son las que realizan la investigación en el área de navegación autónoma y cada vez realizan avances más elaborados y profundos.

Estos avances han ayudado a que los robots sean cada vez más fiables en cuanto a evitar la colisión con los obstáculos del entorno, crear mapa de navegación, percepción de personas o la segmentación. Los procesos de analizar el entorno han ido cambiando y la seguridad de estos sistemas es esencial en los entornos donde se encuentran personas, animales, plantas o cualquier tipo de obstáculo. La navegación puede ser dentro de un entorno aprendido o un entorno nuevo según las necesidades, pero en todos los casos es necesario identificar o captar bien el entorno para luego poder elegir el camino adecuado. Como ejemplos podemos tener diferentes tipos de entornos como oficinas, escuelas, hospitales o cualquier tipo de edificio donde se quiera introducir un robot.

1.1 Motivación

Los robots autónomos son de una gran ayuda para realizar cualquier tarea y la navegación y la percepción son unos de los puntos en los que se están realizando las mejorando. Hoy en día, podemos ver cómo los robots nos limpian la casa, acompañan a personas, realizan búsquedas de elementos en diferentes entornos o realizan las tareas de inventario de almacenes. Estos robots tienen que ser seguros y evitar todos los obstáculos. Estos obstáculos pueden ser seres vivos y pueden estar en peligro si el robot no realiza bien su trabajo.

La segmentación es una de las áreas en progreso para este tipo de problemática y los últimos avances de la navegación autónoma se centran en la visión por computador para realizarla. La segmentación mediante la captación de imágenes del entorno es esencial para que un robot navegue autónomamente con la suficiente fiabilidad y seguridad para que no ocurra ningún imprevisto.

Es necesario entrenar a los modelos de visión por computador mediante el aprendizaje automático para que realicen una segmentación correcta de los elementos que se encuentran

en el entorno y los clasifique para que el robot tenga claro cuál es el sitio por donde puede circular. El robot tiene que navegar tanto en el interior como exterior y tiene que ser capaz de tomar decisiones correctas a la hora de realizar la navegación, aunque el entorno sea desconocido.

1.2 Planteamiento del trabajo

En este trabajo se intenta abordar el problema en los entornos de interior donde el robot tiene que elegir el camino por donde circular. Para ello, en vez de utilizar solo un sensor RGB para la segmentación, se utilizará un sensor RGB-D para tener una mejor segmentación donde se le añade la profundidad del entorno. A la cámara Intel RealSense D435i se le añadirá un sistema de aprendizaje profundo para detectar las trayectorias de los humanos para que realice una correcta segmentación.

Para la movilidad se necesita un robot que sea capaz de realizar movimientos en el entorno. Este robot móvil será el que se llama Jetbot, el robot móvil de software libre que utiliza como módulo de control el kit de desarrollo Jetson Nano. Al robot se le añadirá un sistema de navegación para que puede moverse de forma autónoma.

Todo el software irá dentro del framework ROS “Robot Operating System”, el software libre que facilita el desarrollo de los proyectos robóticos.

1.3 Estructura de la memoria

En el capítulo 2 se realiza el análisis y estado del arte de todo lo referente a los robots móviles y la percepción por computador. Se analizan diferentes publicaciones como sistemas de percepción de entorno para la localización y mapeo simultáneo (SLAM), Detección de objetos o segmentación semántica. Se parte de conceptos básicos para finalmente centrarse en los conceptos que se abordaran en el presente trabajo.

En el capítulo 3 se detallan los objetivos y la metodología a utilizar, mientras que en el capítulo 4 se identifican los requisitos para la elaboración del presente trabajo de la herramienta software.

Todo el desarrollo a detalle se lleva a cabo en el capítulo 5 y seguidamente se muestran los resultados obtenidos en el capítulo 6, las pruebas realizada y sus valoraciones para ser mas exactos. Finalmente, en el capítulo 7 se muestran las conclusiones obtenidas.

2. Contexto y estado del arte

La navegación autónoma está avanzando a pasos agigantados, para ello son evidentes los coches autónomos, robots humanoides o cuadrúpedos que ayudan en muchas de las tareas que tienen que realizar los seres humanos. La interacción de los robots con la humanidad también está creciendo ya que cada vez es más seguro y viable, por el avance de la tecnología y la investigación en este ámbito. La utilización de los diferentes sistemas, con extensas características, es indispensable para establecer la seguridad de los humanos y del entorno. En este apartado se analizarán estos elementos y algoritmos de procesamiento de esta información.

2.1. SLAM mediante visión por computador

La visión por ordenador se encarga de capturar y procesar las imágenes. Existen diferentes algoritmos para procesar las imágenes y obtener una perspectiva de tres dimensiones. En este apartado se analizan diferentes formas de crear un mapa utilizando diferentes configuraciones de cámaras para que los robots autónomos tengan conocimiento previo del entorno.

2.1.1. Cámara monocular

Las cámaras monoculares, las que se han utilizado toda la vida para sacar las fotos, son las más utilizada en todos los sistemas por la simplicidad y el pequeño coste que suponen. Con estas cámaras y los algoritmos desarrollados hasta ahora, se pueden realizar muchas operaciones.

Para obtener la posición de un elemento o comparar los elementos de las imágenes se utilizan los algoritmos de extracción de características. Con la extracción de las características se sacan los descriptores que se compararan con la siguiente imagen. Uno de los algoritmos utilizados es rotated brief (ORB) (R. Wang et al., 2019), que viene después de varios algoritmos como SIFT o SURF para mejorar la eficiencia y el coste computacional.

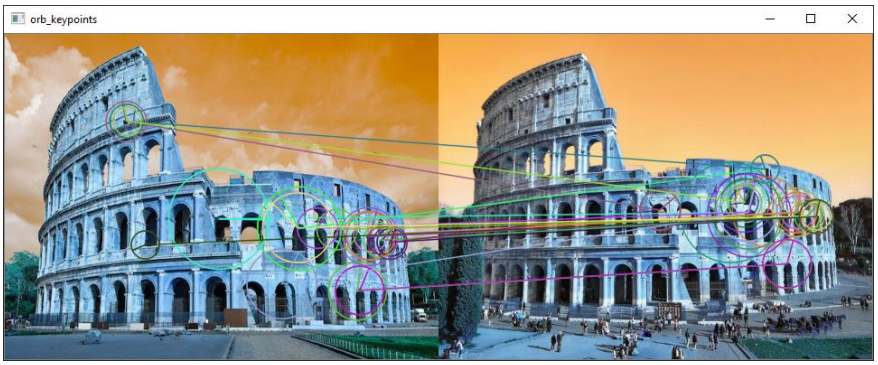


Figura 1: Imagen utilizando el algoritmo ORB para detectar objetos mediante características.

Este mismo algoritmo que se utiliza para la detección de objetos, realizar las imágenes panorámicas o para la mejora de imágenes en movimiento, también se ha utilizado en conjunto con una unidad de mediciones de inercia (IMU) para crear mapas de tres dimensiones (Mur-Artal et al., 2015). Con el sensor de inercias para detección de velocidad, orientación y aceleraciones se capta el movimiento realizado entre cada fotograma y realiza la estimación del entorno en 3D.

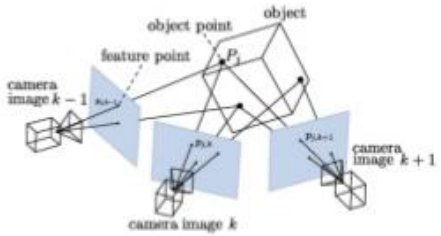


Figura 2: Simulación ORB SLAM de desde diferentes prespectivas (*ORB SLAM Proposal for NTU GPU Programming Course 2016*, n.d.).

La diferencia de movimiento calculado por el sensor IMU y los puntos generados en cada fotograma se convierten en un sistema de varias fotos con diferentes perspectivas que puede ser generada a un sistema de tres dimensiones.

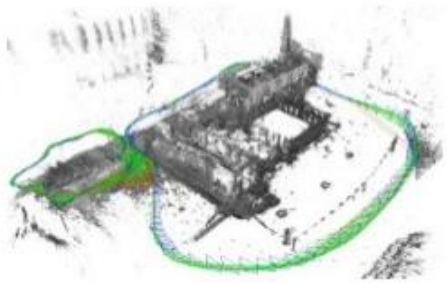


Figura 3: Sistema 3D creado por ORB-SLAM (*ORB SLAM Proposal for NTU GPU Programming Course 2016*, n.d.).

2.1.2. Cámara estéreo doble

El sistema monocular puede funcionar bien para ciertas funcionalidades, pero para asegurarse de que las mediciones se hacen de una manera más exacta, siempre es mejor contar con dos cámaras.

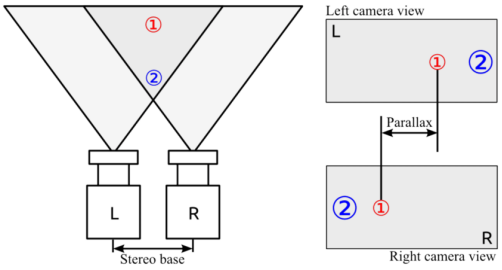


Figura 4: Principios de ajuste para cámaras estéreo (*Principle Drawing of a Stereo Camera Setup. Objects (1,2) in Various... | Download Scientific Diagram*, n.d.).

El sistema de funcionalidad es similar a la de cámara monocular, la diferencia es que estas cámaras, cada una de ellas, tienen diferentes ángulos de visión tal y como se puede ver en la Figura 4. Se ve claramente que la perspectiva de la profundidad se plasma en las dos imágenes que ayudan a realizar las medidas necesarias para obtener la visión estéreo (Zaarane et al., 2020). En este caso se puede ver que con las cámaras estéreo es posible medir las distancias para detectar obstáculos.

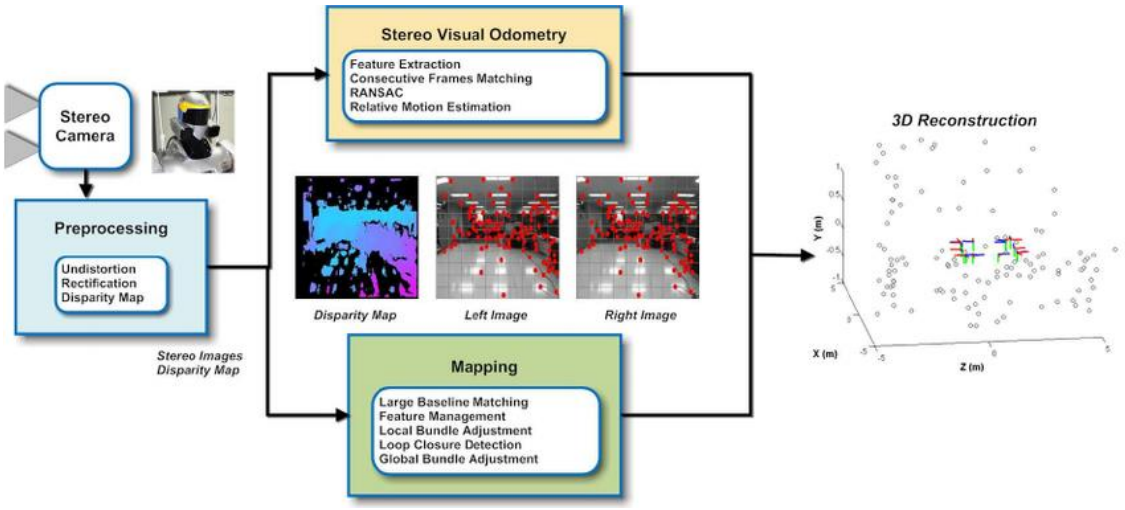


Figura 5: Generando un mapa de cámaras estéreo(*Stereo Visual SLAM System Overview: First, We Undistort and Rectify the... | Download Scientific Diagram*, n.d.).

Estos sistemas también se utilizan para generar mapas de entornos ya que tienen referencias más precisas de las que se obtienen con cámara monocular (Krombach et al., 2018). Se extraen las características del entorno y las distancias para que en cada fotograma que se detecte el movimiento se autogeneren los punto y distancias generando mapas de tres dimensiones.

Figura 6: Tipo cámara estéreo ZED (*ZED 2 - AI Stereo Camera | Stereolabs*, n.d.).

Podemos visualizar en la Figura 6 el tipo de cámaras que se utilizan como sistema de captura estéreo. Puede ser que haya dos cámaras instaladas independientemente o que las cámaras estén dentro de un encapsulado como la que vemos en la imagen.

2.1.3. Cámaras estereoscópicas

Las cámaras estéreo no son las únicas que pueden ser más exactas. Las cámaras estereoscópicas son otro tipo de cámaras que pueden medir las distancias. Como ejemplo la

cámara Intel RealSense D435i que tiene un sensor infrarrojo para la detección de profundidad con dos receptores diferentes, con similar función a la cámara estéreo, y la cámara RGB.

Figura 7: cámara Intel-RealSense-D435i (Intel, 2019).

Estas cámaras son denominadas RGB-D, ya que miden la profundidad mediante los sensores infrarrojos. Se pueden utilizar cada uno de los sensores por separado o también traen incorporada la opción de utilizar todos los sensores en conjunto. Utilizando todos los sensores en conjunto se puede obtener la imagen de la cámara con su respectiva profundidad. Esto no es practico para sacar una imagen de un solo fotograma, la imagen que genera suele ser una imagen con manchas negras por la detección de la profundidad, ver Figura 8.

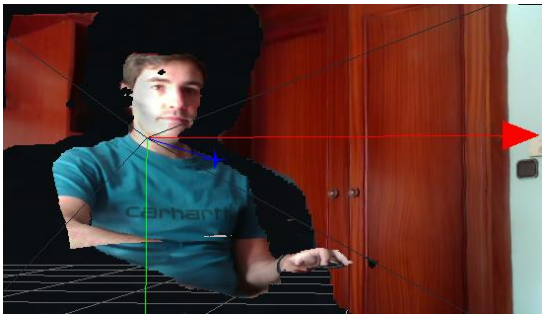


Figura 8: Imagen de profundidad de la cámara estereoscópica.

También lleva incorporado el sensor IMU lo que facilita el proceso de creación de mapas o entornos en tres dimensiones (Zhang et al., 2021). Existen diferentes algoritmos para la creación de mapas, Figura 9, que ayudan en la creación y entrenamiento de los robots para que se localicen e identifique su posición.

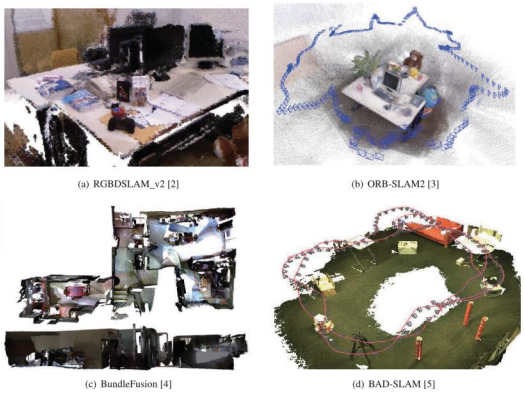


Figura 9: Creación de SLAM mediante cámaras estereoscópicas (Zhang et al., 2021).

2.1.4. LiDAR

Aparte de las cámaras, hay sistemas para la detección de entornos. Uno de ellos se denomina detección de luz y rango (LiDAR, siglas en ingles), que se puede ver hoy en día en los robots aspiradores. Estos sistemas de captación crean un mapa tipo rejilla en dos dimensiones siendo capaces de identificar la localización del robot (Chang et al., 2020), ver Figura 10.

Todos los elementos mencionados previamente se utilizan para un objetivo similar, aunque en realidad sean diferentes. Por ello se han realizado varios estudios identificando puntos de mejora donde se puede ver que si se juntan estos sistemas de captación, los sistemas funcionan de una manera más eficaz (Shin et al., 2020) (Mu et al., 2020).

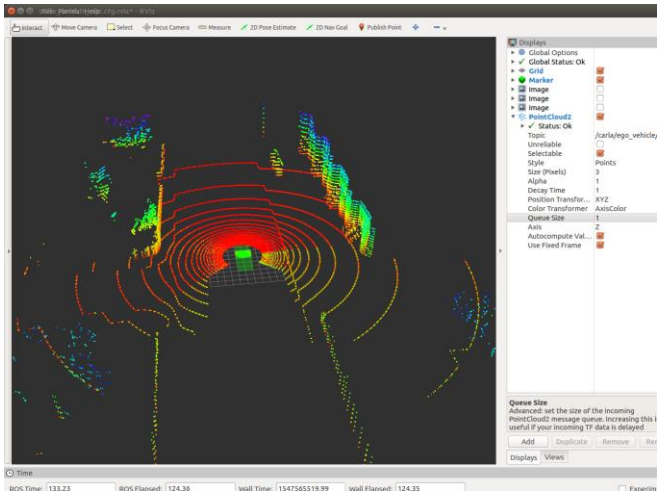


Figura 10. Mapa creado en RViz por un sensor LiDAR (*Incorrect Visualization of LiDAR · Issue #32 · Carla-Simulator/Ros-Bridge*, n.d.)

2.1.4. IMU

El sensor de inercia es un elemento esencial en los robots móviles para su navegación. Estos elementos o sensores dan la habilidad de saber como se mueve el robot. Normalmente constan de dos elementos separados en su interior, que medirán el giro y la aceleración.

Este elemento es indispensable si quieres utilizar tu robot con el sistema ROS. En el entorno o framework Robot Operating System este sistema se utiliza para estimar la posición del robot y además de eso es necesario si se quiere realizar un control exacto de las velocidades, porque los comandos de estos suelen ser la velocidad linear y angular.

En caso de querer realizar un control de estas velocidades es necesario realizar un software que controle el sistema mediante la adquisición de los datos de este sensor de inercia. Para un mejor entendimiento se pueden ver las tablas de la cámara Intel Realsense 435i que trae un sensor IMU integrado como se ha comentado anteriormente, ver Tabla 1 y Tabla 2 para ver las variables que se pueden obtener.

Tabla 1. Medidas de acelerómetro

Tipo	Acelerómetro		
Formato	MOTION_XYZ32F		
Numero de captura	5293		
Tiempo (ms)	1622619854455.03		
Eje (m/segundo^2)	0.0000000	-9.2084446	0.9316317

Tabla 2. Medidas del giroscopio

Tipo	Giro		
Formato	MOTION_XYZ32F		
Numero de captura	10688		
Tiempo (ms)	1622619824299.42		
Eje (º/ segundo)	-0.0052360	0.0000000	0.0017453

En la tabla podemos ver que el tiempo de adquisición es importante ya que la variable suele cambiar mucho.

2.3. Detección y segmentación

Los sistemas descritos en la sección anterior, las de captación, pueden ayudar a crear mapas y con ello identificar por donde circula el robot. Pero en realidad, lo interesante es que el robot identifique el lugar donde se encuentra y además detecte los objetos que tiene alrededor para que no se choque y tenga una perspectiva más clara para circular por el camino que le corresponda.

La detección y la segmentación de objetos son esas áreas que tratan que los robots autónomos identifiquen (Cao et al., 2019) y etiqueten (Mu et al., 2020) respectivamente los objetos de su entorno. Para ello es indispensable saber cómo se puede realizar un detector de objetos. Hoy en día, parece muy simple este concepto de detectar los objetos ya que lo podemos ver en distintas aplicaciones de nuestros móviles o en otros lugares, pero es interesante saber cómo funciona para poder crear o modificar modelos de redes neuronales convolucionales (CNN) que realizan este proceso. La segmentación de objetos se hace de una manera similar, pero en vez de detectar esos elementos e identificar su posición en la respectiva imagen, colorea la imagen de tal manera que se pueden diferenciar los objetos de forma semántica y de esta manera se crea un mapa de la misma imagen.

2.3.1. Detección de objetos

Como se menciona anteriormente, la detección de objetos es uno de los sistemas para que el robot identifique los elementos que le rodean. Es necesario realizar la ejecución estos algoritmos en los sistemas locales, ya que no se prescinde de tiempo para ejecución de estos algoritmos (Xu & Wu, 2020).

Antes de nada, para crear estos modelos de detección de objetos es necesario saber como funcionan las CNN. Pero antes de estas redes tan complejas se empezó a hacer la detección de diferentes elementos mediante Haar Cascade Classifiers, que consta de filtros morfológicos con diferentes kernel para la detección de los elementos como puedes ser las caras humanas.

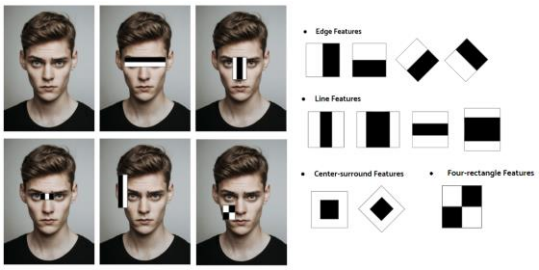


Figura 11: Ejemplo de detector de cara con Haar Cascade (Jiwon, 2019).

Estos sistemas pueden funcionar bien para ciertos elementos, pero hoy en día, hay diferentes modos de utilización de CNN que son más fiables. El proceso de detección de objetos se diferencia en tres fases que son la clasificación de imágenes (Chan et al., 2014), clasificación con localización del objeto y detección de diferentes objetos en una sola imagen (Ren et al., 2017).

Los clasificadores han evolucionado para mejorar la eficiencia y muestra de ello es la Figura 12, que ya en el año 2017 había mejorado mucho el tiempo de ejecución de detección de objetos. Hoy en día, ya se hace casi a tiempo real, muestra de ello es Yolov4 (Bochkovskiy et al., 2020) que realiza un buen desempeño teniendo en cuenta la velocidad o el tiempo de respuesta que tiene el algoritmo.

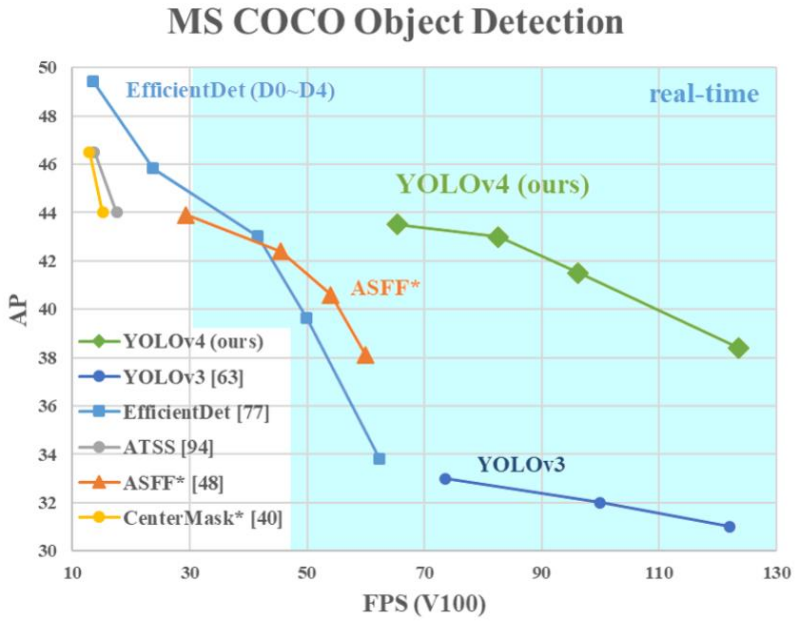


Figura 12: Comparando los algoritmos de detección de objetos en base al tiempo de respuesta(Bochkovskiy et al., 2020).

Uno de los principales motivos de las mejoras de estos sistemas son las competiciones de Imagenet, donde las empresas más punteras en el sector de la inteligencia artificial (IA) toman parte. Podemos ver en la Figura 13 como han evolucionado durante años los resultados de estas redes neuronales.

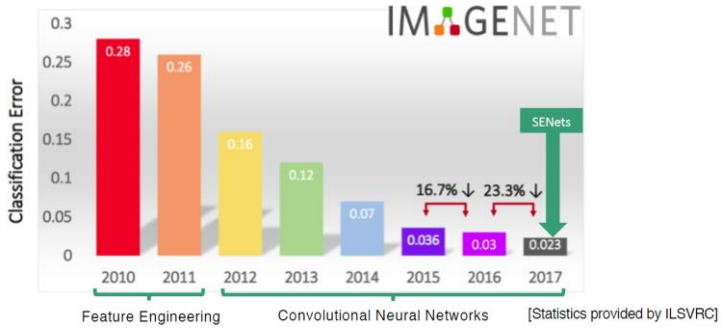


Figura 13: Evolución de Imagenet (*Review: SENet — Squeeze-and-Excitation Network, Winner of ILSVRC 2017 (Image Classification) | by Sik-Ho Tsang | Towards Data Science*, n.d.).

La detección de objetos en una imagen se puede realizar tanto en dos dimensiones como e tres, utilizando para ello los elementos descritos en el apartado de SLAM mediante visión por computador. La detección volumétrica de los elementos es realmente interesante para los robots autónomos, de esta manera pueden realizar las mediciones necesarias.

2.3.2. Segmentación

En adición a todo lo anterior, la segmentación es uno de los métodos más utilizados últimamente en los sistemas de navegación. Se puede decir que es una extensión a la detección de objetos, para que los sistemas computacionales entiendan mejor todo lo que están visualizando (Wolf et al., 2014).

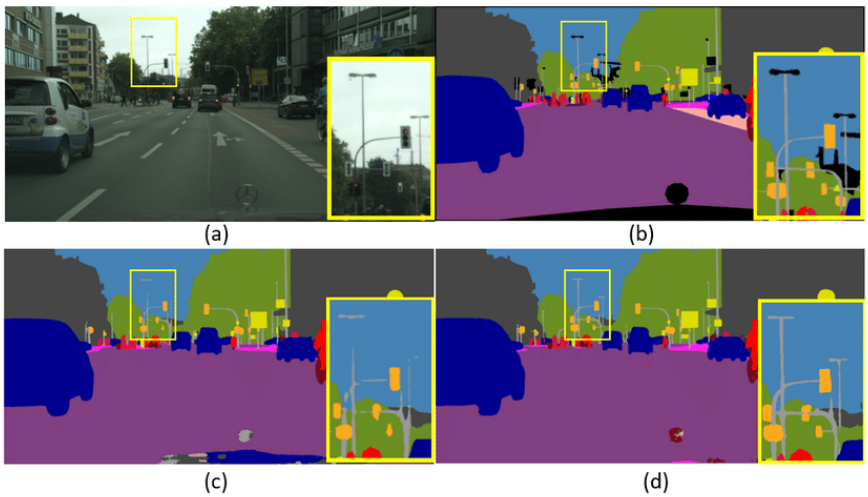


Figura 14: Imagen exterior segmentada (*Illustration of Challenges in Semantic Segmentation. (a), Input Image....* | *Download Scientific Diagram*, n.d.).

La segmentación puede ser realizada por diferentes tipos de algoritmos con los distintos sistemas vistos anteriormente. Se pueden ver sistemas monoculares (Y. Wang et al., 2020), sistemas que utilizan profundidad (Seichter et al., 2020), mapas creadas en base a la segmentación (Deng et al., 2020), para la navegación interior (Teso-Fz-Betoño et al., 2020) o para navegación exterior añadiendo detección de objetos (Feng et al., 2021).

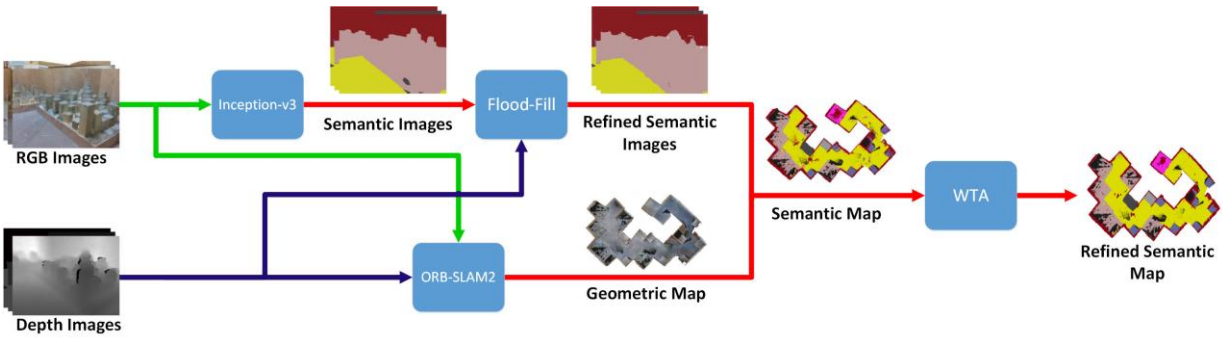


Figura 15: Mapa semántico para navegación (Deng et al., 2020).

Uno de los grandes problemas para los robots es el tiempo de ejecución de estos modelos (Li et al., 2020). Otra cosa tener en cuenta es que las personas no son elementos fijos en estos entornos, por lo que es necesario borrarlos para realizar mapas segmentadas (Ai et al., 2020). También es posible realizar el mapa segmentado mediante un sistema monocular que capta el entorno y lo segmenta (Miyamoto et al., 2020).

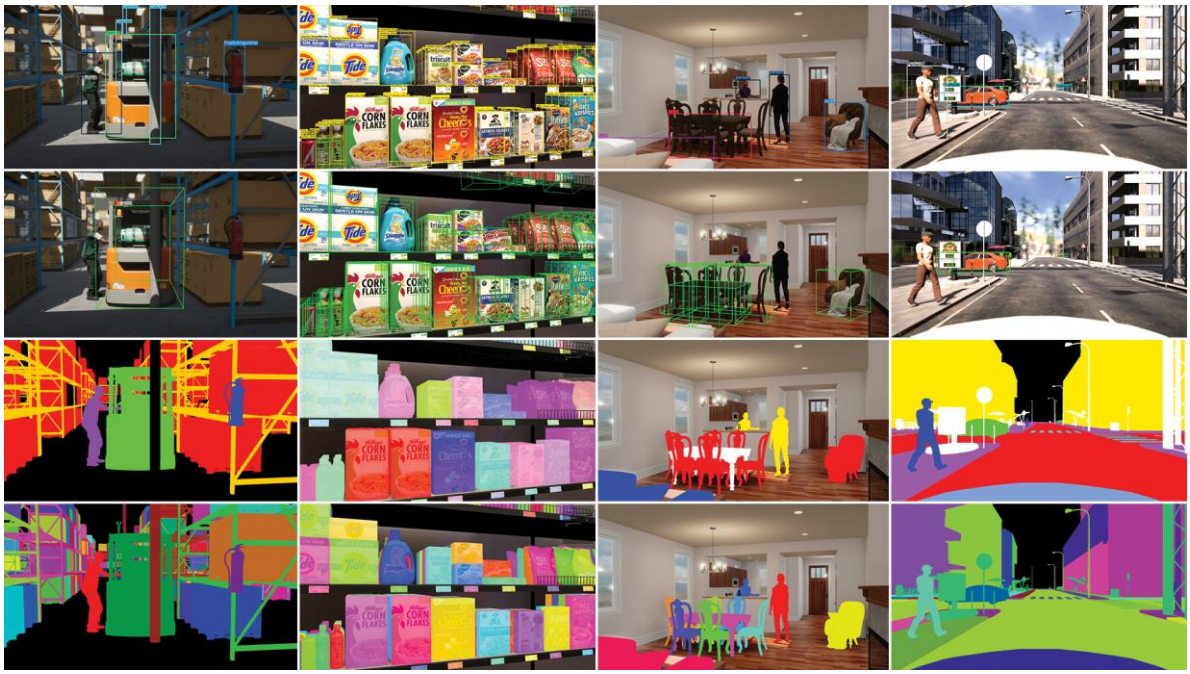


Figura 16: Detección de objetos y segmentación de una imagen (*Supercharge Your Computer Vision Models with Synthetic Datasets Built by Unity* | *Unity Blog*, n.d.).

Estos algoritmos de segmentación pueden ser entrenados de diferentes maneras, supervisado y no supervisado. La diferencia entre ellos es que el no supervisado (Jin et al., 2020), como bien dice el nombre, no se entrena o no realiza el entrenamiento con los resultados, en cambio el supervisado se entrena con el resultado para que puede obtener el mejor resultado posible.

2.3.2. Posición y trayectoria del ser humano

En todos los casos vistos anteriormente, es indispensable saber el estado del ser humano, donde se encuentra y cuál es su pose (Cui et al., 2020). Esto nos ayuda en evitar las colisiones o mantener distancias de seguridad por si el ser humano se mueve.

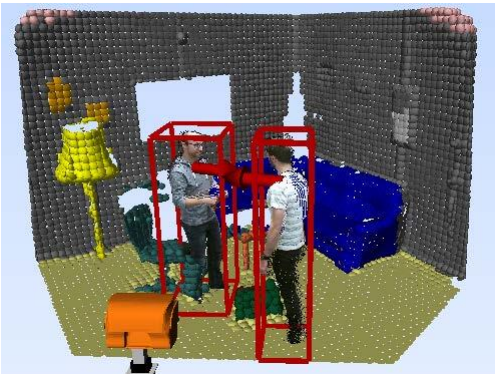


Figura 17: Estimación de posicionamiento de humanos en entorno segmentado (Seichter et al., 2020).

Para la eficiencia de un robot autónomo es de gran ayuda saber por dónde andarán los humanos (Tamaki et al., 2019) o también predecir por donde pueden ir para evitar esa trayectoria y elegir la más rápida posible (Liu et al., 2015), ayudando con la gesticulación y mirada del mismo (Moors et al., 2015).

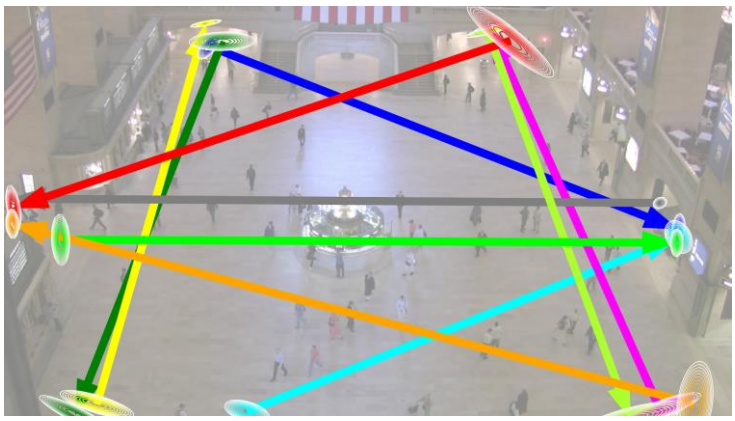


Figura 18: Segmentación de trayectorias después de procesado (Tamaki et al., 2019).

Para poder realizar todo lo mencionado con un buen funcionamiento se necesitan entender bien los entornos y entrenar los algoritmos de una manera eficaz a como se hace para cada lugar donde se encontrará el robot autónomo. Para ello es necesario tener bases de datos de distintos lugares como supermercados (Lewandowski et al., 2020), hospitales (H. M. Gross et al., 2017) o domésticos (H.-M. Gross et al., 2019).

Si queremos predecir lo que va ha hacer un ser humano, es importante tener en cuenta el tiempo, por lo que es necesario analizar lo que realiza un ser humano durante un tiempo determinado para poder predecir las próximas veces lo que pueda realizar.

Para poder definir bien el método de captación y realizar estas previsiones se necesitan las redes neuronales recurrentes, que serán los que analizarán el proceso de los movimientos para una previsión posterior de lo que pueda suceder.

Hay casos en los que se realiza la búsqueda de una trayectoria de las personas para que el robot pueda ayudar y con la previsión realizada minimizar el estorbo y el tiempo de espera del humano (Bruckschen et al., 2020). En este articulo se demuestra que los robots pueden predecir los movimientos de los humanos para no estorbar y además de eso, puede anteponerse a la situación que va a suceder.

Tal y como se puede ver en la Figura 19, la persona que se esta moviendo lleva alguna cosa en las manos. El robot detecta al humano y su movimiento, para después identificar el elemento que lleva en las manos y predecir para donde se va a dirigir la persona. Se puede ver la trayectoria del humano, esta trayectoria, la trayectoria de color morado, es la más probable y por esa causa, el robot evita realizar su navegación por ese lugar, la trayectoria de color blanco. Esa decisión la realiza para no estorbar a las personas y además de eso realiza otra búsqueda de navegación para poder llegar antes de tiempo al lugar deseado.

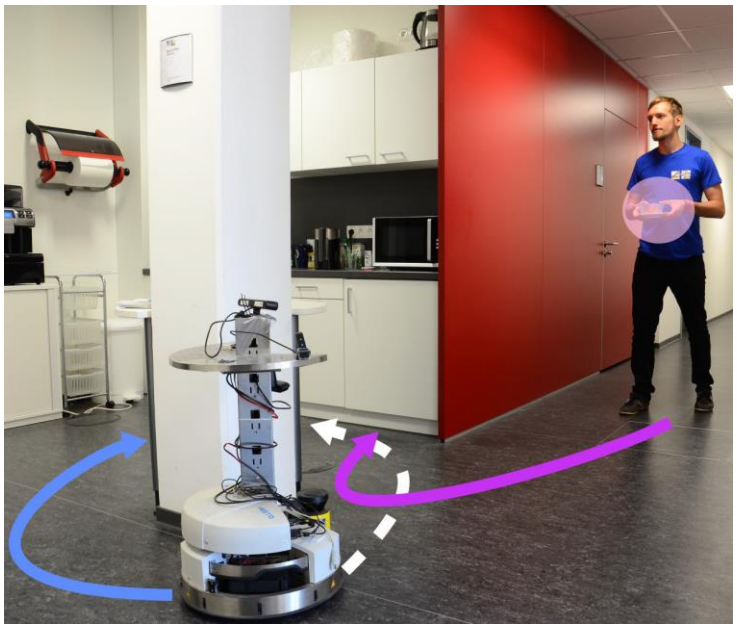


Figura 19. Detección de trayectoria para cálculo de navegación (Bruckschen et al., 2020).

Tal y como se ha realizado la búsqueda, también se puede hacer un mapa de movimientos realizados como se muestra en el artículo (Munaro & Menegatti, 2014). En este articulo se muestra que se pueden realizar los mapas según los movimientos de los humanos, por lo que

se crea un mapa de dos dimensiones con las trayectorias de las personas que han pasado por donde se encuentra el robot, se pueden ver los mapas creados en la Figura 20. Esto puede ayudar a la hora de crear un diseño personalizado de un sistema para prever los siguientes estados que se puedan generar.

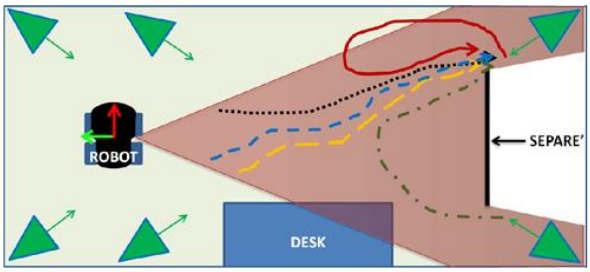


Figura 20. Se muestran las diferentes trayectorias de las personas.

2.4 Resumen

Para realizar un sistema completo, se ha visto que es necesario obtener la información del entorno de una forma correcta y precisa para poder aplicar los algoritmos o redes neuronales que se utilizan en aprendizaje automático. También es importante que pueda realizar la segmentación de interiores de una forma correcta con un índice de MIoU (mean intersection over unión) bastantemente admisible y que al mismo tiempo sea capaz de predecir lo que pueda suceder en un futuro, es decir, los movimientos del ser humano, se necesita una red neuronal con una arquitectura bastante compleja. Y el entrenamiento de esta red neuronal variara mucho según la base de datos utilizada o los hiperparámetros que se elijan, por lo que puede ser interesante realizar transfer learning a la hora de empezar a realizar el entrenamiento.

3. Objetivos y metodología de trabajo

Tras el análisis de los sistemas de captación y procesamiento para mejora y seguridad de los robots autónomos, es necesario definir qué es lo que se quiere hacer, como se pretende hacer y de qué manera se va a hacer para que se pueda realizar una valoración del proceso.

3.1. Objetivo general

El objetivo general es realizar la segmentación del entorno interior para que el robot tenga una visión más precisa al navegar y predecir las trayectorias de los humanos para proteger la integridad de ellos y aumentar la eficiencia a la hora de navegar utilizando para ello las técnicas de aprendizaje profundo.

3.2. Objetivos específicos

En este trabajo se van a realizar las siguientes tareas que ayudaran a describir mejor las especificaciones técnicas.

- Realizar análisis del estado del arte para sistemas de segmentación para navegación autónoma de robots y creación de SLAM
- Detección de personas y su estado de postura para asegurarnos de que no haya colisión entre robot y humano.
- Detectar movimiento en el sistema y predecir su trayectoria para poder esquivar y prevenir el paro del robot.
- Realizar o elegir el algoritmo de segmentación semántica que se ajuste a las necesidades.
- Elegir el algoritmo de navegación que se ajuste a las necesidades del sistema y probar si es la correcta.
- Juntar los objetivos anteriores y valorar si el sistema de predicción de trayectorias es eficiente.
- Implementar el resultado del proceso en la tarjeta de Nvidia Jetson Nano para poder probarlo con Jetbot, el robot creado para ello.
- Realizar la implementación en ROS.

3.3. Metodología del trabajo

Para poder llevar a cabo los objetivos descritos se establecerá una cierta estrategia. Esta estrategia o metodología será la denominada Scrum, que consiste en objetivo a corto plazo o esprints que por si se cambian los pequeños objetivos u ocurre algún cambio en el desarrollo. Se puede ver el ejemplo en la Figura 19. Esta metodología se utiliza mucho en los desarrollos software, que normalmente suelen ser complejos y pueden producirse cambios en cualquier momento durante el transcurso del proyecto.

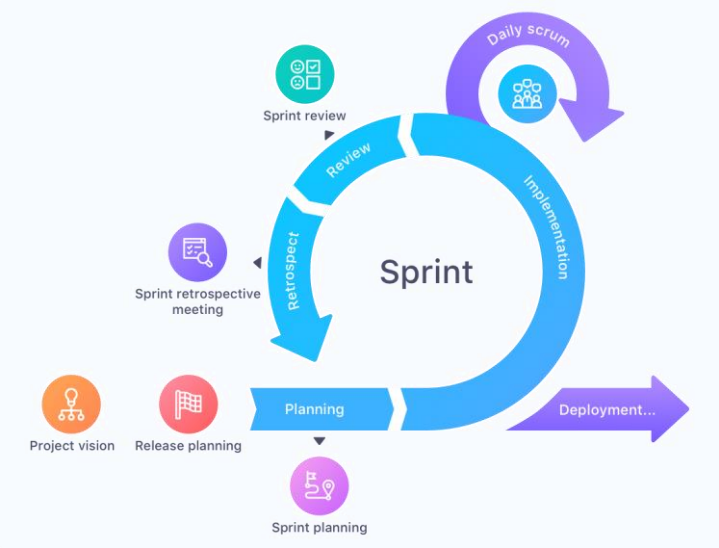


Figura 21: Ejemplo de la metodología Scrum (*SCRUM - ECLEE | European Center for Leadership and Entrepreneurship Education*, n.d.).

Para poder empezar a plantear esta metodología es necesario tener un proyecto y realizar una planificación o objetivos previos. Esta planificación no va a ser la definitiva, pero ayudara en tener un objetivo fijo y tener idea de que trata el proyecto, es decir, tener una visión clara del proyecto en general. Para poder realizar este planteamiento es necesario realizar el estudio del proyecto, analizar la información que existe y mirar si tienes recursos necesarios para poder llevar a cabo el proyecto.

Teniendo la información analizada y los recursos a disposición, se realizará el planteamiento general, que a su vez contendrá los pequeños esprints o objetivos que se marcaran para el cumplimiento del proyecto. Hay que tener claro de que cada esprint no está estrictamente definido, por lo que se realizara este proceso de Scrum cada vez que trabajemos en el proyecto, ya sea para repasar lo hecho hasta ese punto o realizar algún cambio que se ha detectado.

Cada esprint consistirá en distintas tareas que habrá que llevarlas a cabo en un tiempo determinado, siendo este tiempo flexible. Una vez realizada la tarea, es conveniente que sea subida a una plataforma de desarrollo colaborativo para que los integrantes puedan analizar lo que se ha hecho. En este caso al ser un solo integrante no es necesario subir, pero es interesante para llevar un control de versiones.

Puede ser que cada esprint tenga sus apartados o subtareas a realizar. Por ello en cada sprint o subtarea, según donde se encuentre se analizará lo que se ha realizado, como se ha realizado y si existen otras maneras de abordar el problema para poder buscar mejoras para del proyecto.

En este caso los objetivos serán los esprints generales que se han definido y estos tendrán una fecha específica para cumplir el plazo de entrega. Como se comenta anteriormente se evaluará y valorará el trabajo realizado y respecto a esa evaluación se realizarán las modificaciones necesarias para que el proyecto avance.

Para una mejor gestión de este sistema se realizará el concepto de Kanban para que en todo momento se sepa el estado del proyecto. En Kanban se encontrarán los esprints que contendrán las subtareas y en todo momento se podrá saber el estado de cada sprint.

Podemos ver la Tabla 3 donde se definen los esprints del proyecto.

Tabla 3: Planificación de esprints para el proyecto

ESPRINT	OBJETIVO	FECHA ESTIMADA
1	Análisis o estado del arte	6/05/2021
2	Realizar o elegir algoritmo de segmentación	20/05/2021
3	Realizar la detección de personas y sus trayectorias	30/05/2021
4	Elegir el algoritmo de navegación	02/06/2021
5	Juntar los algoritmos para implementarlos en Jetbot	10/06/2021
6	Implementación de los algoritmos en el Sistema ROS	20/06/2021
7	Pruebas para comprobar y calificar el funcionamiento	24/06/2021

4. Identificación de requisitos

Para el desarrollo del problema indicado, la segmentación semántica y predicción de movimiento de personas para una mejor navegación de los robots autónomos, evitando de esta manera que el robot intervenga en la trayectoria del ser humano. Se considera útil definir bien el problema para su mejor entendimiento, donde se identifican las necesidades para realizar el desarrollo software y hardware de la herramienta.

Tabla 4. Tabla de requisitos

Función	**Resultado**	**Ejemplo**
STL del robot	Dibujo en 3D	Introducir el STL en Gazeo, para simulación.
Detección de camino libre	Valor booleano: - Si - No	Si se detecta suelo en la segmentación: Si Si se no detecta suelo en la segmentación: No
Detección de personas	Valor booleano: - Si - No	Si se detecta persona: Si Si se no detecta persona: No
Detección de trayectoria de persona	Dibujar trayectoria en la segmentacion	
Navegación	Se toman las decisiones necesarias para que el robot pueda llegar al lugar indicado. - Elegir la ruta - Cambio de ruta	

Los requisitos del sistema se muestran en la Tabla 4, donde se muestran las funciones iniciales requeridos. Es necesario tener un algoritmo de segmentación semántica para diferenciar los distintos elementos que se encuentran en el entorno y definir bien el camino que se encuentra a disposición del robot. Para realizar la segmentación, se requiere de un modelo de aprendizaje profundo que debe ser capaz de diferenciar los elementos en colores para su posterior tratamiento de navegación. Debe tener un sistema o algoritmo de detección de personas que, al mismo tiempo, en caso de detectar una persona realice el seguimiento

para predecir su trayectoria. Para la predicción de trayectoria también se requiere de un modelo de aprendizaje profundo que prediga los pasos que puede dar esa persona y dibujar en el mapa el camino hacia donde pueda moverse. Al mismo tiempo que se realiza la segmentación y la predicción de la trayectoria, el robot tiene que navegar. Para la navegación es necesario utilizar algoritmos de búsqueda y arboles de comportamiento.

Los diferentes tipos de algoritmos van a ser analizados y en caso de que existan o se puedan utilizar opciones externas que realicen el trabajo, se analizan y se elige el que mejor se adecua al sistema utilizado. Las especificaciones de los algoritmos se encuentran en la Tabla 5.

Tabla 5. Técnicas de la herramienta

Etapa	**Algoritmo/Técnica**	**Fuente**
Segmentación semántica RGBD.	Entrenamiento de una red neuronal profundo con encoder y decoder. Que realizara la segmentación en base a 2 entradas, RGB y profundidad.	ESANet.
Reconocimiento y detección de trayectoria de personas.	Entrenamiento de una red neuronal profundo con encoder con LSTM y decoder. Que realizara la segmentación en base a 2 entradas, RGB y profundidad.	Autoria propia.
Navegación y toma de decisiones	Algoritmo de búsqueda para que un robot se mueva de un punto A a un punto B.	Nav2 (ROS Navigaiton Stack)

Es Imprescindible que la información recibida por la cámara se procese de forma adecuada para que el sistema de navegación pueda realizar un desempeño correcto a la hora de realizar la búsqueda. Por eso es por lo que se a escogido un algoritmo de segmentación realizada por expertos y que tiene por ahora los mejores resultados en código abierto. Además de eso, se decide utilizar este algoritmo porque trabajo con capturas de imágenes con profundidad para un mejor funcionamiento.

En cuanto a la detección de la trayectoria de las personas, es necesario definir cual es el resultado que se quiere obtener, utilizando para ello la profundidad como medidor de distancia. De esta manera, con el procesado de la ejecución de secuencias, se puede predecir la dirección de la persona o las personas identificadas.

5. Descripción de la herramienta software desarrollada

El presente proyecto consiste en avanzar la segmentación semántica y contribuir en el apartado de previsión de los movimientos de las personas para un mejor funcionamiento de la navegación autónoma del robot utilizando el software libre ROS para facilitar el uso del software creado. Para la implementación del sistema se ha escogido la tarjera NVIDIA Jetson Nano con el que muchos desarrolladores realizan sus propios robots o prototipos de robótica, sus especificaciones se indican en la Tabla 6.

Para el apartado de visión, se utiliza la cámara Intel-Realsense D435i que permite realizar capturas en color y profundidad para detectar elementos del entorno y las distancias a las que se encuentra, sus especificaciones indicadas en la Tabla 7. De esta manera se mide la distancia a la que se encuentra el humano para su posterior análisis secuencial para poder predecir el movimiento futuro.

La herramienta desarrollada en Python utiliza algunas librerías externas para el procesamiento de imágenes en el entrenamiento y también para la creación de las redes neuronales para su posterior tratamiento.

Lo ideal es tener un solo sistema que realice la segmentación y la predicción del movimiento, pero al no tener una base de datos con la segmentación y los movimientos segmentados todo en uno, se ha decidido realizar dos sistemas diferentes que funciones en paralelo.

Tabla 6. Especificaciones de la tarjeta Jetson Nano

GPU	Arquitectura NVIDIA Maxwell™ con 128 núcleos NVIDIA CUDA®
CPU	Procesador ARM® Cortex®-A57 MPCore de cuatro núcleos
Memoria	LPDDR4 de 4 GB y 64 bits
Almacenamiento	16 GB de almacenamiento Flash eMMC 5.1
Codificación de vídeo	4K a 30 cuadros (H.264/H.265)
Descodificación de vídeo	4K a 60 cuadros (H.264/H.265)
Cámara	12 vías (3 x 4 o 4 x 2) MIPI CSI-2 DPHY 1.1 (18 Gbps)
Conectividad	Gigabit Ethernet
Pantalla	HDMI 2.0 o DP 1.2 | eDP 1.4 | DSI (1 x 2) 2 simultáneos
UPHY	1 1/2/4 PCIE, 1 USB 3.0, 3 USB 2.0
E/S	1 SDIO / 2 SPI / 4 I2C / 2 I2S / GPIO
Tamaño	69,6 mm x 45 mm
Mecánicas	Conector de 260 pines

Tabla 7. Especificaciones de la cámara Intel-Realsense D435i

Features	**Use environment:**Indoor/Outdoor **Ideal range:**.3 m to 3 m **Image sensor technology:**Global Shutter
Depth	**Depth technology:**Active IR Stereo **Minimum depth distance (Min-Z) at max resolution:**~28 cm **Depth Accuracy:**<2% at 2 m[1] **Depth Field of View (FOV):**87° × 58° **Depth output resolution:**Up to 1280 × 720 **Depth frame rate:**Up to 90 fps
RGB	**RGB frame resolution:**1920 × 1080 **RGB frame rate:**30 fps **RGB sensor FOV (H × V):**69° × 42° **RGB sensor resolution:**2 MP **RGB sensor technology:**Rolling Shutter
Major Components	**Camera module:**Intel RealSense Module D430 + RGB Camera **Vision processor board:**Intel RealSense Vision Processor D4
Physical	**Form factor:**Camera Peripheral **Length × Depth × Height:**90 mm × 25 mm × 25 mm **Connectors:**USB-C* 3.1 Gen 1* **Mounting mechanism:**– One 1/4-20 UNC thread mounting point. – Two M3 thread mounting points.

5.1. Arquitectura del hardware

El robot con el que se realizan todas las pruebas es Jetbot, el robot educacional creado por los desarrolladores de NVIDIA. Este robot consta de la tarjeta Jetson Nano mencionada anteriormente, con una cámara monocular que se ha decidido cambiar por la cámara Intel-Realsense D435i para un mejor funcionamiento del sistema de segmentación. También contiene dos motores con su controlador de puente H para poder hacer el control de velocidad de estos motores. Se puede ver el Jetbot configurado a nuestra manera en la Figura 22.

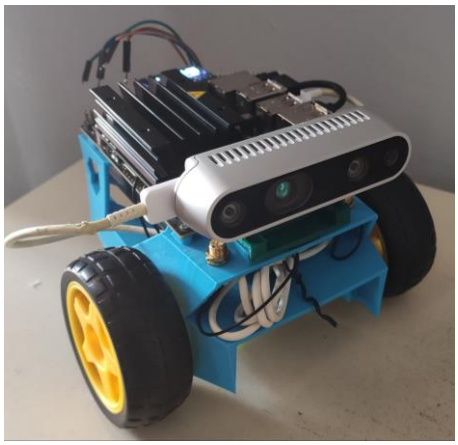


Figura 22. Imagen de Jetbot con la camara integrada.

El procesador Jetson Nano tiene instalado el sistema operativo Ubuntu 18.04 LST para poder utilizarlo como un ordenador. Esta tarjeta se utiliza modo embebido para un funcionamiento más rápido. En caso de utilizar sistemas en la nube hay que tener en cuenta que el robot debe tener en todo momento buena conexión a internet, por lo que se descarta esa opción.

Al tener instalado el sistema operativo Ubuntu, se pueden instalar paquetes adicionales para su mejor funcionamiento. Por eso, se instalan los paquetes de ROS necesarios. Algunos de los paquetes instalados son los que en la actualidad no se soportan en Ubuntu 18, pero los desarrolladores han puesto contenedores Docker que se pueden utilizar para la instalación de cualquier versión de ROS que tenga soporte.

5.2. Utilización de la cámara Intel-Realsense

Las cámaras creadas para la visión RGB-D suelen tener su propio funcionamiento y adquieren los datos de una manera particular. Por ello se va a utilizar la librería del fabricante pyrealsense2, especializada en capturas de la cámara por separado, por un lado, la captura RGB y por otro la profundidad o directamente la captura de la imagen en RGB-D.

Esta cámara adquiere también información sobre la inercia para detectar la posición en la que se encuentra el dispositivo en cada momento. Esta son dos variables, aceleración y giroscopio, que se adquieren también por la librería del fabricante.

Para realizar las pruebas de la velocidad de captura de imágenes se ha realizado un script en Python para medir el FPS (Frames Per Second) que mide cuantas fotos es capaz de sacar en un segundo.

Se han probado distintas dimensiones de captura de imágenes para saber a qué velocidad es posible realizar esta captura.

- Si ponemos la captura de las dimensiones en 1920x1080, podemos ver que la cámara RGB realiza la captura de 8 imágenes por segundo, pero la cámara infrarroja no puede detectar tantos pixeles ya que su máximo es 1920x1080. Los resultados se muestran en la Figura 23.

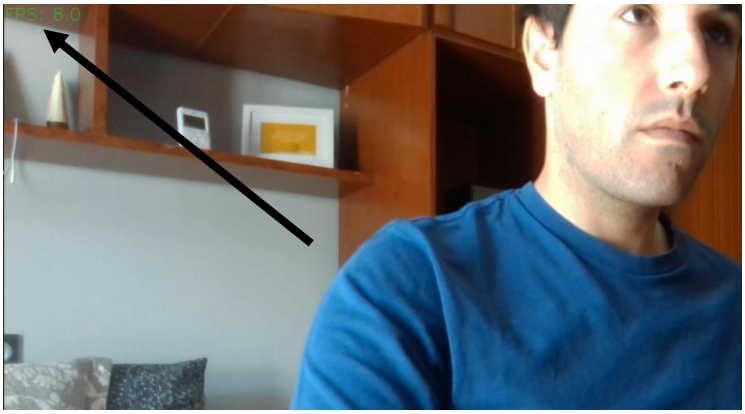


Figura 23. Imagen RGB de resolución 1920x1080

- Si ponemos la captura de las dimensiones en 1280x960, podemos ver que la cámara RGB realiza la captura de 15 imágenes por segundo, pero la cámara infrarroja realiza la captura de 6 imágenes por segundo. Los resultados se muestran en la Figura 24 y 25.

Figura 24. Imagen RGB de resolución 1280x960

Figura 25. Imagen de profundidad de resolución 1280x960

- Si ponemos la captura de las dimensiones en 640x480, podemos ver que la cámara RGB realiza la captura de 16 imágenes por segundo, pero la cámara infrarroja realiza la captura de 16 imágenes por segundo. Los resultados se muestran en la Figura 26 y 27.

Figura 26. Imagen RGB de resolución 640x480

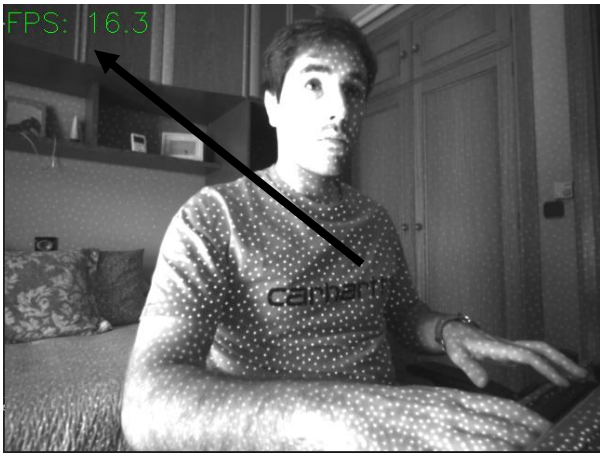


Figura 27. Imagen de profundidad de resolución 640x480

En las imágenes capturadas se muestra la velocidad de captura de imagen que obtenemos para las diferentes cámaras utilizando para ello la librería OpneCV. Pero existe la opción de

realizar lo mismo y más cosas utilizando la librería pyrealsense2 que proporciona la facilidad de realizar la adquisición de las distancias, cámara RGB, la imagen RGB-D y también los datos proporcionados por los sensores de inercia.

Pruebas pyrealsense

5.3. Elección del algoritmo de segmentación semántica RGB-D

Para elegir el algoritmo de segmentación se ha de realizar la búsqueda de algunos algoritmos de código abierto. Esto se hace por causa de que estas redes neuronales están bastante logradas y su creación y entrenamiento de una red neuronal es compleja y podría tardar mucho en entrenarse. Antes de realizar ninguna selección se ha realizado la búsqueda de diferentes algoritmos y se ha visto que los que obtienen mejores resultados son los siguientes:

- ESANet-R34-NBt1D.
- SA-Gate.
- RDFNet.
- SGNet.
- ACNet.

Los resultados de las pruebas de estos algoritmos se recogen del artículo de ESANet (Seichter et al., 2020), pero se han vuelto a realizar en el Jetson Nano asumiendo que los resultados obtenidos serán proporcionales en cuanto a exigencias de computación. Aun y todo, se realiza la prueba y según la asunción realizada, se escoge el que mejor resultados obtenga y sea más rápido en la captura de imágenes por segundo.

Para realizar las pruebas se han utilizado las bases de datos SunRGBD (Song et al., 2015) y NYUv2 (Silberman et al., 2012), que es la base de datos de entornos interiores. Se pueden utilizar otras bases de datos como Cityscapes (Cordts et al., 2016) para exteriores, habitualmente utilizados para vehículos autónomos o robots para exteriores.

Resultados y elección. Como no he podido instalar pycuda no ha sido posible realizar las pruebas en Jetson nano

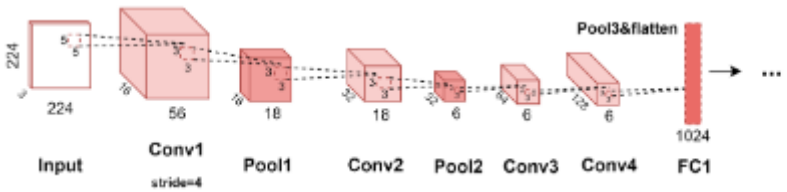
Estos algoritmos de segmentación suelen tener codificadores y descodificadores para que puedan funcionar correctamente. El codificador es una red neuronal convolucional normal, que realiza las convoluciones para la identificación de los elementos y sacar una imagen en

vectores. En cambio, de decodificador realiza la máscara de la imagen partiendo de la imagen vectorizada.

5.4 Entrenamiento de la red neuronal predictor de trayectoria

Elegir el estructura de red y probar los resultados utilizando solo la red entrenada con LSTM a la salida, con LSTM entre encoder y decoder y una red que realice el sistema independientemente solo con encoder o encoder-decoder.

Arquitectura de la red neuronal convolucional:



6. Evaluación

La evaluación debería cubrir por lo menos una mínima evaluación de la usabilidad de la herramienta, así como de su aplicabilidad para resolver el problema propuesto. Estas evaluaciones suelen realizarse con usuarios expertos.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

Este último capítulo (en ocasiones, dos capítulos complementarios) es habitual en todos los tipos de trabajos y presenta el resumen final de tu trabajo y debe servir para informar del alcance y relevancia de tu aportación.

Suele estructurarse empezando con un resumen del problema tratado, de cómo se ha abordado y de por qué la solución sería válida.

Es recomendable que incluya también un resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

7.2. Líneas de trabajo futuro

Finalmente, se suele dedicar una última sección a hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

8. Bibliografía

Ai, Y., Rui, T., Lu, M., Fu, L., Liu, S., & Wang, S. (2020). DDL-SLAM: A robust RGB-d SLAM in dynamic environments combined with deep learning. *IEEE Access*, *8*, 162335–162342. https://doi.org/10.1109/ACCESS.2020.2991441

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv*. http://arxiv.org/abs/2004.10934

Bruckschen, L., Bungert, K., Dengler, N., & Bennewitz, M. (2020). Human-aware robot navigation by long-term movement prediction. In *IEEE International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/IROS45743.2020.9340776

Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., Liu, Y., & Wu, Z. (2019). An Improved Faster R-CNN for Small Object Detection. *IEEE Access*, *7*, 106838–106846. https://doi.org/10.1109/ACCESS.2019.2932731

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2014). PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Transactions on Image Processing*, *24*(12), 5017–5032. https://doi.org/10.1109/TIP.2015.2475625

Chang, L., Niu, X., & Liu, T. (2020). Gnss/imu/odo/lidar-slam integrated navigation system using imu/odo pre-integration. *Sensors (Switzerland)*, *20*(17), 1–18. https://doi.org/10.3390/s20174702

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vols. 2016-Decem). https://doi.org/10.1109/CVPR.2016.350

Cui, X., Lu, C., & Wang, J. (2020). 3D Semantic Map Construction Using Improved ORB-SLAM2 for Mobile Robot in Edge Computing Environment. *IEEE Access*, *8*, 67179–67191. https://doi.org/10.1109/ACCESS.2020.2983488

Deng, W., Huang, K., Chen, X., Zhou, Z., Shi, C., Guo, R., & Zhang, H. (2020). Semantic RGB-D SLAM for Rescue Robot Navigation. *IEEE Access*, *8*, 221320–221329. https://doi.org/10.1109/ACCESS.2020.3031867

Feng, D., Haase-Schutz, C., Rosenbaum, L., Hertlein, H., Glaser, C., Timm, F., Wiesbeck, W.,

& Dietmayer, K. (2021). Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), 1341–1360. https://doi.org/10.1109/TITS.2020.2972974

Gross, H.-M., Scheidig, A., Muller, S., Schutz, B., Fricke, C., & Meyer, S. (2019). Living with a Mobile Companion Robot in your Own Apartment - Final Implementation and Results of a 20-Weeks Field Study with 20 Seniors*. *2019 International Conference on Robotics and Automation (ICRA)*, *2019-May*, 2253–2259. https://doi.org/10.1109/ICRA.2019.8793693

Gross, H. M., Meyer, S., Scheidig, A., Eisenbach, M., Mueller, S., Trinh, T. Q., Wengefeld, T., Bley, A., Martin, C., & Fricke, C. (2017). Mobile robot companion for walking training of stroke patients in clinical post-stroke rehabilitation. *Proceedings - IEEE International Conference on Robotics and Automation*, 1028–1035. https://doi.org/10.1109/ICRA.2017.7989124

Illustration of challenges in semantic segmentation. (a), Input Image.... | *Download Scientific Diagram*. (n.d.). Retrieved June 3, 2021, from https://www.researchgate.net/figure/Illustration-of-challenges-in-semantic-segmentation-a-Input-Image-b-Ground-Truth_fig1_332186435

Incorrect visualization of LiDAR · Issue #32 · carla-simulator/ros-bridge. (n.d.). Retrieved June 3, 2021, from https://github.com/carla-simulator/ros-bridge/issues/32

Intel. (2019). *Depth Camera D435 – Intel® RealSense™ Depth and Tracking Cameras*. Https://Www.Intelrealsense.Com/Depth-Camera-D435/. https://www.intelrealsense.com/depth-camera-d435i/%0Ahttps://www.intelrealsense.com/depth-camera-d435/%0Ahttps://www.intelrealsense.com/depth-camera-d435i/%0Ahttps://www.intelrealsense.com/depth-camera-d435/

Jin, S., Chen, L., Sun, R., & McLoone, S. (2020). A novel vSLAM framework with unsupervised semantic segmentation based on adversarial transfer learning. *Applied Soft Computing Journal*, *90*, 106153. https://doi.org/10.1016/j.asoc.2020.106153

Jiwon, J. (2019). *Computer Vision for Beginners: Part 4 - Towards Data Science*. https://towardsdatascience.com/computer-vision-for-beginners-part-4-64a8d9856208

Krombach, N., Droeschel, D., Houben, S., & Behnke, S. (2018). Feature-based visual odometry prior for real-time semi-dense stereo SLAM. *Robotics and Autonomous*

Systems, *109*, 38–58. https://doi.org/10.1016/j.robot.2018.08.002

Lewandowski, B., Wengefeld, T., Muller, S., Jenny, M., Glende, S., Schroter, C., Bley, A., & Gross, H.-M. (2020). Socially Compliant Human-Robot Interaction for Autonomous Scanning Tasks in Supermarket Environments. *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 363–370. https://doi.org/10.1109/RO-MAN47096.2020.9223568

Li, F., Li, W., Chen, W., Xu, W., Huang, L., Li, D., Cai, S., Yang, M., Xiong, X., & Liu, Y. (2020). A Mobile Robot Visual SLAM System with Enhanced Semantics Segmentation. *IEEE Access*, *8*, 25442–25458. https://doi.org/10.1109/ACCESS.2020.2970238

Liu, W., Chan, A. B., Lau, R. W. H., & Manochaieee, D. (2015). Leveraging long-term predictions and online learning in agent-based multiple person tracking. In *IEEE Transactions on Circuits and Systems for Video Technology* (Vol. 25, Issue 3). https://doi.org/10.1109/TCSVT.2014.2344511

Miyamoto, R., Adachi, M., Ishida, H., Watanabe, T., Matsutani, K., Komatsuzaki, H., Sakata, S., Yokota, R., & Kobayashi, S. (2020). Visual navigation based on semantic segmentation using only a monocular camera as an external sensor. In *Journal of Robotics and Mechatronics* (Vol. 32, Issue 6). https://doi.org/10.20965/jrm.2020.p1137

Moors, P., Germeys, F., Pomianowska, I., & Verfaillie, K. (2015). Perceiving where another person is looking: The integration of head and body information in estimating another person's gaze. *Frontiers in Psychology*, *6*(JUN). https://doi.org/10.3389/fpsyg.2015.00909

Mu, L., Yao, P., Zheng, Y., Chen, K., Wang, F., & Qi, N. (2020). Research on SLAM algorithm of mobile robot based on the fusion of 2D LiDAR and depth camera. *IEEE Access*, *8*, 157628–157642. https://doi.org/10.1109/ACCESS.2020.3019659

Munaro, M., & Menegatti, E. (2014). Fast RGB-D people tracking for service robots. *Autonomous Robots*, *37*(3), 227–242. https://doi.org/10.1007/s10514-014-9385-0

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, *31*(5), 1147–1163. https://doi.org/10.1109/TRO.2015.2463671

ORB SLAM Proposal for NTU GPU Programming Course 2016. (n.d.). Retrieved June 3, 2021, from https://www.slideshare.net/MindosCheng/orb-slam-proposal-for-ntu-gpu-

programming-course-2016

Principle drawing of a stereo camera setup. Objects (1,2) in various... | Download Scientific Diagram. (n.d.). Retrieved June 3, 2021, from https://www.researchgate.net/figure/Principle-drawing-of-a-stereo-camera-setup-Objects-1-2-in-various-depth-ranges-are_fig5_303307354

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 39, Issue 6). https://doi.org/10.1109/TPAMI.2016.2577031

Review: SENet — Squeeze-and-Excitation Network, Winner of ILSVRC 2017 (Image Classification) | by Sik-Ho Tsang | Towards Data Science. (n.d.). Retrieved June 3, 2021, from https://towardsdatascience.com/review-senet-squeeze-and-excitation-network-winner-of-ilsvrc-2017-image-classification-a887b98b2883

SCRUM - ECLEE | European Center for Leadership and Entrepreneurship Education. (n.d.). Retrieved June 3, 2021, from https://www.eclee.com/training/scrum/

Seichter, D., Köhler, M., Lewandowski, B., Wengefeld, T., & Gross, H.-M. (2020). Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. *ArXiv*. http://arxiv.org/abs/2011.06961

Shin, Y.-S., Yeong, ·, Park, S., & Kim, A. (2020). DVL-SLAM: sparse depth enhanced direct visual-LiDAR SLAM. *Autonomous Robots*, *44*, 115–130. https://doi.org/10.1007/s10514-019-09881-0

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from RGBD images. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 7576 LNCS* (Issue PART 5). https://doi.org/10.1007/978-3-642-33715-4_54

Song, S., Lichtenberg, S. P., & Xiao, J. (2015). SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Vols. 07-12-June). https://doi.org/10.1109/CVPR.2015.7298655

Stereo visual SLAM system overview: First, we undistort and rectify the... | Download Scientific Diagram. (n.d.). Retrieved June 3, 2021, from https://www.researchgate.net/figure/Stereo-visual-SLAM-system-overview-First-we-

undistort-and-rectify-the-stereo-images_fig1_257523126

Supercharge your computer vision models with synthetic datasets built by Unity | Unity Blog. (n.d.). Retrieved June 3, 2021, from https://blog.unity.com/technology/supercharge-your-computer-vision-models-with-synthetic-datasets-built-by-unity

Tamaki, T., Ogawa, D., Raytchev, B., & Kaneda, K. (2019). Semantic segmentation of trajectories with improved agent models for pedestrian behavior analysis. *Advanced Robotics*, *33*(3–4), 153–168. https://doi.org/10.1080/01691864.2018.1554508

Teso-Fz-Betoño, D., Zulueta, E., Sánchez-Chica, A., Fernandez-Gamiz, U., & Saenz-Aguirre, A. (2020). Semantic segmentation to develop an indoor navigation system for an autonomous mobile robot. *Mathematics*, *8*(5). https://doi.org/10.3390/MATH8050855

Wang, R., Zhang, W., Shi, Y., Wang, X., & Cao, W. (2019). GA-ORB: A New Efficient Feature Extraction Algorithm for Multispectral Images Based on Geometric Algebra. *IEEE Access*, *7*, 71235–71244. https://doi.org/10.1109/ACCESS.2019.2918813

Wang, Y., Chen, Q., Chen, S., & Wu, J. (2020). Multi-Scale Convolutional Features Network for Semantic Segmentation in Indoor Scenes. *IEEE Access*, *8*, 89575–89583. https://doi.org/10.1109/ACCESS.2020.2993570

Wolf, D., Bajones, M., Prankl, J., & Vincze, M. (2014). *Find my mug: Efficient object search with a mobile robot using semantic segmentation*. http://arxiv.org/abs/1404.5765

Xu, D., & Wu, Y. (2020). Improved YOLO-V3 with densenet for multi-scale remote sensing target detection. *Sensors (Switzerland)*, *20*(15), 1–24. https://doi.org/10.3390/s20154276

Zaarane, A., Slimani, I., Al Okaishi, W., Atouf, I., & Hamdoun, A. (2020). Distance measurement system for autonomous vehicles using stereo camera. *Array*, *5*, 100016. https://doi.org/10.1016/j.array.2020.100016

ZED 2 - AI Stereo Camera | Stereolabs. (n.d.). Retrieved June 3, 2021, from https://www.stereolabs.com/zed-2/

Zhang, S., Zheng, L., & Tao, W. (2021). Survey and Evaluation of RGB-D SLAM. *IEEE Access*, *9*, 21367–21387. https://doi.org/10.1109/ACCESS.2021.3053188

Anexos

Cuestionarios, encuestas, resultados de pilotos, documentos adicionales, capturas de pantalla, etc.

Además, al final de la memoria y como un anexo obligatorio deberá incluirse un artículo de investigación que resuma el trabajo realizado y los principales resultados obtenidos.

Anexo I. Título del anexo I

Anexo I.

Anexo II. Título del anexo II

Anexo II.

Anexo. Artículo de investigación

Al final de la memoria y como un anexo obligatorio deberá incluirse un artículo de investigación que resuma el trabajo realizado y los principales resultados obtenidos. Este artículo deberá seguir la plantilla proporcionada a continuación y tendrá una extensión de entre 6 y 8 páginas. El artículo se podrá desarrollar en español o en inglés (para ello utilizar la plantilla adecuada).

Título

Nombre y Apellidos del Estudiante

Universidad Internacional de la Rioja, Logroño (España)

Fecha

Resumen

Breve resumen del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Palabras Clave

Tres a cinco palabras clave ordenadas alfabéticamente y separadas por comas.

I. Introducción

Introducción en la que debes resumir de forma esquemática pero suficientemente clara lo esencial de cada una de las partes del trabajo.

La lectura de esta introducción ha de dar una primera idea clara de lo que se pretendía, las conclusiones a las que se ha llegado y del procedimiento seguido.

II. Estado del Arte

Estudio a fondo el dominio de aplicación, citando numerosas referencias.

Debe aportar un buen resumen del conocimiento que ya existe en el campo de los problemas habituales identificados.

Numerar las citas de forma consecutiva entre corchetes [1].

III. Objetivos y Metodología

Objetivo general, objetivos específicos y metodología de trabajo aplicada.

IV. Contribución

Desarrollar la descripción de tu contribución.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

V. Evaluación y Resultados

Descripción de la evaluación y los resultados obtenidos (Tipo 2. Desarrollo de Software).

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

Evaluación 1

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

En la Figura 1…

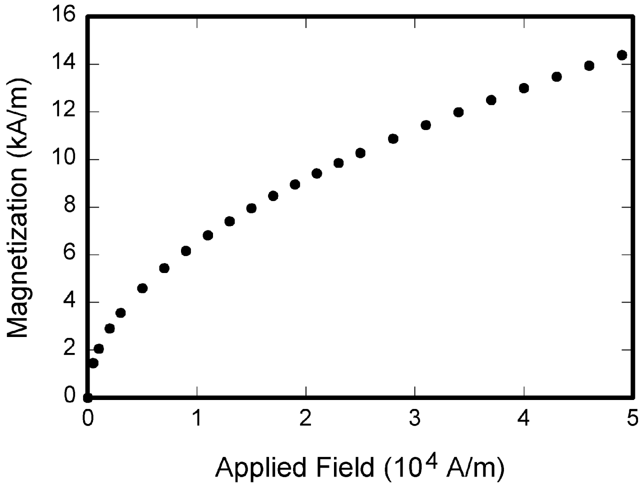


Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

En la Tabla I …

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Evaluación 2

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX

VI. Discusión

Tras la presentación objetiva de los resultados, querrás aportar una discusión de los mismos.

VII. Conclusiones

Resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

Finalmente, hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

Apéndices

Apéndices, en caso de ser necesario.

Referencias

[1] G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

Tabla I

Units for Magnetic Properties

Symbol	Quantity	Conversion from Gaussian and CGS EMU to SI [a]
Φ	magnetic flux	1 Mx $\rightarrow 10^{-8}$ Wb = 10^{-8} V·s
B	magnetic flux density, magnetic induction	1 G $\rightarrow 10^{-4}$ T = 10^{-4} Wb/m^2
H	magnetic field strength	1 Oe $\rightarrow 10^3/(4\pi)$ A/m
m	magnetic moment	1 erg/G = 1 emu $\rightarrow 10^{-3}$ A·m^2 = 10^{-3} J/T
M	magnetization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 10^3$ A/m
4πM	magnetization	1 G $\rightarrow 10^3/(4\pi)$ A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g $\rightarrow$ 1 A·m^2/kg
j	magnetic dipole moment	1 erg/G = 1 emu $\rightarrow 4\pi \times 10^{-10}$ Wb·m
J	magnetic polarization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 4\pi \times 10^{-4}$ T
χ, κ	susceptibility	$1 \rightarrow 4\pi$
χ_ρ	mass susceptibility	1 cm^3/g $\rightarrow 4\pi \times 10^{-3}$ m^3/kg
μ	permeability	$1 \rightarrow 4\pi \times 10^{-7}$ H/m = $4\pi \times 10^{-7}$ Wb/(A·m)
μ_r	relative permeability	$\mu \rightarrow \mu_r$
w, W	energy density	1 erg/cm^3 $\rightarrow 10^{-1}$ J/m^3
N, D	demagnetizing factor	$1 \rightarrow 1/(4\pi)$

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

[a]Gaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

Title

Name and Surname of the Student

Universidad Internacional de la Rioja, Logroño (España)

Date

Abstract

Breve resumen del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Keywords

Tres a cinco palabras clave ordenadas alfabéticamente y separadas por comas.

I. Introduction

Introducción en la que debes resumir de forma esquemática pero suficientemente clara lo esencial de cada una de las partes del trabajo.

La lectura de esta introducción ha de dar una primera idea clara de lo que se pretendía, las conclusiones a las que se ha llegado y del procedimiento seguido.

II. State of the Art

Estudio a fondo el dominio de aplicación, citando numerosas referencias.

Debe aportar un buen resumen del conocimiento que ya existe en el campo de los problemas habituales identificados.

Numerar las citas de forma consecutiva entre corchetes [1].

III. Objectives and Methodology

Objetivo general, objetivos específicos y metodología de trabajo aplicada.

IV. Contribuiton

Desarrollar la descripción de tu contribución.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

V. Evaluation and Results

Descripción de la evaluación y los resultados obtenidos (Tipo 2. Desarrollo de Software).

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

Evaluación 1

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

En la Figura 1…

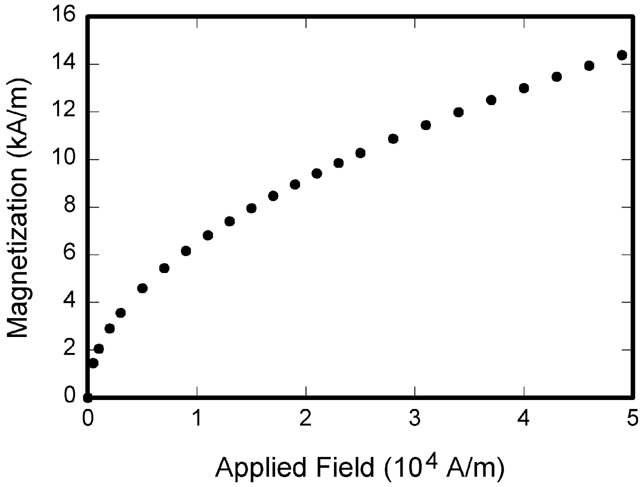


Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

En la Tabla I …

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Evaluación 2

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX

VI. Discussion

Tras la presentación objetiva de los resultados, querrás aportar una discusión de los mismos.

VII. Conclusion

Resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

Finalmente, hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

Appendix

Apéndices, en caso de ser necesario.

References

[3] G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[4] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

Tabla I

Units for Magnetic Properties

Symbol	Quantity	Conversion from Gaussian and CGS EMU to SI [a]
Φ	magnetic flux	1 Mx $\rightarrow 10^{-8}$ Wb = 10^{-8} V·s
B	magnetic flux density, magnetic induction	1 G $\rightarrow 10^{-4}$ T = 10^{-4} Wb/m^2
H	magnetic field strength	1 Oe $\rightarrow 10^3/(4\pi)$ A/m
m	magnetic moment	1 erg/G = 1 emu $\rightarrow 10^{-3}$ A·m^2 = 10^{-3} J/T
M	magnetization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 10^3$ A/m
4πM	magnetization	1 G $\rightarrow 10^3/(4\pi)$ A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g $\rightarrow$ 1 A·m^2/kg
j	magnetic dipole moment	1 erg/G = 1 emu $\rightarrow 4\pi \times 10^{-10}$ Wb·m
J	magnetic polarization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 4\pi \times 10^{-4}$ T
χ, κ	susceptibility	1 $\rightarrow 4\pi$
χ_ρ	mass susceptibility	1 cm^3/g $\rightarrow 4\pi \times 10^{-3}$ m^3/kg
μ	permeability	1 $\rightarrow 4\pi \times 10^{-7}$ H/m = $4\pi \times 10^{-7}$ Wb/(A·m)
μ_r	relative permeability	$\mu \rightarrow \mu_r$
w, W	energy density	1 erg/cm^3 $\rightarrow 10^{-1}$ J/m^3
N, D	demagnetizing factor	1 $\rightarrow 1/(4\pi)$

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

[a]Gaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.