

1. -----

Título: Aplicación de reinforcement learning al control de un vehículo autónomo.

Explicación: Diseño de algoritmos de control de vehículo autónomo basado en reinforcement learning y simuladores.

Otros involucrados: Angel Madridano, Abdulla Al Kaff, Martín Palos.

Dificultad: alta

Competencias: Linux, Python, ROS.

Responsable: Fernando Garcia, fegarcia@ing.uc3m.es

2. -----

Título: Detección en imagen médica basada en segmentación semántica .

Explicación: Diseño de algoritmos de detección de patologías en imagen médica empleando datasets públicos, tipo Kaggle .

Dificultad: media

Competencias: Linux, Python, ROS, OpenCV

Responsable: Fernando Garcia, fegarcia@ing.uc3m.es

3. -----

Título: Detección de sueño al volante usando CNNs

Explicación: Diseño de algoritmo de detección de somnolencia, basado en redes neuronales convolucionales.

Dificultad: media

Competencias: Linux, Python, ROS, OpenCV

Responsable: Fernando Garcia, fegarcia@ing.uc3m.es

4. -----

Título: Monitorización de tráfico de vehículos en entorno controlado mediante radares fijos usando simulador

Explicación: Diseño de algoritmo de detección de vehículos en un entorno controlado, empleando simuladores tipo CARLA o similares, para evaluación de comportamiento.

Dificultad: Baja

Competencias: Linux, Python, ROS, OpenCV

Responsable: Fernando Garcia, fegarcia@ing.uc3m.es

5. -----

Título: Detección y análisis de estado de contenedores basado en redes neuronales.

Explicación: Diseño de algoritmo de detección de contenedores y el estado de los mismos , basado en redes neuronales convolucionales.

Otros involucrados: Sergio Campos.

Dificultad: media

Competencias: Linux, Python, ROS, OpenCV

Responsable: Abdulla Al Kaff, akaff@ing.uc3m.es

6. -----

Título: Generación de entornos 3D digitales utilizando información LiDAR para su utilización en el gemelo digital de un coche autónomo.

Explicación: Generar entornos 3D digitales, réplicas de entornos reales, utilizando datos recopilados del LiDAR de un vehículo autónomo real, con el fin de utilizar estos entornos para realizar ensayos concluyentes con un gemelo digital dentro de un entorno simulado.

Dificultad: Medio

Competencias: C++ o Python (necesario) + Blender, Meshlab o algún otro programa de modelado 3D (muy recomendado, pero no necesario) + ROS (recomendado, pero no necesario) + Linux (recomendado, pero no necesario).

Responsable: Martín Palos Lorite, mpalos@ing.uc3m.es.

7. -----

Título: Desarrollo de un sistema IoT de vigilancia y control para impresoras 3D.

Explicación: Implementación del software necesario así como la arquitectura de comunicación que permita la supervisión y el control de forma remota de una impresora 3D. También se incluirá el diseño y construcción de todos los elementos que sean necesarios utilizando la propia impresora.

Otros involucrados: Dr. Abdulla Al-Kaff, Dr. Ángel Madridano

Dificultad: Media

Competencias: Sistemas operativos basados en Linux (recomendable), modelado 3D (recomendable), impresión 3D (recomendable).

Responsable: Sergio Campos Novoa, secampos@ing.uc3m.es

8. -----

Título: Sistema de sensores interconectado para la monitorización de tráfico

Explicación: Desarrollar un algoritmo capaz de tomar imágenes desde varias cámaras RGB, detectar vehículos y peatones, realizar un tracking 2D sobre imagen, proyectar en el plano los objetos, hacer match entre varias cámaras y mandar alertas en caso de situaciones con riesgo de colisión.

Otros involucrados: Dr. Abdulla Al-kaff, Martín Palos

Dificultad: media-alta

Prioridad: media

Competencias: Linux, Python, ROS, OpenCV, (Arduino)

Responsable: Armando Astudillo Olalla, aastudil@ing.uc3m.es

9. -----

Título (TFM): Measuring your measurements

Explicación: Desarrollar un método de machine learning capaz de aprender el modelo de medición (sensor) de cualquier algoritmo genérico de detección de objetos.

Ejemplo de aplicación: Tenemos un algoritmo de detección de landmarks (basado en LiDAR), pero necesitamos saber cómo de buenas son esas detecciones (modelar el error) para utilizarlas en un módulo de localización. Es probable que el error de detección dependa de la distancia al objeto, su tamaño, etc. Este algoritmo también puede utilizarse con cualquier otro algoritmo de detección de objetos si se requiere una estimación del error de las detecciones.

La idea principal es construir un entorno de simulación donde podamos obtener datos con distintas medidas de los objetos a detectar en diferentes posiciones. Los datos obtenidos se utilizarán para que el algoritmo aprenda el modelo estadístico del error en las detecciones.

Dificultad: Muy alta

Competencias: Linux, Entornos de simulación (Gazebo o Carla), python, estadística y machine learning (regresiones / redes neuronales).

Responsable: Francisco Miguel Moreno, franmore@ing.uc3m.es .

10. -----

Título: Algoritmo de localización basada en landmarks para vehículos autónomos.

Explicación: Implementar un algoritmo en C++ que resuelva el sistema no lineal de mínimos cuadrados (NLLS) producido por un conjunto de múltiples detecciones de landmarks; con el fin de estimar la posición del vehículo desde donde se hicieron las detecciones. El algoritmo debe ser capaz de funcionar a un mínimo de 10Hz, incluso con un alto número de landmarks (hasta 15). Se recomienda utilizar una librería de optimización como g2o para evitar reinventar la rueda. Una versión preliminar básica de este algoritmo de localización ya ha sido implementada en python utilizando la librería lmfit, y se proporcionará como ejemplo para implementar la versión más eficiente en C++. La versión final deberá integrarse en ROS (Robot Operating System).

Dificultad: Media/Alta

Competencias: Linux, ROS, C++, Optimización.

Responsable: Francisco Miguel Moreno, franmore@ing.uc3m.es .

11. -----

Título: Selección de datos con el uso de Active Learning para la creación de datasets.

Explicación: El Active Learning es una técnica del machine learning que permite la selección de datos que necesitan ser etiquetados para obtener los mejores resultados en el entrenamiento de un modelo supervisado. Se utiliza en situaciones en las que la cantidad de datos es demasiado grande para ser etiquetada completamente y es necesario dar cierta prioridad para etiquetar los datos que vayan a resultar más útiles.

Otros involucrados: -

Dificultad: Muy alta

Competencias: Autonomía, Resolución de problemas, Linux, Python, (Machine learning), (Neural networks), (Perception datasets).

Responsable: Irene Cortés Lafuente, irecorte@ing.uc3m.es

Observaciones: A partir de abril

12. -----

Título: Generación de mapa digital para vehículo autónomo, basado en nube de puntos segmentada

Explicación: Desarrollar un algoritmo en C++ que genere un mapa 2D del entorno a partir de la nube de puntos segmentada y la localización basada en GPS. La nube de puntos incluye información espacial (X,Y,Z) y de la clase a la que pertenece (señal de tráfico, línea de la carretera, etc).

Transformando la nube de acuerdo a la localización GPS y acumulándola, se irá generando el mapa 2D. El algoritmo desarrollado se integrará en un vehículo autónomo real para probarlo.

Otros involucrados: Fernando García.

Dificultad: Media

Competencias: C++ (necesario nivel alto) + ROS (recomendado, pero no imprescindible) + Linux (recomendado, pero no imprescindible).

Responsable: Miguel Ángel de Miguel, mimiguel@ing.uc3m.es

