Detección de zonas de aterrizaje seguro de drones mediante Machine Learning.

Detection of safe landing zones for drones using Machine Learning.

Daniel Robles González (1).
Estudiante Tecnológico Nacional de México, Instituto Tecnológico de Tuxtla Gutiérrez.
L19270577@tuxtla.tecnm.mx.

Néstor Antonio Morales Navarro* (2). I. T. de Tuxtla Gutiérrez, nestor.mn@tuxtla.tecnm.mx.

Cesar Morales López (3). Estudiante I. T. de Tuxtla Gutiérrez. L19270572@tuxtla.tecnm.mx.

Madaín Pérez Patricio (4). I. T. de Tuxtla Gutiérrez, madain.pp@tuxtla.tecnm.mx.

Aída Guillermina Cossío Martínez (5). I. T. de Tuxtla Gutiérrez, aida.cm@tuxtla.tecnm.mx.

*corresponding author.

Artículo recibido en noviembre 28, 2023; aceptado en diciembre 14, 2023.

Resumen.

El presente articulo describe la propuesta de un modelo de inteligencia artificial basado en Deep Learning orientado a la detección de zonas de aterrizaje. El aterrizaje de un dron debe realizarse en zonas planas y lo suficientemente grande, sin embargo, en ocasiones no siempre se da en las condiciones óptimas, por lo que a veces resulta en daños al equipo. Para lograr minimizar daños, es necesario analizar y procesar datos del entorno del dron en tiempo real, que permita identificar y seleccionar las zonas de aterrizaje más seguras y accesibles. En este proyecto, se propone un algoritmo que realiza la detección de zonas de aterrizaje por medio de YOLOv7, que nos proporciona la segmentación semántica de objetos para clasificarlos. Por lo tanto, proponemos el uso de dos clases llamadas "aterrizables" y "no-aterrizables", donde los resultados son mostrados en este trabajo.

Palabras claves: Dron, inteligencia artificial, machine learning, segmentación, YOLOv7.

Abstract.

This article describes the proposal of an artificial intelligence model based on Deep Learning aimed at detecting landing zones. Drone landings should occur in flat and sufficiently large areas; however, they don't always happen under optimal conditions, resulting at times in equipment damage. To minimize such damages, it is necessary to analyze and process real-time drone environment data to identify and select the safest and most accessible landing zones. This project proposes an algorithm that performs landing zone detection using YOLOv7, which provides semantic object segmentation for classification. Therefore, we propose the use of two classes named 'aterrizables' and 'no-aterrizables,' and the results are showcased in this work.

Keywords: Artificial intelligence, drone, machine learning, segmentation, YOLOv7.

1. Introducción.

En la última década, el uso de drones ha experimentado un notable aumento en muchas partes del mundo esto debido a que su costo disminuyó, haciéndolo más accesible para muchos usuarios (Wihbey, 2017). Por lo tanto, el uso de drones se aplica en sectores como la agricultura, logística, vigilancia y entretenimiento. Lo anterior, propicia que se coloquen aditamentos a los drones como cámaras, micrófonos, sensores o actuadores proporcionando una mayor funcionalidad (Estrada, 2018).

Sin embargo, la operación segura y eficiente de estos dispositivos aéreos autónomos plantea desafíos significativos, especialmente en lo que respecta a la identificación y selección de áreas seguras para su descenso. En ausencia de experiencia en zonas de vuelo, resulta improbable poder identificar el lugar más adecuado para llevar a cabo un aterrizaje seguro. La familiaridad con estas áreas es esencial para determinar dónde se encuentra la mejor ubicación (Shon 2018).

Aterrizar un dron de manera segura en un lugar específico puede presentar una serie de desafíos, ya que requiere una precisión considerable y la consideración de múltiples factores, que como menciona Adrián (2023) como pilotos de drones, debemos analizar el entorno y elegir el mejor sitio para despegar o aterrizar nuestra aeronave. Superficies con polvo, inclinadas, con hierba o superficies que simplemente no sean planas pondrán en peligro las hélices y motores de nuestro dron.

El Machine Learning, una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos que permiten a las máquinas aprender patrones y tomar decisiones a partir de datos, ha emergido como una herramienta poderosa en la identificación de zonas seguras para la navegación de drones. Este enfoque innovador no solo busca mejorar la precisión y fiabilidad de las rutas de vuelo, sino también minimizar los riesgos asociados con la interacción de los drones en entornos dinámicos y variables.

El minimizar riesgos a la hora de aterrizar los drones ha significado un reto para los pilotos, por lo que han surgido diversas soluciones propuestas por diferentes autores. Gil (2016) aborda este reto con el desarrollo de un sistema de aterrizaje automático sobre una plataforma de carga para un Vehículo Aéreo no Tripulado (UAV, del inglés Unmanned Aerial Vehicle). Para lograrlo, su sistema tiene varios componentes como sensores, actuadores y un microcontrolador, que ayudan a detectar la plataforma de carga y ajustar la posición y velocidad del dron para aterrizar seguro.

Delbergue (2021) aborda este desafío desde una perspectiva un tanto diferente ya que el propone el desarrollar un algoritmo de visión por computadora, destinado a supervisar el aterrizaje automático de un dron sobre una plataforma objetivo. Teniendo sobre su plataforma una letra “H” para el control basado en visión por computadora, debido que se trata de un control de visión, cualquier detección realizada por la cámara tiene un impacto directo de la aeronave.

A diferencia de los trabajos mencionados anteriormente, nuestro proyecto tiene una mayor accesibilidad, ya que no se limita solamente a un objetivo fijo, sino que debido al entrenamiento se realizó este puede reconocer una mayor variedad de áreas para el aterrizaje, como calles, campos o cualquier otra zona libre de riesgos.

2. Métodos.

Herramientas.

Yolov7.

Se decidió utilizar una herramienta que nos permitió detectar objetos en tiempo real que lleva por nombre YOLO (You Only Look Once) que utiliza técnicas de aprendizaje automático (Machine Learning), específicamente aprendizaje profundo (Deep Learning). En el contexto del Deep Learning, nos referimos a algoritmos que poseen la capacidad de emular las acciones del cerebro humano a través de redes neuronales artificiales. Estas redes, como describe DataScientest (2021), se componen de docenas o incluso cientos de "capas" de neuronas, cada una de las cuales recibe e interpreta información de la capa anterior (figura 1).

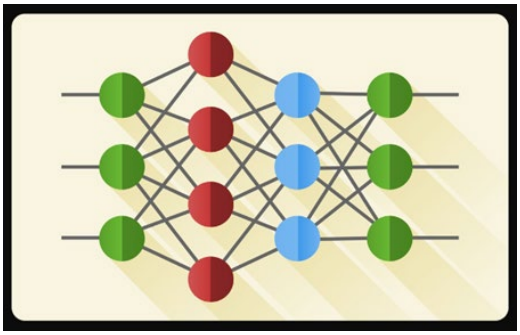


Figura 1. Redes neuronales en Deep Learning

Partiendo de que la evolución del machine learning es el Deep learning en el que se utilizan redes neuronales convolucionales. La elección de la arquitectura de una red neuronal convolucional (CNN) se refiere a decidir qué diseño específico de capas y conexiones se utilizará para construir el modelo. La elección de la arquitectura de una CNN es crucial, porque diferentes arquitecturas están diseñadas para abordar diferentes tipos de tareas y desafíos en el procesamiento de imágenes. La implementación más conocida de YOLO, como YOLOv3 y versiones posteriores, se basa en Redes Neuronales Convolucionales (CNN) para realizar la tarea de detección de objetos en imágenes y videos.

Las CNN designan una subcategoría de redes neuronales y hoy en día son uno de los modelos de clasificación de imágenes considerados como más eficaces (DataScientest, 2022). Bagnato (2018) destaca que las CNN son un tipo de Red Neuronal Artificial que se asemeja al córtex visual del ojo humano al procesar sus capas. Esto les permite identificar diversas características en las entradas, lo que a su vez les permite reconocer objetos.

El modelo YOLO es entrenado utilizando conjuntos de datos etiquetados que contienen imágenes con objetos de interés y sus correspondientes etiquetas que indican la clase y la ubicación de los objetos. Durante el entrenamiento, la red neuronal ajusta sus pesos y parámetros internos para minimizar la diferencia entre las predicciones del modelo y las etiquetas reales del conjunto de datos.

Su principal característica es que realiza la detección de objetos en una sola pasada a través de la red neuronal, en lugar de dividir la imagen en regiones y realizar múltiples predicciones como en algunos métodos anteriores.

Roboflow.

Roboflow es una aplicación web que ayuda a los usuarios a crear modelos de visión por computadora de manera más eficiente. Ofrece una variedad de funcionalidades, que incluyen la creación de proyectos, la carga de conjuntos de datos, la gestión del flujo de trabajo de etiquetado, la exportación de conjuntos de datos, así como el entrenamiento, las pruebas y la implementación de modelos.

Para este proyecto se eligió específicamente para la parte de etiquetado de las imágenes debido a que cuenta con características como:

- Utilizar modelos previamente entrenados para aplicar etiquetas automáticamente.
- Acelerar los flujos de trabajo de anotaciones manuales con etiquetado asistido por IA.
- API de anotación automática. Etiqueta automáticamente grandes lotes de datos mediante programación.
- Ver un historial de todos los cambios realizados en una imagen.Enfoque Segmentación semántica

Una red neuronal convolucional puede aplicarse a clasificación, detección o segmentación de objetos, dependiendo del uso que se le quiera dar. Para el proyecto se decidió utilizar segmentación de objetos que se refiere al proceso de dividir una imagen en una cuadrícula y asignar predicciones a cada celda de esa cuadrícula.

En YOLO, se usa la segmentación semántica para detectar objetos de manera eficiente y simultánea en toda la imagen, a diferencia de otros enfoques de segmentación que se centran en subdividir una imagen en regiones. YOLO asigna una etiqueta semántica a cada píxel de la imagen, clasificando así cada región según su contenido (figura 2). Por ejemplo, identifica píxeles que corresponden a clases como persona, automóvil, árbol, etc.

Figura 2. Segmentación semántica.

3. Desarrollo.

Análisis de las posibles muestras.

Se eligieron imágenes con terreno irregular, inclinado y desfavorable para el entrenamiento y pruebas de descenso de los drones. Esta elección fue estratégica, buscando identificar lugares que ofrecieran condiciones óptimas para el avance del proyecto. Se utilizó un conjunto de 200 imágenes a color y de profundidad que están disponibles en el trabajo de Schleiss et al. (2022). Estas imágenes abarcan tanto entornos urbanos como rurales; los entornos rurales exhiben campos de cultivo, árboles, ríos, lagos, y más, mientras que los entornos urbanos muestran edificios, vehículos, calles, autopistas y otros elementos característicos. Se considero este dataset porque tiene diversidad de entornos y eso nos da una visión de cómo funciona nuestra propuesta en distintos escenarios (figura 3).

Figura 3. Conjunto de muestras.

Elección del enfoque de clasificación de imágenes.

La segmentación semántica se eligió sobre la detección de objetos debido a su precisión al separar objetos en una imagen. A diferencia de los Bounding Boxes, que pueden incluir áreas no relacionadas, la segmentación permite identificar y separar objetos de manera clara como se observa e la figura 4. Esta técnica es crucial en aplicaciones que requieren precisión visual, detectando bordes y contornos de objetos. En la determinación de zonas aterrizables para drones, ayuda a identificar áreas libres de obstáculos como árboles o edificaciones, mejorando la seguridad y eficiencia operativa.

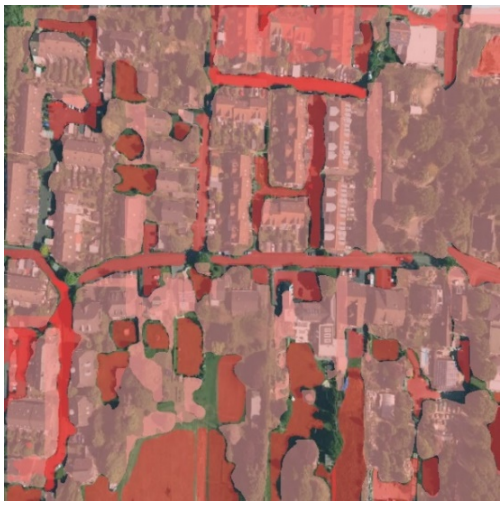


Figura 4. Ejemplo de segmentación.

Etiquetado de las muestras.

Este punto se enfoca en el proceso de etiquetado de imágenes con el propósito de entrenar la red neuronal, lo cual le permitirá reconocer patrones y características específicas. Para este trabajo se utilizó la herramienta llamada Roboflow (figura 5), debido a que facilita la administración de datos, permitiendo crear conjuntos de imágenes etiquetadas, donde el usuario puede cargar las imágenes, y etiquetarlas manualmente o utilizar herramientas automáticas para hacerlo más rápido y generar conjuntos de datos exactos.

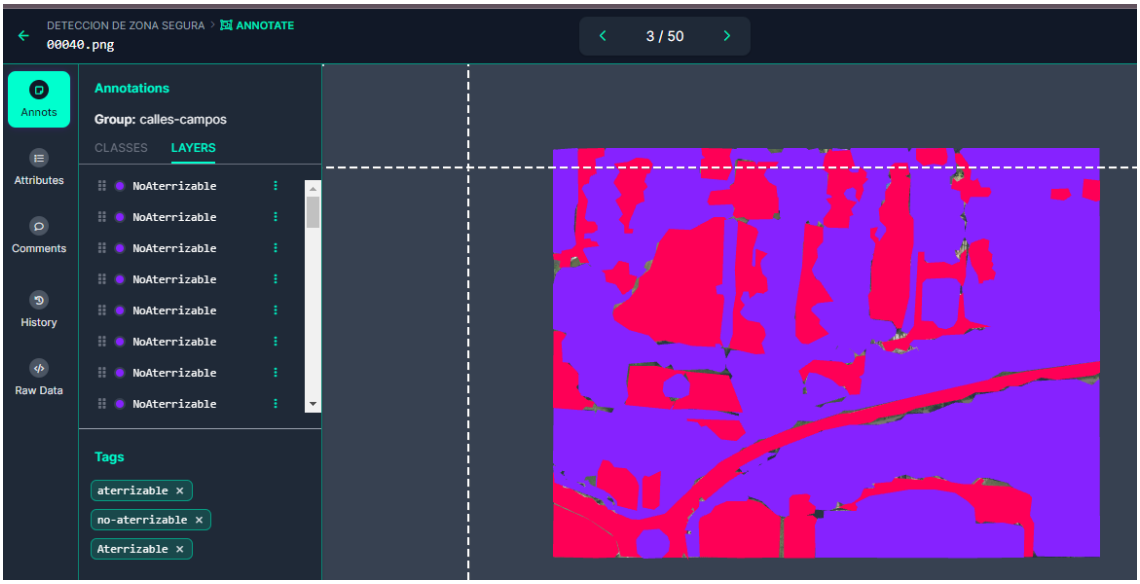


Figura 5. Herramienta Roboflow.

En el etiquetado, las imágenes deben ser analizadas a profundidad, con tiempo y detenimiento, debido a que si se quieren obtener buenos resultados para el entrenamiento se deben etiquetar con exactitud cada una de las zonas. Para el proyecto se utilizaron dos clases que fueron las zonas aterrizables (Rojo) y las zonas no aterrizables (Morado), en donde el criterio para poder establecer una etiqueta a cierto objeto dependerá de la persona que realice el etiquetado.

Elección de la arquitectura de la red neuronal.

El algoritmo que se eligió por la estructura de su CNN fue YOLOv7 que realiza la detección de objetos a través de la red neuronal, en lugar de hacer múltiples procesamientos como otros enfoques. Esto hace que Yolo sea más rápido y eficiente en términos computacionales, siendo capaz de realizar detección de objetos en tiempo real. A pesar de que YOLO se centra en detectar objetos en lugar de segmentar imágenes, otra razón para elegirlo radica en su reputación por ser simple y fácil de implementar. En comparación con enfoques más complejos para la segmentación, la simplicidad de YOLO hace que sea más accesible para aquellos que buscan realizar tareas de segmentación sin enfrentar una curva de aprendizaje lenta.

Entrenamiento de la red neuronal del algoritmo elegido.

En este apartado se llevó a cabo el entrenamiento de la red neuronal, haciendo uso del dataset etiquetado previamente, utilizando los parámetros específicos de la versión del algoritmo. Para el entrenamiento se hizo uso de la herramienta Google Colab, debido a que este permite programar y ejecutar sin necesidad de configurar un entorno de desarrollo en el equipo local.

Posteriormente, se realizó la instalación de los requerimientos para poder ejecutar los códigos del entrenamiento, después se hizo la descarga del dataset que se obtuvo de Roboflow, donde se incluían las 200 imágenes en tres

carpetas: train, test y valid, y también un archivo .yaml el cual contienen el número de clases, los nombres de estas y la ruta de cada una de las carpetas.

Teniendo todo lo anterior, se procedió al entrenamiento en el que se colocaron parámetros como la ruta del archivo que contiene los pesos por defecto del YOLOv7, el tamaño del batch en el que se colocaron 4, el número de épocas que para mejores resultados se puso en 200 y la ruta del archivo .yaml. como se muestra en la figura 6.

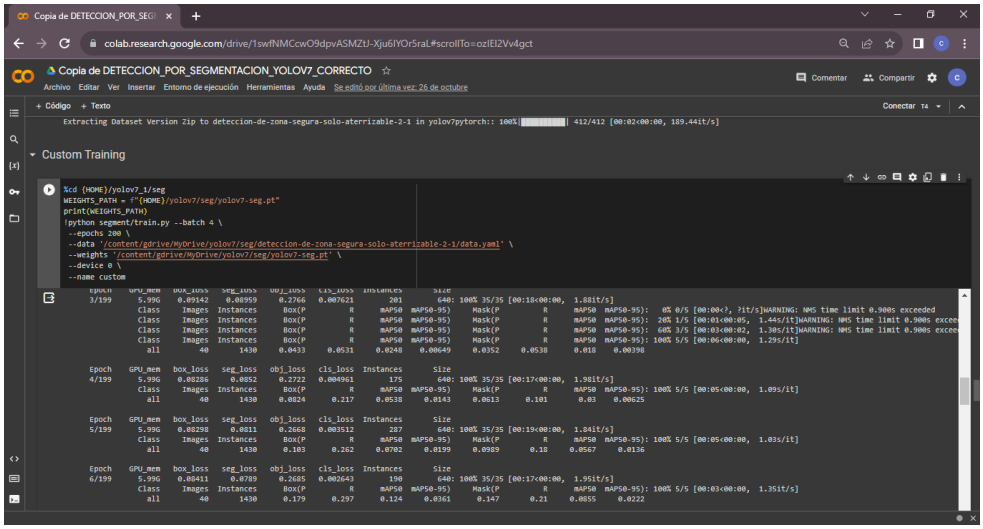


Figura 6. Proceso de entrenamiento.

Una vez finalizado el entrenamiento se creó un archivo que contiene los pesos nuevos que será el que contendrá el modelo de detección de aterrizaje de zonas seguras que necesitaremos para hacer las pruebas de que tan eficaz fue todo el proceso que se realizó.

Resultados.

El entrenamiento se llevó a cabo en dos instancias: la primera se establece la detección de dos clases, tanto las zonas aterrizables como las no aterrizables. La segunda se estableció una clase definida como zonas aterrizables. Para evaluar el desempeño del algoritmo de entrenamiento, se empleó una matriz de confusión, que proporcionó una visión detallada al comparar las predicciones del modelo con las clases reales en los datos de prueba.

En el primer entrenamiento, los resultados se observan en la figura 7 en el que la clase "Aterrizable" tuvo una precisión del 59 por ciento, se presenta un 3 por ciento en el que fue erróneamente identificado como aterrizable en objetos que no son aterrizables, y un 38 por ciento fue detectado en zonas que no pertenecen a ninguna clase. En cuanto a la clase "No aterrizable", fueron detectados correctamente en un 55 por ciento, un 4 por ciento fue incorrectamente detectado como no aterrizable en objetos que eran aterrizables, y un 41 por ciento fue detectado en áreas que no pertenecían a ninguna clase. Respecto a los objetos que debían ser detectados con una clase específica, pero fueron identificados como si no pertenecieran a ninguna, el 46 por ciento eran no aterrizables y el 54 por ciento eran aterrizables.

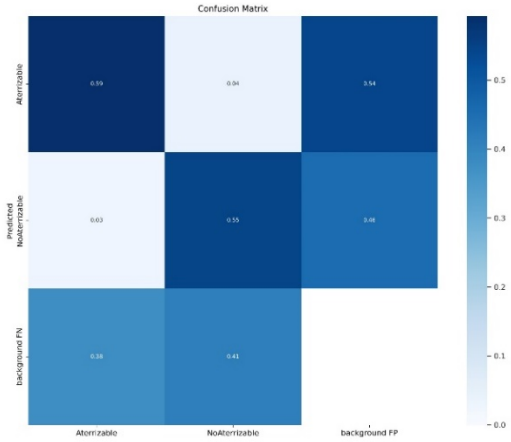


Figura 7. Matriz de confusión zonas aterrizables y no aterrizables.

Para el segundo entrenamiento, en el cual solo se detectaron las zonas aterrizables un 65 por ciento fue correctamente identificado como aterrizable. El 35 por ciento restante se detectó en regiones de la imagen que no pertenecían a ninguna clase. Además, el 100 por ciento de los objetos que no fueron detectados con una clase específica que debían ser etiquetados fueron clasificados como aterrizables, dado que en este modelo solo se consideraba una clase, como se ilustra en la figura 8.

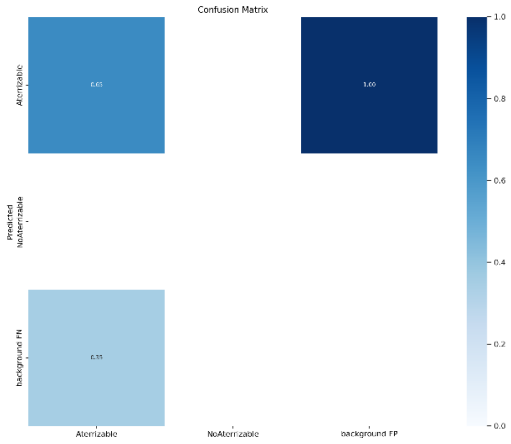


Figura 8. Matriz de confusión zonas aterrizables.

En la figura 9 se muestran los resultados de detección del primer modelo, donde la identificación de áreas es interpretada como pertenecientes a ambas categorías. Estas áreas son marcadas en color rojo para las zonas consideradas aterrizables y en color rosado para las no aterrizables. Además, se observa que en algunas calles que deben ser consideradas como aterrizables, no se detectan. Asimismo, se aprecia que áreas que deberían ser consideradas como no aterrizables, principalmente árboles y vehículos, no son detectadas.

Figura 9. Resultados del primer modelo.

Respecto al segundo modelo, se decidió entrenarlo con una sola clase (figura 10) con el propósito de mejorar la precisión en la identificación de las zonas aptas para el aterrizaje. El resultado obtenido reveló la detección de áreas consideradas aterrizables, pero estas incluían objetos de tamaño reducido que el modelo interpretó erróneamente como un área aterrizable, como fue el caso de algunos árboles y vehículos. Por consiguiente, esta interpretación no es precisa. Asimismo, se identificaron zonas con detecciones duplicadas, señaladas por un tono rojo más intenso.

Figura 10. Resultados del segundo moldeo.

Conclusiones.

En este artículo, hemos explorado el uso del aprendizaje automático en la identificación de zonas seguras para el aterrizaje de drones. Específicamente, nos enfocamos en YOLOv7, un avanzado algoritmo de detección de objetos que ha demostrado su eficacia en esta tarea, aunque su desempeño en segmentación no sea igual de destacado.

A pesar de su principal enfoque en la detección de objetos, optamos por emplear YOLOv7 para segmentar zonas aterrizables y no aterrizables. Realizamos dos entrenamientos, y el segundo modelo, que identificaba exclusivamente zonas aterrizables, arrojó los mejores resultados teniendo como resultado una precisión del 65 por ciento.

El modelo obtenido podría ser utilizado por drones automáticos en situaciones de emergencia, como pérdida de batería, señal o daños en el dron, facilitando la detección automática de áreas de aterrizaje y asegurando un descenso seguro.

No obstante, reconocemos los desafíos pendientes, como la necesidad de recopilar y procesar grandes volúmenes de datos, así como mejorar aún más la precisión del algoritmo. El conjunto de datos utilizado para entrenar nuestro algoritmo fue limitado en tamaño. Creemos que la precisión del algoritmo podría aumentar significativamente si se empleara un conjunto de datos más robusto, aunque es esencial considerar el riesgo de sobreajuste durante el proceso de aprendizaje.

A pesar de estos desafíos, estamos convencidos del enorme potencial del aprendizaje automático para mejorar la seguridad en este contexto.

Referencias bibliográficas.

Adrian (2023). 12 errores de los pilotos de drones y las soluciones. https://www.droneguru.es/errores-pilotos-de-drones/

DataScientest (2021, 16 de diciembre). Convolutional Neural Network: definición y funcionamiento. https://datascientest.com/es/convolutional-neural-network-es

DataScientest (2022, 19 de abril). Deep Learning o Aprendizaje profundo: ¿qué es? https://datascientest.com/es/deep-learning-definicion

Estrada (2018). Uso de drones: retos y oportunidades. https://foroconsultivo.org.mx/INCyTU/documentos/Completa/INCYTU_18-016.pdf

Ignacio Bagnato. (2018, noviembre 29). ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador. Aprende Machine Learning. https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/

John Wihbey (2017). Los geodatos generados mediante UAV mejoran las políticas de suelo desde América Latina hasta China. Lincoln Institute. https://www.lincolninst.edu/es/publications/articles/la-revolucion-los-drones

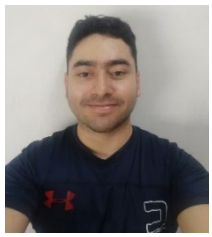
Luis Alfonso Delbergue Chico (2021). Aterrizaje de un dron de forma autónoma con perturbaciones mediante algoritmo de visión. Proyecto Fin de Carrera / Trabajo Fin de Grado, E.T.S.I. Industriales (UPM).

Marcos Enrique Gil Fernández (2016). Desarrollo de un sistema de aterrizaje autónomo sobre una plataforma de carga para un UAV tipo Parrot AR.Drone 2. [Proyecto Fin de Carrera / Trabajo Fin de Grado, Universidad Politecnica de Madrid]. Archivo Digital UPM. https://oa.upm.es/43909/1/TFG_MARCOS_ENRIQUE_GIL_FERNANDEZ.pdf

Schleiss Michael, Rouatbi Fahmi & Cremer Daniel (2022). VPAIR-Aerial Visual Place Recognition and Localization in Large-scale Outdoor Environments. arXiv preprint arXiv:2205.11567.

Shon (2022). ¿Necesito una plataforma de aterrizaje para mi dron?. https://drones-camera.com/es/do-i-need-a-landing-pad-for-my-drone/#google_vignette

Información de los autores.



Daniel Robles González, estudiante del Instituto Tecnológico de Tuxtla Gutiérrez, pasante de la carrera Ingeniería en Sistemas Computacionales.

Néstor Antonio Morales Navarro es Doctor en Desarrollo Tecnológico. Docente del departamento de la carrera de Ingeniería en Sistemas Computacionales y Jefe del departamento del Centro de Cómputo del Tecnológico Nacional de México campus Tuxtla Gutiérrez. Miembro del Sistema Nacional de Investigadores, nivel Candidato. Se especializa en el área de Visión e Inteligencia Artificial.

César Morales López, estudiante del Instituto Tecnológico de Tuxtla Gutiérrez, pasante de la carrera Ingeniería en Sistemas Computacionales.

Madaín Pérez Patricio es graduado como Doctor en Ciencias en Automatización y Computación Industrial en 2005, por la Universidad de Ciencias y Tecnologías de Lille, Francia. Investigador y Docente desde 1997 en el Departamento de Posgrado y la Carrera en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez. Es miembro del Claustro Doctoral "Doctorado en Ciencias de la Ingeniería" perteneciente al Programa Nacional de Posgrados de Calidad, I.T. de Tuxtla Gutiérrez desde 2016. Miembro del Sistema Nacional de Investigadores, nivel II.

Aída Guillermina Cossío Martínez es Maestra en Ciencias en Administración por el Instituto Tecnológico de Tuxtla Gutiérrez en 2002. Es profesora de tiempo completo del área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez, desde 1994. Se especializa en la formulación y evaluación de proyectos, así como el emprendimiento y desarrollo de planes de negocio, actualmente es perfil deseable y trabaja en la línea de investigación Tecnología de Información y Base de Datos.