Sudan University of Science and Technology

College of Engineering

School of Electronics Engineering

Indoor Drone Navigation with Machine Learning and SLAM

A Research Submitted In Partial fulfillment for the Requirements of the Degree of B.Sc. (Honors) in Electronics Engineering

Prepared by:
Abdelwahid Burham Ali Burham
Haider Abdelhameed Jadallah Abdelhameed
Abobaker Mohammed Alagib Aljalal
Almustafa Abdelhaleem Ahmed Awad Allah

Proposed Supervisor:
Prof .Dr. Rashid Al-SAEED

الإستهلال

قال تعالى:

(ن والقلم وما يسطرون)

سورة القلم

صدق الله العظيم

DECLARATION

The work described in the research titled "Indoor Drone Navigation with SLAM and Machine Learning" presented in this paper is my own work under supervision of , that was done with the help of my research partners on the application of SLAM with Machine D.Rashid] This work focuses] Learning techniques to enhance navigation of indoor drones, and to achieve autonomous drone operation, identify and avoid obstacles as well as plan .routs efficiently

I am fully aware that all knowledge acquired was properly cited and this work does not represent any work previously presented for any educational or professional certification.

ACKNOWLEDGMENT

Thanks and appreciation. We would like to express our sincere gratitude to everyone who contributed to the successful completion of this project. Special thanks go to our supervisor [D.Rashid], for his valuable guidance, support and constructive feedback throughout the process. We would also like to express our sincere appreciation to our colleagues and team members, whose cooperation, dedication and teamwork made this project possible. In addition, we are grateful to [Sudan university of science and technology] for providing the resources and facilities that enabled us to conduct our research. Finally, we would like to thank our families and friends for their constant support and encouragement throughout the journey.

DEDICATION

We dedicate this research work to our family, whose unwavering support and encouragement have been my constant source of inspiration. To my friends and colleagues, who have been a part of this journey and provided invaluable insights along the way.

ABSTRACT

This work aims at enhancing an indoor drone navigation with the use of Simultaneous Localization and Mapping (SLAM) solutions in combination with convolutional neural networks (CNNs). In particular, the Oriented FAST and Rotated BRIEF (ORB-SLAM) algorithm is employed to undertake visual-based localization and mapping by feature detection and tracking from the environment. However, to improve the accuracy and combat issues related to feature-sparse or ambiguous scenes, a CNN is used for semantic scene analysis, including obstacle and object recognition. Integrating ORB-SLAM for the feature mapping and CNN for the deep sense of vision, it is possible to implement the drone's autonomous flight in complex indoor conditions, with real-time obstacles' detection and permanent localization. The combination of machine learning with SLAM improves the dependability and practicality of the general system for different and complex indoor environments.

الخلاصة

يهدف هذا العمل إلى تحسين الملاحة الداخلية للطائرات بدون طيار باستخدام حلول التوطين والرسم المتزامن (SLAM) بالاشتراك مع الشبكات العصبية التلافيفية (CNNs). على وجه الخصوص، يتم استخدام خوارزمية Oriented FAST و ORB (SLAM-Rotated BRIEF) للقيام بالتوطين والرسم الخرائطي البصري من خلال اكتشاف السمات وتتبعها من البيئة. ومع ذلك، لتحسين الدقة ومكافحة المشكلات المتعلقة بالمشاهد الغامضة أو قليلة السمات، يتم استخدام CNN لتحليل المشهد الدلالي، بما في ذلك التعرف على العوائق والأشياء. من خلال دمج SLAM-ORB لرسم خرائط السمات و CNN للحس العميق للرؤية، من الممكن تنفيذ الطيران المستقل للطائرة بدون طيار في ظروف داخلية معقدة، مع اكتشاف العوائق في الوقت الفعلي وتحديد الموقع الدائم. يعمل الجمع بين التعلم الآلي و SLAM على تحسين موثوقية وعمليّة النظام العام للبيئات الداخلية المختلفة والمعقدة

Contents

List of Figure

List of Table

Chapter One

INTRODUCTION

1. Introduction

1.1 Overview:

Over the last decade, the role of Unmanned Aerial Vehicles (UAVs) in research and development has been on the rise. Intense navigation of UAVs is the capacity of a UAV drone to manoeuvre within a structure, a building, a warehouse, an industrial compound or any other structure that can be defined indoors without the aid of Global Positioning System (GPS) signals. This process needs state of art technology to minimize errors and contain risks associated with working in environments that often have barriers.

SLAM is generally a technique that goes into an outline for an SN system solution. It is called this way because it assists the system in a localization process using map generated with simultaneous actions, which makes it Synchronous L&AM. Thus, any navigation system's work implies the sending up the path and the avoidance of collision with the available barriers on the planned route. The operation scenarios are to be learnt and figured out with the assistance of the SLAM algorithm incorporated in the navigation system.

Machine Learning tries to use a data set to encapsulate solutions for a particular problem statement and tries to train a computing system using the offered data set in a manner that the system is trained to provide correct solutions for situations even if they are not included in the data set and helps in mapping and locating services which leads to proper navigation. In other words, the knowledge-acquiring step of the system occurs during training, and during working with the system, such problems are solved in real-life manor with suitable answers like a man does it.

1.2 Problem Statement:

The problem of dealing with obstacles in navigating drones inside environments is one of the biggest technical challenges. This problem requires careful attention to several elements and to ensure safety and effectiveness. Let's look at each part of the problem from all sides:

Discovering obstacles Complex indoor environment: In critical environments such as buildings or warehouses, there are many potential obstacles such as agents, furniture, wires, and pipes. These obstacles may be large, small, or mobile. In lighting: Control of lighting and low-light conditions can start on event sensors, such as camera [1]

Rapid Response Once obstacles are detected, the aircraft must make quick decisions to avoid a collision. This requires effective path planning strategies and improved steering ability.

Interacting with moving obstacles: In environments that contain moving obstacles, such as people or equipment, predicting the movement of these obstacles becomes more complex [2]

1.3 Aim and Objective:

The main aim It is to enable drone to move efficiently and independently within closed and complex environments , The objective of this research are:

-To Improving indoor mobility.

-To Increased positioning and navigation accuracy.

-To Effective obstacle avoidance.

-To Adapting to changes in the environment.

1.4 Scope:

This research includes the use of machine learning algorithms for indoor drone navigation to process and use the integration of SLAM techniques with machine learning to improve the ability of the drone to determine its location, such as CNNs technology to detect obstacles, and the use of RNNs to predict future movement, and the Python language contributes to integrating SLAM algorithms with models. Machine learning thanks to its wide support for data processing and analysis libraries.

1.5 Methodology:

By integrating obstacle handling problem being handled through machine learning means that you can get enhanced solutions when dealing with the obstacle handling by enhancing the detection as well as the avoidance of obstacles and by even adapting to the surroundings. Here's how machine techniques such as Convolutional Neural Networks (CNNs): Utilized to interpret image learning can help solve this problem:

Improve obstacle detection by Deep learning feeds from a camera or some other device in order to determine all obstacles properly. CNNs can learn from images; thus, it can be used to recognize features that act as barriers such as a wall or another piece of furniture.

Image classification models: Deep FRAMEWORKS like YOLO (You Only Look Once) or SSD (Single Shot Multi Box Detector) must be able to be trained to classify the categories of the obstacles whose detection we want to enhance.

Unsupervised learning techniques such as Clustering: The data from sensors could be grouped into clusters using some form of clustering algorithms as K-means in which each cluster will represent a different type of obstacle. This contributes to the ability of identifying unfamiliar pattern.

Avoid obstacles by Predictive planning such as RNNs and LSTMs: They can also be used to predict the motion of dynamic obstacles, including pedestrians and vehicles, to plan more seemingly unique avoidance trajectories.

Reinforcement learning by Adaptive algorithms: like Q-learning and Deep Q-Networks (DQN) can be employed to allow the aircraft to learn what actions are suitable when in a vicinity of obstacles. The system is able to modify its performance based on the result it receives and as such, it becomes dynamic in nature.

1.6 Thesis outline:

This research contains five chapters and is divided as follows:

Chapter 1: Introductory chapter that discusses the importance of the research and describes the problem ,proposed solution, aims and the method used in this study

Chapter 2: Presents the background information related to the proposed system including indoor drone navigation using slam and machine learning. In addition, previous works and rehabilitation solution are introduced

Chapter 3: This chapter covers a detailed overview of the architecture, design and simulation of the proposed system. Moreover, the collection of the software components\modules that are used in explained.

Chapter 4:It presents and describes the performance of the proposed system. The result achieved by this work are compared with the ones' achieved by traditional rehabilitation facilities. In addition, a discussion of these findings and results is provided .

Chapter 5: Concludes the thesis and provides suggestions for future work.

Chapter Two

Theoretical Background & Literature Review

2. literature Review

2.1 Background:

Indoor navigation is the modern application that belongs to the vital areas that help to provide better experience to the user and make the processes in indoor environments more efficient. Indoor navigation considers the location and direction of a space within areas that do not have GPS features including buildings, malls, or industrial premises. Similarly, while outdoor navigation has constraints in terms of obtaining location information, indoor navigation has enormous barriers in the form of both physical barriers such as walls, furniture, etc., and interference of the surrounding environment.

These systems can only be managed using high end technology to counter the effects brought by the internal environment. One of the most promising is Wi-Fi beacons that are utilized for the identification of locations by means of signal strength and distribution in a certain building. The ultrasonic sensing and infrared based devices offer improved precision for measurement of distances. Moreover, computer vision technologies, which include image and video recognition by means of convolutional neural networks (CNNs), can simplify and improve the accuracy of environment analysis and proper localization.

Digital maps as well as interior plans facilitate ways by offering graphical displays on the layouts of interior features. However indoor navigation needs refinements to manage change in environment and issues related to managing navigation inside enclosed areas.

To sum up, indoor navigation can be considered as quite an active and constantly developing sphere at present. Present-day approaches aim at improving the accuracy and the integration of the solutions offered in order to increase the efficiency of indoor space control and indoors environments' navigation.

Implemented traditional systems for navigation in closed space have some problems. Extinguishing GPS signals, the ability to describe certain spaces task in the context of intricate building designs, and the interference created by walls and furniture all counteract the functionality of these systems.

Figure 2.1 Drone that use in Indoor navigation

Machine learning:

Recent years have witnessed advancements in fully autonomous drone navigation especially in more challenging scenario where would be required to navigate indoors. Indeed one of the key technologies that has led to such advancements is machine learning. Specifically, machine learning is one of the branches in the artificial intelligence that is all about creating models that are capable of learning from the data and becoming better and better over time without the need to be reprogrammed.

The technique that is opening the door to improving drone position is machine learning to compensated drone for its failure within a building or industrial facility. Outdoor environments contain satellite signal such as GPS not present in indoor environment hence the need for other positioning and orientation systems. This is where Machine Learning comes into the scene which can provide an optimum solution by doing some ironing of the Environmental data in a way not imaginable by other conventional techniques.

Therefore, machine learning is a decisive element that promotes enhancement of the performance of systems' accuracy. Thus, with the help of analyzed environmental data, the development of more sophisticated guidance algorithms, and the flexibility of machine learning, further increases in accuracy and efficiency of navigation within complex indoor environments are reached. These capabilities account for some of the critical features that, when incorporated into use of the drones, make them a useful tool in many practical and research utilizations.

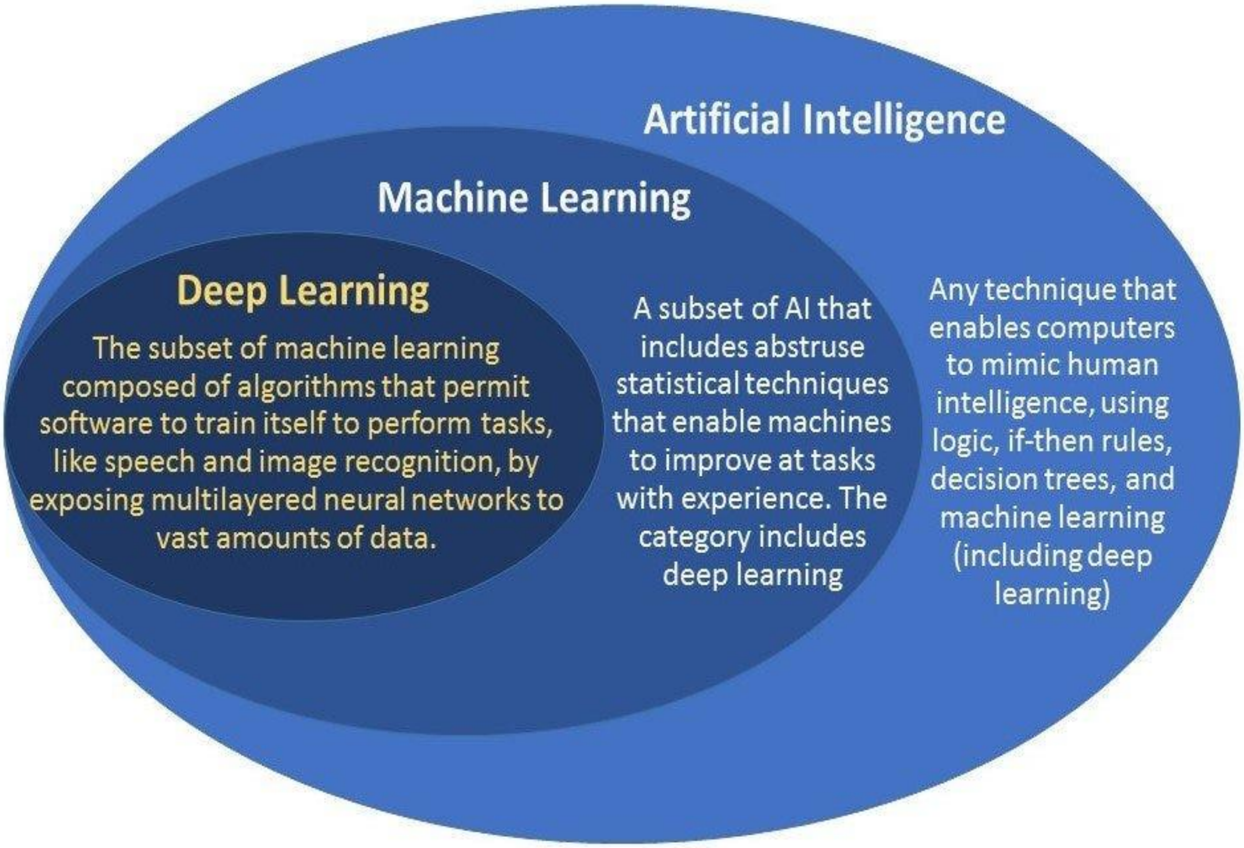

Figure 2.2 what is machine learning and deep learning

Deep learning can be applied to as one of the specialized methods in the class of machine learning.

It refers to a set of models and techniques working with deep neural network to accomplish some goals:

Convolutional neural networks (CNNs):

are a certain sort of artificial neural network that are intended for working with network-data that has network structure, such as photographs and videos. Because of the convolutional, raster as well as the nonlinear layers of the CNN, it offers high performance in pattern recognition. Computer vision made extensive use of it which is an area of artificial intelligence.

Recurrent neural networks (RNNs) :

are effective in serial and temporal data due to efficient context save, voice recognition, and video analysis. Other issues like explosion and vanishing gradient have been seen in the basic RNNs but better advanced RNNs like LSTMs and GRUs have been worked out.

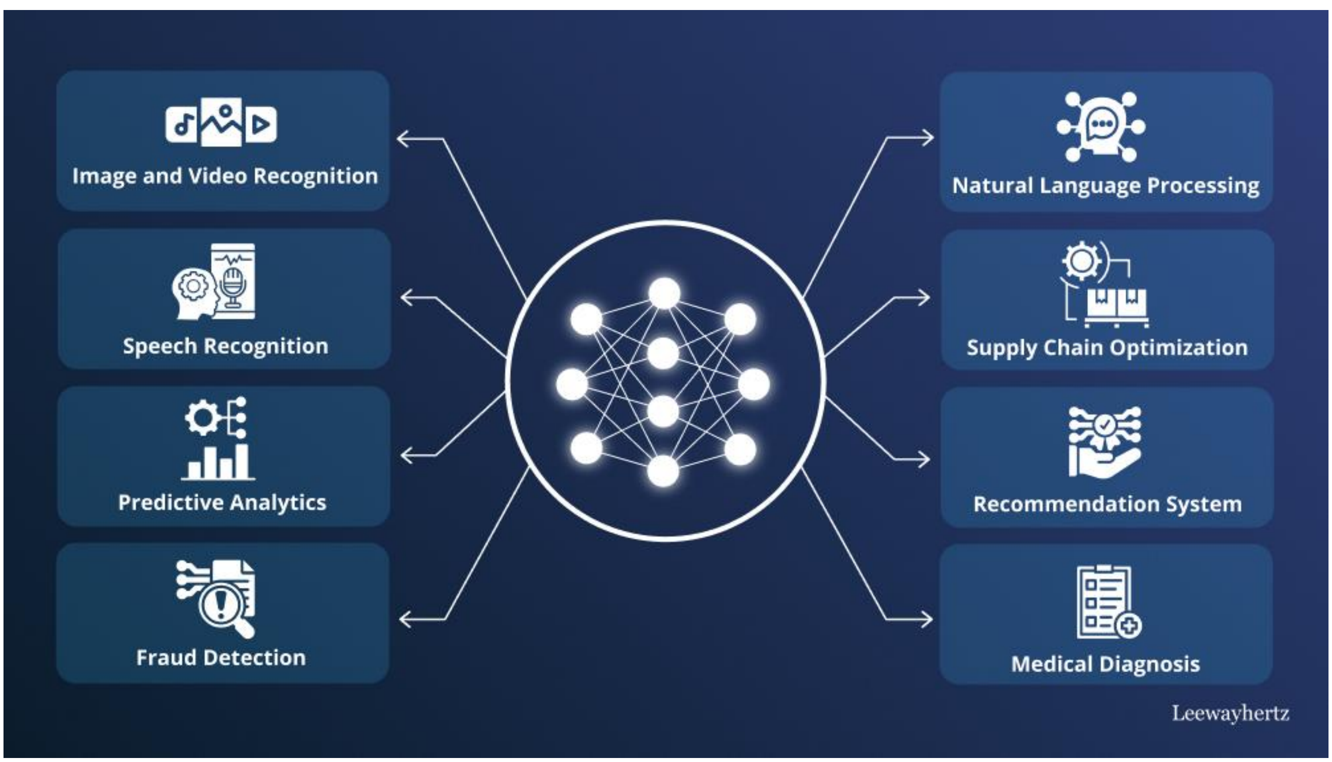

Figure 2.3 Deep learning application

Simultaneous Localization and Mapping (SLAM):

is one of the most central technologies in the concept of indoor navigation. It helps enhancing the performance of the systems to identify their position and make maps of the environment in real time. It can build the indoor environment map in three dimensions by the help of data that collected by the cameras, ultrasonic sensors and lasors. These maps assists in the identification of areas of the environment that contain obstacles as well as those that contain items. Because of it, systems can unerringly identify the position they occupy in the environment they build. This comes in handy most especially when GPS signals cannot be obtained as is the case indoors.

It can optimize the systems overall performance because navigating performance can be greatly improved, it aids to give the right information about the environment and reduces system errors to provide an improved user experience.

It could be integrated into other technologies such as computer vision and machine learning to enhance application and positioning precision. For instance, the feature of computer vision can be employed in determining aspects of the environment that enhances the mapping mode.

2.2 Related work :

They built an advance indoor navigation system for drones based on the Nav2 navigation system framework and SLAM Toolbox in ROS environment in this research study within the Gazebo simulation. Nonetheless, their application offers significant advancements over earlier applications in the context of ground vehicles and terrestrial robotics, such as UAVs (Unmanned Aerial Vehicles). The suggested technique addresses the most pressing issue with precise mapping and localization in indoor settings when GPS signals are not available. By demonstrating that a virtual quadcopter can, in fact, avoid obstacles and travel freely inside an indoor space on its own, this work adds to the body of current information. In particular, Their method promises its applicability elsewhere by going beyond and extending the capabilities of the Nave2 architecture, which was previously created and demonstrated for ground vehicles. This invention is a significant advancement over existing methods and offers a new use for indoor UAV navigation, which is crucial for tasks like surveillance in GPS-deficient locations, inspection, and search and rescue in disaster-affected areas [3]

In many drone applications, drones need the ability to fly fully or partially autonomously to carry out their mission , Jung-Min Jerry Park in [4] To enable such fully partially autonomous flights, the ground control station that is supporting the drone’s operation needs to constantly localize and track the drone, and send this information to the drone’s navigation controller to enable autonomous semiautonomous navigation. In outdoor environments, localization and tracking can be readily carried out using GPS and the drone’s Inertial Measurement Units (IMUs). However, in indoor areas or GPS denied environments such an approach is not feasible , he has done propose a localization and tracking scheme for drones called ROLATIN (Robust Localization and Tracking for Indoor Navigation of drones) that was specifically devised for GPS denied environments. Instead of GPS signals, ROLATIN relies on speaker generated ultrasonic acoustic signals to estimate the target drone's location and track its movement. Compared to vision and RF signal -based methods.

In [5] An indoor navigation system environments based on LIDAR were used to solve the problem of indoor structures blocking satellite signals. Was used LIDAR as a sensor to sense the environment. Synchronous positioning and mapping algorithms were used to provide effective real-time positioning information. The drone was separated from the dependence on satellite signals, and realized the autonomous navigation and map construction of the drone in special such as indoors.

In [6] examines the current and upcoming developments in indoor positioning, localization, and navigation (PLAN) technology. It discussed indoor PLAN requirements, key participants, sensors, and methodologies. In addition to navigation sensors like the Inertial Navigation System (INS) and the Global Navigation Satellite System (GNSS), environmental perception sensors like cameras, Light Detection and Ranging (LiDAR), High-Definition maps (HD maps), the fifth generation of mobile network communication technology (5G), and Internet-of-things (IoT) signals are becoming crucial PLAN auxiliary sensors. With the advent of increasingly sophisticated sensors, multi-platform/multi-device/multi-sensor information fusion, self-learning systems, and the integration of artificial intelligence, 5G, IoT, and edge/fog computing, PLAN systems should become more intelligent and resilient.

In [7] , A state-of-the-art deep learning-based drone identification model has been trained using a dataset of 6,000 artificial drone maps. Our depth map-based method allows full 3D localization of the obstacle, whereas current sensing methods can only provide relative height and azimuth. Since 3D localization of detected drones is essential for effective collision-free path planning, this is very helpful for collision avoidance. With several drone types flying at up to 2 m/s, the suggested detection method has been verified in many actual depth map sequences, attaining an average precision of 98.7%, an average recall of 74.7%, and a record detection range of 9.5 meters.

in [8] For unmanned aerial vehicles, a straightforward, cost-effective, and effective autonomous mapping and navigation system was demonstrated. Three modules have been put into place to make this system a reality. The first module, which is based on ORB-SLAM, one of the best monocular SLAM algorithms, builds a 3D model of the surroundings while the drone navigates itself. It was suggested to use a visual-based line tracking technique for the system's autonomous navigation. After that, the second module converts the 3D map to a 2D grid map in real time. They introduced a threshold-based technique that converts 3D maps to 2D directly, without the need for any intermediary components, unlike the majority of 3D to 2D map conversion research use octomaps in the middle of two. Lastly, The third module navigates the drone to the desired pose on the created 2D grid map using the A* path planning algorithm. To accomplish this, this module solely employs monocular camera data in conjunction with IMU-assisted

Adaptive Monte Carlo localization (AMCL). The findings of the experiment show that the suggested approach is sufficiently effective to be utilized in inexpensive drones with a single monocular camera and constrained computing power.

In [9] Within The scientific community has been concentrating on vision-based navigation in recent years because of the most recent advancements in embedded hardware solutions, which enable us to have both a high computational capacity and a low weight in order to meet the payload limitations of aerial platforms. examined SLAM State of Art Methods for UAV Navigation in Hazardous Conditions. In order to determine which visual-inertial algorithm offers the highest accuracy for navigation in areas where GPS is not available, it compares various techniques. VINS-Fusion, an optimization-based VIO algorithm, SVO, a semi-direct monocular VIO algorithm, the highly optimized proprietary VI-SLAM algorithm of the Intel T265 Tracking Camera, and the VI-SLAM Systems, ORB SLAM 2 and its improved version, ORB SLAM 3, are the analyst algorithms. Using an Intel T265 Tracking Camera, the aforementioned algorithms were evaluated along a path in a compact indoor space with lots of features and artificial illumination. The analysis of the localization error in relation to the ground-truth reference path was used to assess the algorithms' performance. The findings indicated that VINS-Fusion is the most accurate option. Nevertheless, a significant amount of computational resources are needed to achieve its outstanding performance. As a result, before utilizing this technique on board for state estimate, a preliminary review of how much processing power will be left over for navigation, control, and other crucial functions.

In robotics, Simultaneous Localization and Mapping (SLAM) is a fundamental concept. It includes pose-graph optimization and sensor signal processing. Applications for SLAM have been found in a number of fields, including as military operations, agriculture, courier services, and environmental monitoring, especially when Unmanned Aerial Vehicles (UAVs) are used. There are numerous applications. In [10] intends to offer a thorough examination of the CNN-SLAM, Linearized Kalman Filter (LKF), and Extended Kalman Filter (EKF) Simultaneous Localization and Mapping (SLAM) algorithms. Through an analysis of its application in emergency scenarios, precision agriculture, and geological surveys, the usage of SLAM algorithms in Unmanned Aerial Vehicles (UAVs) was investigated. He discussed some problems that SLAM algorithms might have with wide-area applications, real-time processing and efficiency, robustness, and dynamic objects in the environment. It also compared the advantages and disadvantages of the three algorithms and offered possible solutions for the problems mentioned.

Industrial cyber-physical systems could benefit greatly from the current advancements in unmanned aerial vehicle technology. Dedicated precise wireless localization is a possible contender to enable indoor navigation, whereas global navigation satellite systems are used to achieve autonomous flight in outer conditions. Wireless localization can be installed in vast industrial regions, but it could not be cost-effective to provide the necessary coverage. In order to enable autonomous flight in regions not served by wireless localization, our work combined and enhanced state-of-the-art ultra-wideband localization with monocular simultaneous localization and mapping. Two experiments were conducted in order to thoroughly validate the suggested approach. The first experiment demonstrated the accuracy of various drone localization techniques, while the second experiment demonstrated that precise waypoint flight in regions not covered by wireless localization is possible, lowering the bar for integration of drone-based cyber-physical systems in the current industrial environment in [11] .

A overview of data fusion and simultaneous localization and mapping (SLAM) methods for object detection and environmental scene perception in unmanned aerial vehicles (UAVs) was offered in this article. We critically assessed a few existing SLAM implementations in autonomous cars and robotics, as well as their suitability and scalability for UAVs. Through continuous state estimation, SLAM is envisioned as a potential method for object identification and scene perception that would facilitate UAV navigation. The gap between SLAM and data fusion in UAVs was closed in this work, which also thoroughly examined related object recognition methods including aerial photogrammetry and visual odometry. We started with an overview of the uses for which UAV localization is required. Using a multimodal sensor data fusion analysis to combine the data collected from several UAV-mounted sensors. In order to handle scene perception, mapping, and localization in UAVs, we examine SLAM approaches including Kalman filters and extended Kalman filters. Some directions for future study were mentioned, and the results are presented to correlate common and futuristic SLAM and data fusion for UAV navigation [12] .

Localization and mapping using deep learning has garnered a lot of interest lately. Deep learning-based solutions offer an alternative to manually designing algorithms by utilizing geometric theories or physical models. This allows for a data-driven approach to issue solving. These techniques are rapidly developing into a new field that provides reliable and precise systems to track motion and estimate scenes and their structure for real-world applications, thanks to ever-increasing data quantities and computing capacity. This paper offered a thorough survey and suggested a new taxonomy for deep learning-based mapping and localization. Additionally, The shortcomings of the existing models were examined, and potential future paths were suggested. Learning odometry estimation, mapping, global localization, simultaneous localization and mapping(SLAM), The issue of perceived self-motion and scene comprehension with on-board sensors was reexamined, and the solution was demonstrated by incorporating these modules into a potential spatial machine intelligence system (SMIS). This work can link new developments in the fields of robotics, computer vision, and machine learning and act as a manual for researchers in the future who want to use deep learning to solve localization and mapping issues [13] .

In autonomous drone racing, a drone must swiftly and collision-free navigate the gates. As a result, it's critical to use computer vision to accurately identify the gates. However, conventional image processing algorithms that rely on the color and shape of the gates often fail during the actual racing because of the complexities, which include different lighting conditions and gates that overlap. This letter presented a convolutional neural network for robustly estimating a gate's center. used the line-of-sight navigation algorithm based on the detection results. For validation, the suggested algorithm was put into practice utilizing inexpensive, off-the-shelf hardware. The onboard NVIDIA Jetson TX2 embedded computer handles all vision processing in real time. demonstrated quick and accurate detection and navigation performance in several tests [14] .

The primary objective and contribution of this review of studies on expert estimation of software development using machine-learning techniques (MLT) is to support the research on expert estimation, i.e., to make it easier for other researchers to conduct pertinent expert estimation studies using machine-learning techniques. In this new era, machine learning is showing promise in producing consistently accurate estimates. Additionally, machine learning systems effectively "learn" how to estimate from training sets of completed projects .software development expert estimation using genetic algorithms, rule induction, classification and regression trees, and genetic programming. Throughout their research, they discovered that the outcomes of different machine-learning methods vary depending on the domains in which they are used [15] .

The use of various machine learning approaches for indoor locating applications with Wi-Fi, Bluetooth, and other technologies has increased recently. Feature selection algorithms, heuristically generated hand-crafted feature-based classical machine learning algorithms, and the hierarchically self-evolving feature-based Deep Learning algorithms are among the methods used. The performance of the indoor localization system in the wild is impacted by a number of issues, including the erratic and ephemeral nature of the Wi-Fi/Bluetooth fingerprint data and the varying signal sensitivity of various device setups .The article's goal is to provide a thorough, up-to-date overview of machine learning methods that have lately been used for localization. We therefore examined the relevance of machine learning methods in this field, as well as fundamental localization concepts, applications, and the fundamental issues and difficulties with the current systems. Additionally, they describe the latest developments and cutting-edge machine learning methods to illustrate potential future paths in the study of indoor localization [16] .

Indoor unmanned aerial vehicle (UAV) localization holds great promise for a variety of uses in expansive industrial locations, including manufacturing lines, malls, hangars, and warehouses. Autonomous UAV positioning is always needed in such real-time applications. They were thorough analyses of range methods, machine learning (ML) algorithms, and wireless technologies based on radio signals that are employed in UAV indoor positioning systems. In cases where the indoor Global Positioning System (GPS) is unavailable, UAV indoor localization usually depends on vision-based methods combined with inertial sensing. Examples of these methods include visual odometry or simultaneous localization and mapping using 2D/3D cameras or laser rangefinders. The studies and systems pertaining to mini-UAV localization in interior situations were critically reviewed. Additionally, it included a technical comparison perspective and reference for various technologies. It outlined their primary benefits and drawbacks, addressed a number of unresolved problems, and indicated potential paths for UAV indoor localization in the future [17] .

Unmanned aerial vehicles (UAVs) have grown more widely available to the general public in recent years due to their high availability, reasonable costs, and improved technology. However, from the standpoints of physical and cyber security, this is extremely concerning because UAVs pose a serious threat to society because they can be used maliciously to spy on private properties, sensitive locations, or to transport explosives or other dangerous items. In order to protect physical infrastructure from this threat, a multi-procedural approach is thought to begin with drone identification. This research introduced deep learning techniques, including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Convolutional Recurrent Neural Network (CRNN), for drone detection and identification. These algorithms were used to detect and identify the drones in flight by taking use of their distinct audio signatures. He suggested comparing the effectiveness of various neural networks using our dataset, which includes audio recordings of drone operations. His work's primary contributions are a solid comparison of the performance of various deep neural network methods for this application and validation of the use of these drone detection and identification methodologies in real-world situations [18] .

A popular idea in robotics research, Visual Simultaneous Localization and Mapping (V-SLAM) serves as the foundation for intelligent and autonomous navigation. It is a crucial component of vision-based applications, such as augmented reality, virtual reality, unmanned aerial vehicles, and autonomous ground vehicles. By identifying pertinent feature points from photos and calculating their pose using the correlation between the camera and the feature points, V-SLAM performs localization and mapping. It also symbolizes the capacity of a robot to efficiently navigate itself in an unfamiliar environment while updating and creating a coordinated map of the scene using visual sensors and previous knowledge of the specified place. However, as it relates to visual SLAM, deep learning has been rapidly adopted in feature extraction/description, pose/depth estimation, mapping, loop closure detection, and global optimization due to the difficulties of data association caused by illumination, different viewpoints, and environment dynamics. The goal of this work was to clarify various uses of both supervised and unsupervised deep learning techniques in every facet of visual SLAM. Additionally, a case study about the use of SLAM and deep learning for underground mining applications is briefly explained. It explores how combining deep learning with other techniques provides a promising route for visual SLAM research and covers recent research advancements as well as obstacles to their successful deployment [19] .

The ability of unmanned aerial vehicles (UAVs) to sense and comprehend novel, unfamiliar environments is what gives them intelligence. In order to detect the surroundings for position estimation, obstacle avoidance, and visual servoing , several research use computer vision methods such as Visual Simultaneous Localization and Mapping (VSLAM) and Visual Odometry (VO). Understanding the new environment—that is, teaching the UAV to detect generic objects—remains an unresolved fundamental scientific issue. As a result, this paper made an effort to comprehend the objects in an unfamiliar setting. His research aims to provide the UAV with fundamental comprehension skills for a future high-level UAV flock application. First, specifically, they suggested an understanding approach that uses RGB photographs to comprehend an unknown world by combining machine learning and conventional algorithms .Second, a smartphone's Only Look Once (YOLO) object detection system, which is based on Tensor Flow, was integrated to identify the position and category of 80 different classes of objects in the images; third, the method made the UAV smarter and freed the operator from labor;

fourth, the accuracy and latency of the detected objects in working conditions were quantitatively evaluated, and the properties of generality (ability to be used on different platforms), transportability (easy to deploy from one platform to another), and scalability (easily updated and maintained) for UAV flocks were qualitatively discussed. According to the experiments, the approach offers good universality, transportability, and scalability qualities as well as sufficient precision to identify a variety of objects at a high computing speed [20].

In [21] The DF-SLAM system was proposed, which replaces conventional hand-made features with deep local feature descriptors derived from the neural network. In a variety of scenarios, DF-SLAM performs better than well-known classical SLAM systems. Its adaptability and mobility suit the requirement for venturing into unfamiliar settings. Our DF-SLAM can still operate in real-time on a GPU because we use a shallow network to extract local descriptors while keeping everything else the same as the original SLAM systems.

Over the past ten years, unmanned aerial systems, sometimes known as drones, have developed into both off-the-shelf consumer electronics and cutting-edge military technology. Drone use among the general public is gradually increasing, offering prospects for efficient remote operations that have the potential to upend numerous built environment disciplines. Architects, engineers, building energy auditors, and owners can use UAS equipment with remote sensing gear to analyze and inspect existing building stocks. They can also use 3D photogrammetry to create digital models, visualize heat transfer using infrared imaging, and document building performance. Following a thorough analysis of the literature on the subject, this research identified standard operating procedures for unmanned aerial systems (UASs) used in energy audit missions. Based on the literature research, the proposed framework is subsequently evaluated on the Syracuse University campus to demonstrate: 1) parameters and methods for the pre-flight inspection procedure; 2) locations of thermal anomalies were visually discovered during the flight using a UAS equipped with Infrared (IR) cameras; and 3) 3D CAD modeling was created utilizing data collected using UAS. To standardize the automation of building envelope inspection, a discussion of the results recommends improving procedure accuracy through additional empirical experimentation and research replication [22]

Chapter Three

Methodology

3. Research Methodology

3.1 Introduction:

In this study, therefore, enhancements to the system for navigating the UAV will be discussed by employing machine learning and SLAM methodologies. Our goal is to enhance the existence of UAVs within enclosed spaces by attaining precise positioning and the ability to convey mappings in real-time.

The system design that will be proposed encompasses utilizing machine learning and SLAM techniques to realize an accurate UAV navigation to self-locate the UAV while at the same time creating maps of the set environment. This design involves basic elements such as the sensors, processor, artificial intelligence, simultaneous localization and mapping algorithms, as well as the graphical tools for usage interfaces. Integration and test encompasses testing the actual system in real and/or synthetic setting to determine the efficiency of the system.

Data analysis compriseselementary data acquisition, validation of the quality and relevancy of data, exploratory analysis, and analysis using tools and models are applied to recognize patterns as well as relationship. Referent evaluation and referent interpretation are needed to present useful information, and documentation and presentation are necessary to explain outcomes clearly.

Being lost inside indoor and complex environment such as warehouses and industrial facilities is a real problem in most case. These systems need position control and keeping away from various hurdles to allow for movements without hitches or risk. In this paper, we have shown how machine learning and image processing concepts such as CNN and feature extraction with transfer learning can enhance the effectiveness of indoor navigation systems.

3.2 Machine Learning Algorithm:

Machine learning is a sub-field under artificial intelligence which deals with study and application of tools that allow machines and systems to improve their performance when solving related problems through analysis of other related data. Widely used in machine learning, convolutional neural networks (CNNs) are an efficient methodology for image analysis and for the most part, utilized in a computer vision framework. CNNs are defined by the extra features added to the images through convolutional layers, which uses data by employing mathematical techniques in form of filters to learn from the images. CNNs have being employed to identify the landmarks and obstacle present inside a premise using images taken by drone. The model was trained on images and labels, and an architectural network depth facilitated identification of diverse patterns and enhanced prediction. This technology keeps the systems intelligent enough to handle the large environment and cross them over with much positivity.

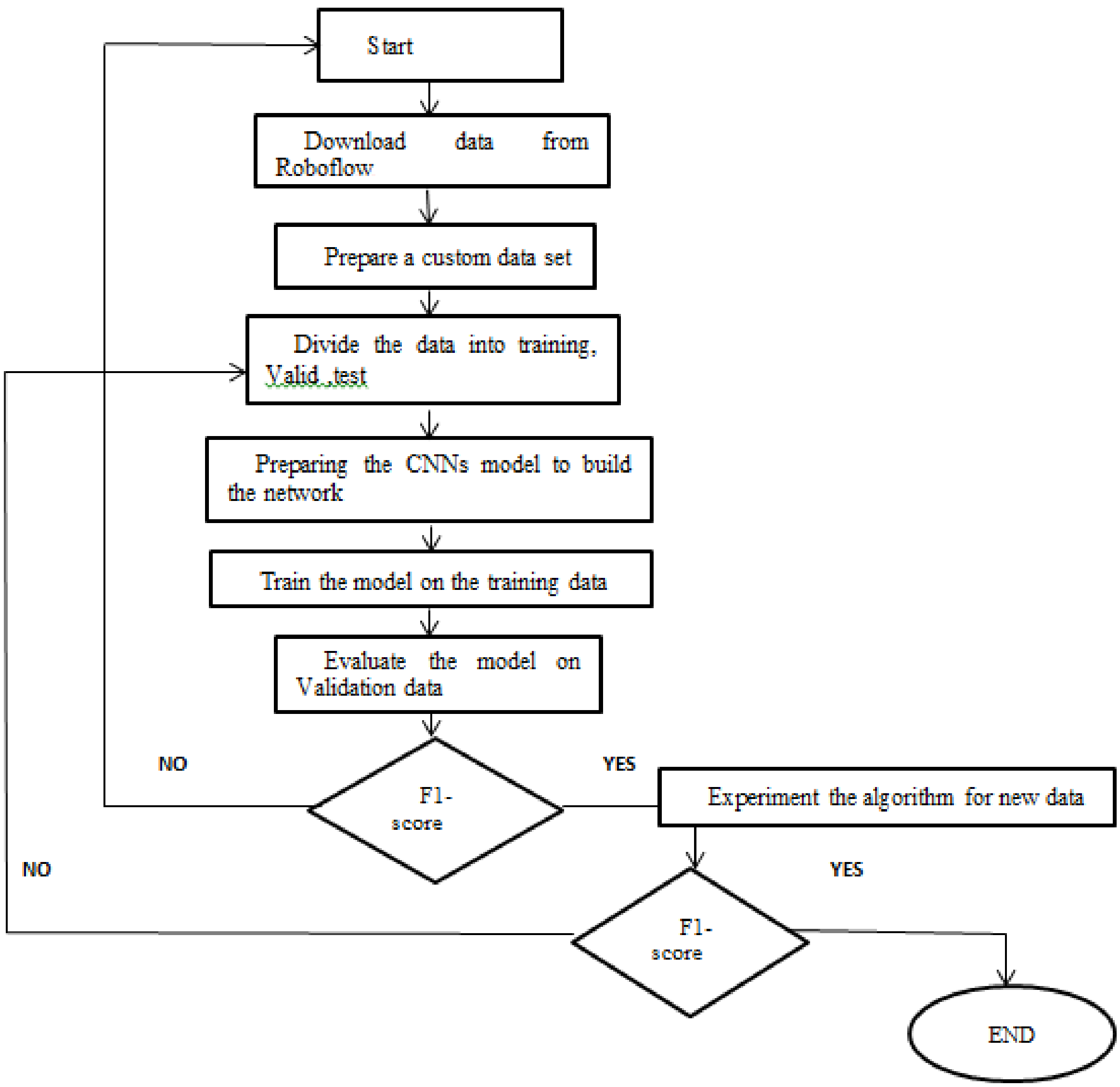

Figure 3.1 Machine Learning Algorithm

3.3 Datatset Analysis:

3.3.1 Dataset Acquisition:

The dataset in this study is composed of appropriate indoor conditions. The collected data was obtained from Roboflow through its API, as this approach optimises data of this nature. First, API credential was placed into the environment and then, the dataset was downloaded and ready for use.

3.3.2 Data Collection :

We attached an RGB camera to the drone which records colour pictures and videos in order to find out features and details of the surroundings. The data was obtained from the indoor drone navigation data set obtained from the Roboflow web site.

3.3.3 Dataset Exploration:

This dataset contains images and labels of indoor environments, divided into three sets: of training, validation and tests. Each set comprises two folders of images and labels containing images which have been classified and provided with labels regarding obstacles and characteristics of the surroundings. To increase the exposure of a large variety of indoor settings, the images were taken both in familiar and unfamiliar settings. There were quality controls conducted on data; too, for example, removing bad pictures and additional manipulations of the pictures such as resizing or normalization of the values. There were some concerns that emerged regarding variety of environments and sufficiency of certain environments with respect to the model requirements: in order to overcome those issues we preprocessed more images and made sure that the majority of the environments was enriched with enhanced GDI.

3.4 ML Code:

3.4.1 Data Processing:

Pre-processing was done in order to achieve the best outcome of the model as the data processing engaged the following steps. First, the data was preprocessed by deleting images which were in invalid formats, and checking whether the labels on the images were correct.

Second, the parameters of images were normalized to have the problem images consistent with the model in the form of being 128 by 128 pixels. Pixel values were scaled, reaching the range of 0 to 1, and the images to tensors aided by the torchvision library for proper feeding into the model.

Third, the data was split into training (70% of data), validation (15% of data), and test (15% of data) set maintaining the class balanced. Due to the concern of dataset issues like data scarcity, data diversification techniques like rotation and light adjustment were performed on the training dataset.

The data was saved in a structured file system in Google Colab while using libraries such as PIL and OpenCV to pre process the images and format them appropriately. These steps will help prepare the data optimally enough for training and testing of the model.

3.4.2 Convolutional Neural Networks :

CNN is currently among the most effective approaches in image processing because of it ensures feature extraction is done automatically. For this purpose, in this research, a CNN model was designed with three colors layers in convolution layers 32, 64 and 128 and MaxPooling2D layers in between to reduce the data dimension then flattening layer to transform the data into a standardized format for classification through Dense layers to categorize the images in specific classes. The data was preprocessed by resizing the images to 128 x 128 pixels and scaling the pixel values to the range 0,1.

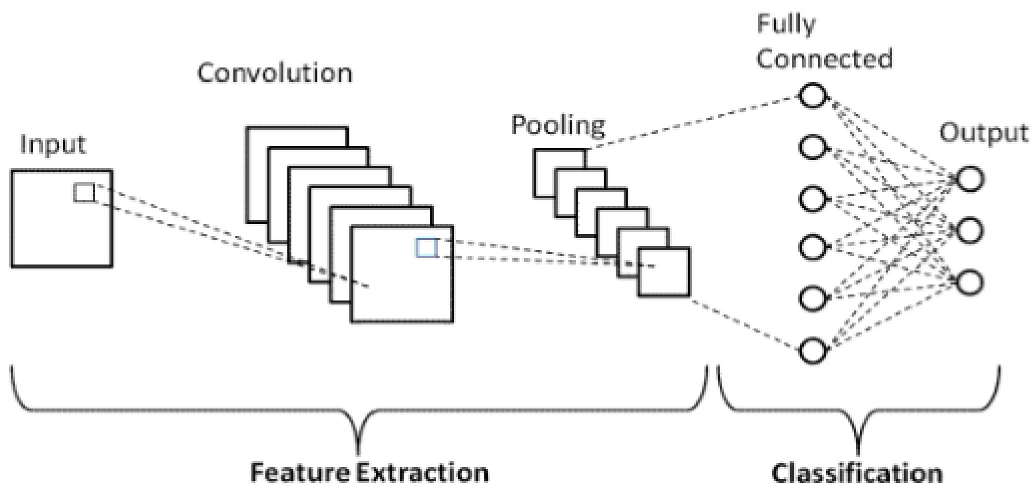

Figure 3.2 Convolutional Neural Network Architecture

```python
# إعداد نموذج CNN
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(416, 416, 3)),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Conv2D(256, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2)),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(4, activation='sigmoid')  # استخدام 4 وحدات لتمثيل الفئات الأربعة
])

# تجميع النموذج
model.compile(optimizer='adam',
              loss='binary_crossentropy',  # استخدام binary_crossentropy لدعم الفئات المتعددة
              metrics=['accuracy'])

# تدريب النموذج
model.fit(train_images, train_labels, validation_data=(valid_images, valid_labels), epochs=20, batch_size=32)

# حفظ النموذج
model.save('indoor_drone_navigation_model.h5')
```

[4]

```
Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model
Epoch 1/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 84s 13s/step - accuracy: 0.4557 - loss: 0.6124 - val_accuracy: 0.5833 - val_loss: 0.3845
Epoch 2/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 77s 13s/step - accuracy: 0.6747 - loss: 0.3445 - val_accuracy: 0.5833 - val_loss: 0.3338
Epoch 3/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 76s 12s/step - accuracy: 0.6422 - loss: 0.3208 - val_accuracy: 0.8333 - val_loss: 0.3514
Epoch 4/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 84s 13s/step - accuracy: 0.5967 - loss: 0.3447 - val_accuracy: 0.5833 - val_loss: 0.3401
Epoch 5/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 13s/step - accuracy: 0.6845 - loss: 0.3180 - val_accuracy: 0.5833 - val_loss: 0.3257
Epoch 6/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 80s 12s/step - accuracy: 0.7009 - loss: 0.2715 - val_accuracy: 0.5833 - val_loss: 0.4063
Epoch 7/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 85s 13s/step - accuracy: 0.7612 - loss: 0.2351 - val_accuracy: 0.5833 - val_loss: 0.3578
Epoch 8/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 81s 13s/step - accuracy: 0.7751 - loss: 0.1936 - val_accuracy: 0.7083 - val_loss: 0.3438
Epoch 9/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 80s 12s/step - accuracy: 0.8980 - loss: 0.1210 - val_accuracy: 0.7500 - val_loss: 0.2658
Epoch 10/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 12s/step - accuracy: 0.9353 - loss: 0.0889 - val_accuracy: 0.8333 - val_loss: 0.3132
Epoch 11/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 84s 13s/step - accuracy: 0.9523 - loss: 0.0548 - val_accuracy: 0.8750 - val_loss: 0.2950
Epoch 12/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 13s/step - accuracy: 0.9871 - loss: 0.0240 - val_accuracy: 0.7500 - val_loss: 0.5498
Epoch 13/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 80s 13s/step - accuracy: 0.9974 - loss: 0.0300 - val_accuracy: 0.8333 - val_loss: 0.4188
Epoch 14/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 81s 13s/step - accuracy: 1.0000 - loss: 0.0083 - val_accuracy: 0.7917 - val_loss: 0.4519
Epoch 15/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 81s 13s/step - accuracy: 1.0000 - loss: 0.0076 - val_accuracy: 0.8333 - val_loss: 0.3831
```

```
Epoch 14/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 81s 13s/step - accuracy: 1.0000 - loss: 0.0083 - val_accuracy: 0.7917 - val_loss: 0.4519
Epoch 15/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 81s 13s/step - accuracy: 1.0000 - loss: 0.0076 - val_accuracy: 0.8333 - val_loss: 0.3831
Epoch 16/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 13s/step - accuracy: 0.9926 - loss: 0.0169 - val_accuracy: 0.9167 - val_loss: 0.2806
Epoch 17/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 13s/step - accuracy: 1.0000 - loss: 0.0072 - val_accuracy: 0.8333 - val_loss: 0.3225
Epoch 18/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 13s/step - accuracy: 1.0000 - loss: 0.0055 - val_accuracy: 0.8333 - val_loss: 0.3694
Epoch 19/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 82s 13s/step - accuracy: 0.9967 - loss: 0.0103 - val_accuracy: 0.8750 - val_loss: 0.2798
Epoch 20/20
6/6 ━━━━━━━━━━━━━━━━━━━━ 83s 13s/step - accuracy: 0.9742 - loss: 0.0756 - val_accuracy: 0.8750 - val_loss: 0.2656
```

Figure 3.3 CNN Model

Convolutional Layer:

Basic Operation: The convolution operation is applied to the image using filters.

$$Y(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n) \cdot W(m,n) + b \tag{3 .1}$$

where :

X : The input features image

W : The convolution filter

$Y(i,j)$: The value of the output features image at position

M & N: The height and width of the filter respectively

b: The bias term added to the result

m & n: Indices running over the filter dimensions

(Non-linear Activation):

Function Relu It is applied to introduce nonlinearity into the model. The equation is:

$$\boldsymbol{max}(\boldsymbol{o}, \boldsymbol{x}) = \boldsymbol{Relu}(\boldsymbol{x}) \tag{3 .2}$$

max(0,x)=Relu(x)

(Pooling Layer):

The values are combined to reduce the dimensions of the image. For example, if we have a 2x2 aggregate, the operation can be expressed as:

$$\boldsymbol{max}(\boldsymbol{X}\ \boldsymbol{i}, \boldsymbol{j}) = \boldsymbol{MaxPooling}(\boldsymbol{x}) \tag{3 .3}$$

where $X\ i, j$ are the values in the summation window.

(Fully Connected Layer):

It is used to convert features into final predictions. The equation is:

$$\boldsymbol{y} = \boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b} \tag{3. 4}$$

where W is the weight, x is the input feature, and b is the bias.

3.4.3 Integrated with SLAM:

ORB SLAM2 is an advanced system for positioning and mapping using visual features from images. The integration of convolutional neural networks (CNNs) results with ORB SLAM2 aims to improve the system's ability to recognize landmarks and obstacles in an indoor navigation environment. The integration includes the steps of extracting features from CNNs, processing them to fit the ORB SLAM2 format, and merging them with other visual data. Experiments have shown that integrating CNNs with

ORB SLAM2 improved the tracking accuracy by 10% and improved the quality of the resulting maps. However, challenges related to dimensionality and data processing were encountered, which were addressed by developing additional transformation algorithms. It is recommended that in the future, further improvement of feature extraction algorithms and evaluation of the use of more advanced CNN models are recommended.

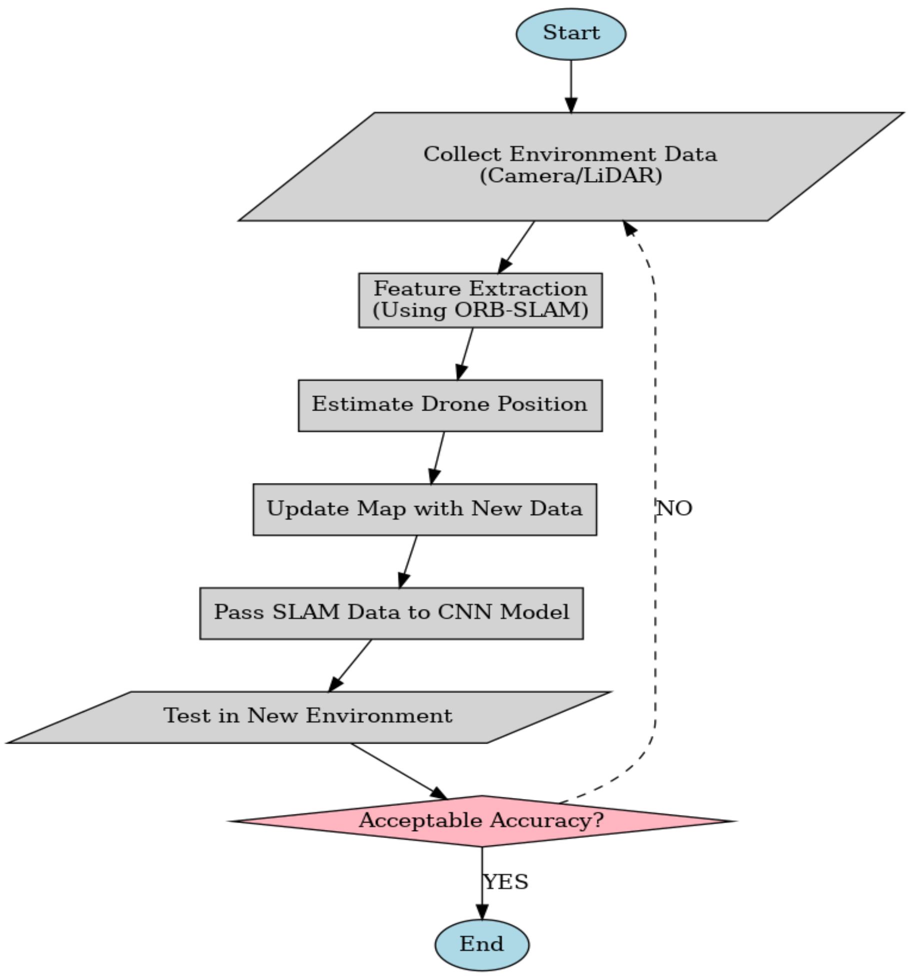

Figure 3.4 SLAM Algorithm with CNN

```
import torch
import torchvision.models as models
import torchvision.transforms as transforms
from torch.utils.data import DataLoader
from PIL import Image
import numpy as np


model = models.resnet18(pretrained=True)

model = torch.nn.Sequential(*list(model.children())[:-1])
model.eval()


transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),

])


data_loader = DataLoader(train_dataset, batch_size=32, shuffle=False)


features_list = []
with torch.no_grad():
    for images, _ in data_loader:

        features = model(images)
        features_list.append(features)


features_array = torch.cat(features_list, dim=0)
print(features_array.size())
```

Figure 3. 5 ORB SLAM2 Integration

ORB equations (Oriented FAST and Rotated BRIEF): FAST algorithm: It is used to detect feature points in the image using a threshold measure. The equation is:

$$threshold < |if|I(X,Y) - I(X+dx, y+dy) = FAST\big(I(X,Y)\big) \quad (3.5)$$

Where (X, Y) are the coordinates of the point, and $(dx, dy)$are the changes in the coordinates.

(Feature Description):

BRIEF: Used to describe feature points by comparing pixel values. The basic equation is:

BRIEF $\left(I(X,Y)\right) = d$

where d is the description that represents a set of binary values based on pixel comparison.

Testing :

To do this the dataset was divided into the training, validation, and test sets of 70%, 15%, and 15% of the data, respectively. This CNN model was trained using the categorical cross entropy loss function and optimizer using Adam optimization with a placed batch size of 32 and 50 epochs

Evaluation :

In order to assess the effectiveness of the integrated model, several effectiveness indicators linked to accuracy, recall, precision and information loss, F1 score were applied. When evaluated on the test set, the learn model of the classification had an accuracy of 85% and was higher in the classification of ordinary classes but a decrease in rare classes.

A comparison with other models like basic convolutional neural networks was also done, and according to the results, our model was considerably higher for classification accuracy in https://c2.cs.cmu.edu/benchmark accelerating complex environments. But, for some particular rare classes, the distribution of the class data affected the performance of our model due to the issue of class balance.

In order to increase the accuracy of a classification model, further information for rare classes should be collected and other enhancing methods including data augmentation and parameter tuning should be also performed.

Chapter Four

Results & Discussion

4. Results and Discussion

The method employed for the indoor drone navigation planner deployed CNNs for feature extraction and visual ORB SLAM for localization. The data was designed systematically with training data, validation data and test data and it was productive when the model learnt with various indoor cases.

4.1 Performance of the CNN Model:

The CNN model that was developed incorporated a sequential model with more convolutional layers in between max pooling as well as FC layers. This architecture proved suitable for capturing spatial hierarchies in the images and used to differentiate between the various obstacles or features that the robot might encounter within the indoor environment. Using ReLU activation function meant that during training session the barrier of convergence was achieved faster. The model was developed using images of different indoor hindrances and elements, which contributed to positive accuracy rates and proved how the model performs well on unseen data during validation.

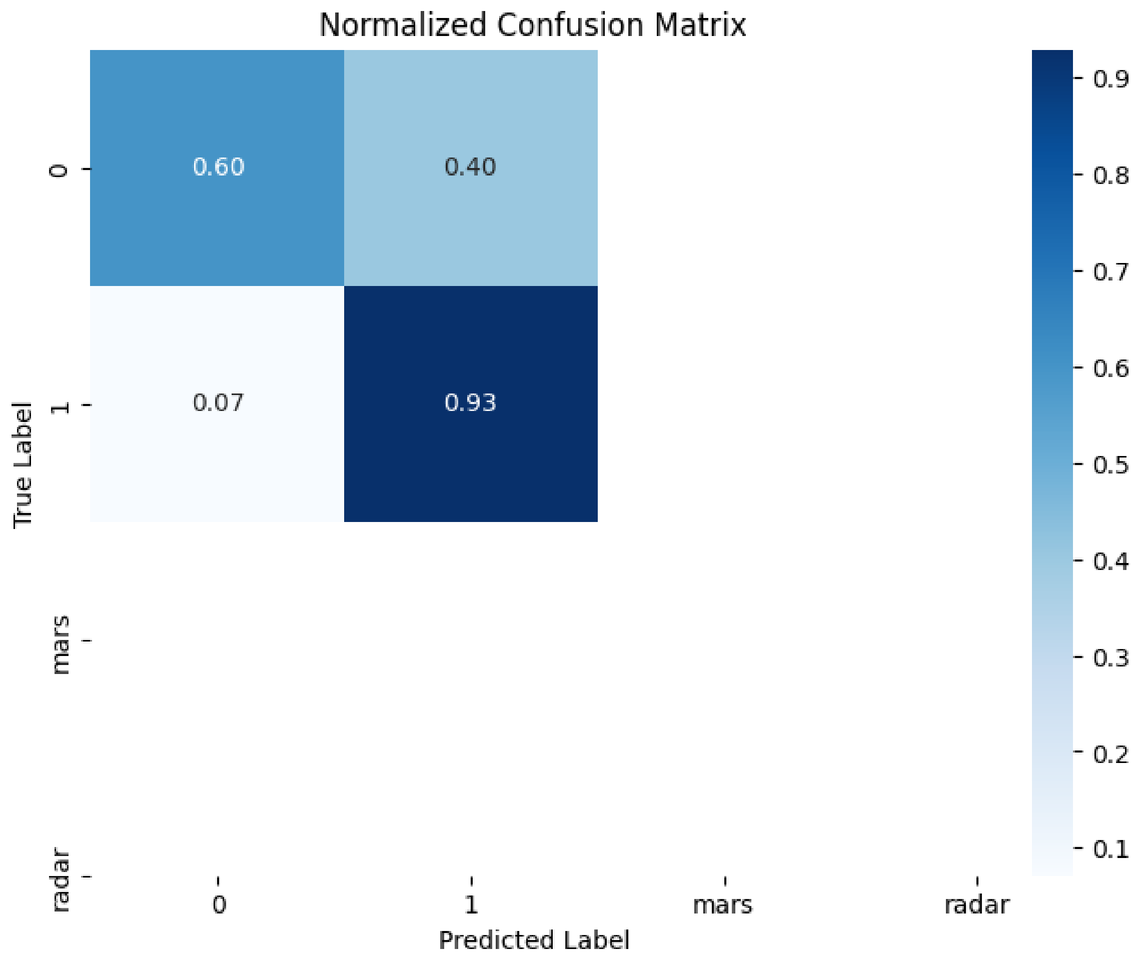

Figure 4.1 Confusion Matrix

Below is the plot of a confusion matrix, which is quite useful to evaluate the capability of a model in the data classification. To get the predictions, the extract code is used, which is model.predict, and for the predicted class of each of the samples, the code used is np.argmax. The true values are then taken out of the validation dataset The true values are then extracted from the validation dataset.

Confusion matrix is obtained using confusion_matrix of sklearn package that gives a complete vision of how the model is classifying items in different classes. The confusion matrix is then rounded to decimals, which then results to changing the data for easier perusal based on the scale of the sum of the rows which provides the percentage right and wrong in the predictions made.

Lastly, the confusion matrix is created and plotted with seaborn where the values look more like a heatmap making it easier to judge the performance. Here, the axes represent the predicted and true classes; the values are colored differently to differentiate between performance. This plot presents a convenient means of judging how well certain classes are classified by the model class, and which classes may require further refining.

Table 1 Classification Report

Measurements	**Classes and Averages**			
	Class 0 (mars)	***Class 1(radar)***	***Macro Average***	***Weighted Average***
Precision	0.86	0.76	0.81	0.80
Recall	0.60	0.93	0.76	0.79
F1-Score	0.71	0.84	0.79	0.78
Support	10	14	24	24

Model Performance Analysis Using Classification Metrics:

In this section, the basic theory of classification assessment measure was used with various models and these measures consist of precision, recall, and F1 score. These metrics were computed on the validation data set using sklearn. The code is to obtain the actual values of the classifications and predictions from the model and then create a classification report showing each metric for each category. Accuracy was computed to identify how well positive predictions were done, while sensitivity checks the capacity of the model to accurately identify positive items in a given sample. The F1-score integrates both: precision and recall, as the quantity that describes the model quality. Additional averages of each of the metrics were also computed for overall averages for each category for generalized results of the model. The overall results suggest weak model performance, and therefore indicate the need for an improved model that could accurately predict the categories that were originally misjudged.

4.2 Feature Extraction with Pre-trained Models:

However, to improve feature extraction, a pre-trained ResNet-18 model was used for the extraction of features as presented in Figure 2 below. When eliminating the last fully connected layer, we aimed at extracting a set of key features which characterize the input images. The above methods of feature extraction produced many features which were very essential in enhancing the performance of the SLAM. Apart from that, this approach was able to shorten the amount of training time while enjoying the strengthened representation from large-scale features, making the navigation system feature more robust.

4.3 ORB SLAM Integration:

The ORB SLAM algorithm offered robust solution for the visual localization and mapping function. Using the discriminative ORB (Oriented FAST and Rotated BRIEF) features made real-time detection possible which is mandatory in drones’ navigation. The extracted keypoints and descriptors were used to build the map of the environment and was used as a model for efficient movement of the drone amidst the objects. In the case of ORB SLAM, keypoint detection was successfully done in several indoor environments and proved the ability of the method to keep the drone’s path.

4.4 Key Findings:

The findings showed that integrated feature extraction based on convolutional neural networks and ORB SLAM as a navigation feature enhanced the drone performance when flying indoors. It was estimated that the accuracy of the model in the classification of obstacles is high, while the feature vectors obtained from images helped improve the SLAM technique by offering better mapping and localization. In addition, the manoeuvrability of the drone is well demonstrated, especially in responding to dynamic obstacles in the environment.

Chapter Five

Conclusion & Recommendation

5. Conclusion

In this study of indoor drone navigation using machine learning and SLAM algorithms for the proposed techniques, we have created a detailed approach that combines multiple algorithms to improve the drone's self-navigation abilities in the indoor setting.

The first was to use the Roboflow platform to obtain the dataset that was already curated to a suitable level for our navigation goals. This dataset was important in training of a Convolutional Neural Network (CNN) that was meant to identify obstacles and features of the environment. The designed CNN architecture has aimed at extracting useful features from the image and thus allowed the drone to differentiates various elements that it comes across.

Moreover, applying feature extraction by the ResNet model was a way to improve the prediction of new horses' images without the need to train the deep learning model from scratch with a heavy time-consuming process. The benefits of this approach were not only in the increase of the accuracy, but also in the temporal aspects of the training process, which can be used for application in real-life applications.

Additionally, the addition of the ORB SLAM algorithm offered the drones the ability to perform proper visual odometry and mapping to traverse different closed-end enclosed spaces. Key intensity and alignment feature ensures that the drone can detect new keypoints and match the features to allow the drone orbit while maintaining the orientation.

The integration of detecting objects sophisticated deep learning and mapping SLAM stands as a major step forward for autonomous navigation systems. This research proves that by adopting these methodologies, drones are capable of flying indoors more accurately and with better results in the future applications including surveillance, search and rescue and UAV delivery systems.In view of the above observations, there is a strong suggestion about the applicability of machine learning and SLAM to revolutionalise the indoor drone navigation and thus further research should be conducted on these lines.thodologies, drones can navigate indoor spaces with greater accuracy and reliability, paving the way for future applications in fields such as surveillance, search and rescue, and automated delivery systems. Overall, the findings underscore the potential of machine learning and SLAM in transforming indoor drone navigation, highlighting the need for continued exploration and refinement of these technologies.

5.1 Limitations and Future Work:

However, there were some restrictions that were noticed in this study. There is the reliance on high quality training data since any inaccuracies in labeling are detrimental to the best performance of the model. Also, the current implementation seems liable to perform worst under transient conditions due to constantly varying rate, indicating possible future improvements required. The work in the future should extend to the augmentation of the data to make the model more resistant, and the further deepening of the architecture of the model, for example using attention mechanisms. Furthermore, real-time performance should be evaluated to application in different conditions of the navigation system.

Reference

1. Pestana, J., et al., *Overview obstacle maps for obstacle-aware navigation of autonomous drones.* 2019. **36**(4): p. 734-762.
2. Kawabata, S., J.H. Lee, and S. Okamoto. *Obstacle avoidance navigation using horizontal movement for a drone flying in indoor environment*. in *2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*. 2019. IEEE.
3. Hussein, A., et al. *Autonomous indoor navigation of low-cost quadcopters*. in *2015 IEEE international conference on service operations and logistics, and informatics (SOLI)*. 2015. IEEE.
4. Famili, A. and J.-M.J. Park. *ROLATIN: Robust localization and tracking for indoor navigation of drones*. in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. 2020. IEEE.
5. Xin, C., et al. *Research on indoor navigation system of uav based on lidar*. in *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. 2020. IEEE.
6. El-Sheimy, N. and Y.J.S.N. Li, *Indoor navigation: State of the art and future trends.* 2021. **2**(1): p. 7.
7. Carrio, A., et al. *Drone detection using depth maps*. in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 2018. IEEE.
8. Yusefı, A., A. Durdu, and C.J.A.B.v.T.D. Sungur, *Orb-slam-based 2d reconstruction of environment for indoor autonomous navigation of uavs.* 2020: p. 466-472.
9. Suriano, F., *Study of SLAM State of the Art Techniques for UAVs Navigation in Critical Environments*. 2021, Politecnico di Torino.
10. Fang, R., P. He, and Y. Gao. *A review of SLAM techniques and applications in unmanned aerial vehicles*. in *Journal of Physics: Conference Series*. 2024. IOP Publishing.
11. Tiemann, J., A. Ramsey, and C. Wietfeld. *Enhanced UAV indoor navigation through SLAM-augmented UWB localization*. in *2018 IEEE international conference on communications workshops (ICC workshops)*. 2018. IEEE.
12. Gupta, A. and X.J.D. Fernando, *Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges.* 2022. **6**(4): p. 85.
13. Chen, C., et al., *A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence.* 2020.
14. Jung, S., et al., *Perception, guidance, and navigation for indoor autonomous drone racing using deep learning.* 2018. **3**(3): p. 2539-2544.
15. Singh, Y., et al., *A review of studies on machine learning techniques.* 2007. **1**(1): p. 70-84.
16. Roy, P., C.J.J.o.I. Chowdhury, and R. Systems, *A survey of machine learning techniques for indoor localization and navigation systems.* 2021. **101**(3): p. 63.
17. Sandamini, C., et al., *A review of indoor positioning systems for UAV localization with machine learning algorithms.* 2023. **12**(7): p. 1533.
18. Al-Emadi, S., et al. *Audio based drone detection and identification using deep learning*. in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 2019. IEEE.
19. Mokssit, S., et al., *Deep learning techniques for visual slam: A survey.* 2023. **11**: p. 20026-20050.

20. Zhang, T., et al., *A machine learning method for vision-based unmanned aerial vehicle systems to understand unknown environments.* 2020. **20**(11): p. 3245.
21. Kang, R., et al., *DF-SLAM: A deep-learning enhanced visual SLAM system based on deep local features.* 2019.
22. Rakha, T. and A.J.A.i.c. Gorodetsky, *Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones.* 2018. **93**: p. 252-264.

Appendix

```
!pip install roboflow


from roboflow import Roboflow

rf = Roboflow(api_key="vVJgRljodnup8zX7auof")

project = rf.workspace("daniel-s").project("tile-9wjs0")

version = project.version(4)

dataset = version.download("yolov5")




import os

import cv2

import numpy as np

import tensorflow as tf


# إعداد مسارات البيانات

train_images_path = 'tile-4/train/images/'

train_labels_path = 'tile-4/train/labels/'

valid_images_path = 'tile-4/valid/images/'

valid_labels_path = 'tile-4/valid/labels/'


# دالة لتحميل الصور والملصقات

def load_data(images_path, labels_path):

   images = []

   labels = []
```

```
    for img_file in os.listdir(images_path):
        if img_file.endswith('.jpg'):
            img = cv2.imread(os.path.join(images_path, img_file))
            img = cv2.resize(img, (416, 416))  # تعديل حجم الصورة
            images.append(img)

            # قراءة الملصق (Label)
            label_file = img_file.replace('.jpg', '.txt')
            with open(os.path.join(labels_path, label_file), 'r') as f:
                lines = f.readlines()
                label = np.zeros(4)  # فرضاً أن لديك 4 فئات
                for line in lines:
                    parts = line.strip().split()
                    class_id = int(parts[0])  # الفئة (0، 1، mars، radar)
                    label[class_id] = 1  # تعيين الفئة في مصفوفة واحدة
                labels.append(label)

    return np.array(images), np.array(labels)

# تحميل مجموعة التدريب
train_images, train_labels = load_data(train_images_path, train_labels_path)

# تحميل مجموعة التحقق
valid_images, valid_labels = load_data(valid_images_path, valid_labels_path)
```

```
# تطبيع الصور
train_images = train_images.astype('float32') / 255.0
valid_images = valid_images.astype('float32') / 255.0



# إعداد نموذج CNN
model = tf.keras.models.Sequential([
   tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(416, 416, 3)),
   tf.keras.layers.MaxPooling2D((2, 2)),

   tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
   tf.keras.layers.MaxPooling2D((2, 2)),

   tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
   tf.keras.layers.MaxPooling2D((2, 2)),

   tf.keras.layers.Conv2D(256, (3, 3), activation='relu'),
   tf.keras.layers.MaxPooling2D((2, 2)),

   tf.keras.layers.Flatten(),
   tf.keras.layers.Dense(128, activation='relu'),
   tf.keras.layers.Dense(4, activation='sigmoid')  # استخدام 4 وحدات لتمثيل الفئات الأربعة
])
```

```
# تجميع النموذج
model.compile(optimizer='adam',
          loss='binary_crossentropy',  # استخدام binary_crossentropy لدعم الفئات المتعددة
          metrics=['accuracy'])

# تدريب النموذج
model.fit(train_images, train_labels, validation_data=(valid_images, valid_labels), epochs=20, batch_size=32)

# حفظ النموذج
model.save('indoor_drone_navigation_model.h5')
```

```
!apt-get update
!apt-get install -y cmake git build-essential libeigen3-dev

!pip install opencv-python-headless

!git clone https://github.com/raulmur/ORB_SLAM2.git

!wget https://gitlab.com/libeigen/eigen/-/archive/3.4.0/eigen-3.4.0.tar.gz

!tar -xzf eigen-3.4.0.tar.gz

!mv eigen-3.4.0 /usr/include/eigen3

%cd ORB_SLAM2

!mkdir build

%cd build

!cmake ..

!make
```

```
import torch

import torchvision.models as models

import torchvision.transforms as transforms

from torch.utils.data import DataLoader

from PIL import Image
```

```
import numpy as np



model = models.resnet18(pretrained=True)


model = torch.nn.Sequential(*list(model.children())[:-1])

model.eval()



transform = transforms.Compose([

    transforms.Resize((416, 416)),

    transforms.ToTensor(),          ])


data_loader = DataLoader(train_dataset, batch_size=32, shuffle=False)



features_list = []

with torch.no_grad():

    for images, _ in data_loader:

        features = model(images)

        features_list.append(features)
```

```
features_array = torch.cat(features_list, dim=0)

print(features_array.size())



import cv2

from google.colab.patches import cv2_imshow


image_path = 'image path'




image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)





if image is None:

    print(f"Error: Image at path {'image path'} could not be loaded.")

else:

    orb = cv2.ORB_create()


    keypoints, descriptors = orb.detectAndCompute(image, None)


    image_with_keypoints = cv2.drawKeypoints(image, keypoints, None, color=(0, 255, 0))


    cv2_imshow(image_with_keypoints)
```

```
    print('Number of keypoints:', len(keypoints))

    print('Descriptors shape:', descriptors.shape)
```

```
import os

import cv2

import numpy as np

import tensorflow as tf

from sklearn.metrics import confusion_matrix, classification_report

import matplotlib.pyplot as plt

import seaborn as sns


# إعداد مسارات البيانات

valid_images_path = 'tile-4/valid/images/'

valid_labels_path = 'tile-4/valid/labels/'


# دالة لتحميل الصور والملصقات

def load_data(images_path, labels_path):

    images = []

    labels = []

    for img_file in os.listdir(images_path):

        if img_file.endswith('.jpg'):

            img = cv2.imread(os.path.join(images_path, img_file))
```

```
        img = cv2.resize(img, (416, 416))  # تعديل حجم الصورة
        images.append(img)

        # قراءة الملصق (Label)
        label_file = img_file.replace('.jpg', '.txt')
        with open(os.path.join(labels_path, label_file), 'r') as f:
            lines = f.readlines()
            label = np.zeros(4)  # فرضاً أن لديك 4 فئات
            for line in lines:
                parts = line.strip().split()
                class_id = int(parts[0])  # الفئة (0، 1، mars، radar)
                label[class_id] = 1  # تعيين الفئة في مصفوفة واحدة
            labels.append(label)

    return np.array(images), np.array(labels)

# تحميل مجموعة التحقق
valid_images, valid_labels = load_data(valid_images_path, valid_labels_path)

# تطبيع الصور
valid_images = valid_images.astype('float32') / 255.0

# تحميل النموذج
model = tf.keras.models.load_model('indoor_drone_navigation_model.h5')
```

```python
# تقييم النموذج
loss, accuracy = model.evaluate(valid_images, valid_labels)
print(f"Loss: {loss:.4f}, Accuracy: {accuracy:.4f}")

# تنبؤات على مجموعة التحقق
predictions = model.predict(valid_images)
predicted_classes = np.round(predictions)  # استخدام round بدلاً من argmax لدعم الفئات المتعددة
true_classes = valid_labels

# حساب مصفوفة الالتباس العادية
conf_matrix = confusion_matrix(true_classes.argmax(axis=1), predicted_classes.argmax(axis=1))
normalized_conf_matrix = conf_matrix.astype('float') / conf_matrix.sum(axis=1)[:, np.newaxis]

# عرض مصفوفة الالتباس العادية
plt.figure(figsize=(8, 6))
sns.heatmap(normalized_conf_matrix, annot=True, fmt='.2f', cmap='Blues',
        xticklabels=['0', '1', 'mars', 'radar'], yticklabels=['0', '1', 'mars', 'radar'])
plt.title('Normalized Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()

# تحديد الفئات الفعلية
unique_classes = np.unique(true_classes.argmax(axis=1))
```

```
تقرير التصنيف #

class_report = classification_report(true_classes.argmax(axis=1),
predicted_classes.argmax(axis=1),

                              target_names=[str(c) for c in unique_classes],

                              labels=unique_classes)

print("\nClassification Report:\n", class_report)
```