Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
April 29, 2021
A Regional Conference of the Society for Industrial and Systems Engineering

Drone-Based Parking Enforcement

Solomon Barer, Lucy Benack, Jordan Congdon, Alexandra Rizopoulos, Irene Yuan, and Yong Wang

SUNY Binghamton University
Department of Systems Science and Industrial Engineering
Vestal, NY

Corresponding Author: sbarer1@binghamton.edu

Author Note: Barer, Benack, Congdon, Rizopoulos, and Yuan are seniors at SUNY Binghamton as students in the Industrial and Systems Engineering major. During the current academic year, the team has been working under the supervision of university Assistant Professor Yong Wang and Professor Mark Poliks. This senior design project was completed in partial fulfillment of the students' ultimate capstone project.

Abstract: This project aims to develop the functionality of License Plate Recognition (LPR) software as it is processed from the camera feed of an unmanned aerial system (UAS), a quadcopter drone. In the future, this project can be advanced to operate within and under the authority of Binghamton University's Transportation and Parking Services (TAPS). The current endeavor includes an autonomously flying drone, a functioning LPR program, a referenceable database of registered vehicles, and a system to notify the operator of an unregistered license plate. Programmed to read and relay license plate information, the drone will monitor for unregistered vehicles in university parking lots. The program will return registered vehicle information or notification of an unregistered vehicle. The ultimate system should provide reduced costs and environmental benefits. Tasked with developing baseline functionality, this project sets fundamental standards for future success.

Keywords: License Plate Recognition, Unmanned Aerial System, Autonomous Flight, Parking Enforcement

1. Background

The current age of innovation has seen futuristic technologies become integral within everyday life. Commercial and recreational drones embody this shift as they are utilized in various fields and industries. A drone, or an unmanned aerial system (UAS), first used for military purposes, is an unpiloted aircraft or spacecraft that has various roles such as photography, material transportation, traffic management, etc. It is a flying robot that can be controlled remotely or through an autonomous software-based flight control. The basic components of a drone are a power source, propellers and a frame and this frame is often made of durable but lightweight material. Drones are often used because they do not require a person's physical presence to perform tasks, have no mechanical difficulties and do not require rest as they can fly for as long as there is fuel or battery power. Drones have become more ubiquitous and less costly over time, allowing for more commercial use with more industries utilizing its functions and advancing its technologies. Some modern technological accessories to drones include GPS modules, antennas, cameras, Electronic Speed Controllers, and sensors. These accessories have allowed for features such as hover accuracy, obstacle sensory range, altitude hold, flight logs, live video feed, and more. For this endeavor, the DJI Mavic 2 Pro was selected due to its high-quality camera, flight stability, battery life, and omnidirectional obstacle avoidance technology.

As this project was completed in fulfillment of the team's senior capstone project, its advisor tasked the team to explore the application of drones within the university's parking monitoring and operations. While the current goal is to gather information and set standards for future experimentation, the hope is that this project can be advanced to fully operate within and under the authority of Binghamton University's Transportation and Parking Services (TAPS). The system was motivated by the potential economic and environmental benefits of its application. Multiple drones surveying the parking lots, rather than the 2 SUVs currently doing so, would allow a significantly higher number of scans to be recorded. Further, reducing the number of SUVs perpetually driving around campus would decrease emissions and carbon footprint. The system would function under the university's authority, and impact all individuals who park on university premises. This includes resident parkers, commuters, guests, staff and faculty.

SUNY Binghamton has a modern transportation and parking system to monitor its properties. The system's users include students, faculty, staff, guests, and state employees. The system is comprised of 57 parking lots, including 33 commuter lots and 24 residential lots. The current model entails a registered database which houses the license plate number, name, and contact information for users who have registered a vehicle to park on campus. The university deploys two SUVs as part of TAPS, that enforce the university's parking regulations. Currently, the SUVs have two mounted cameras that allow scanning of both sides of a row while driving down the aisle of a parking lot. These cars cyclically monitor the different parking lots under their jurisdiction. It is believed that complementing this operation with drones could lead to economic and environmental benefits. The team seeks to explore the benefits of utilizing drone technology to improve the existing state, and potentially replace the current system in the future. Further research, testing, and implementation may lead to reduced emissions, more accurate readings, and lower costs associated with maintenance and operations. Eventually, the small-scale testing conducted in this project will be expanded, and licensed LPR software will be allocated to implement this endeavor on a university-wide scale utilizing a fleet of synchronized drones. If successful, the system designed in this project will operate concurrently with TAPS software and allow for an optimized, cost-effective operation.

2. Summary of Contributions

In this paper, a new system for parking enforcement is attempted, through the integration with an unmanned aerial system (UAS), to test for feasibility and for improvement on performance of the previous parking enforcement system.

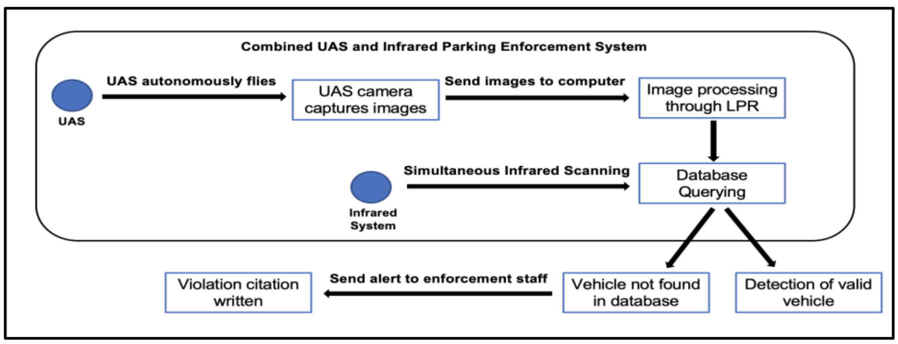

Figure 1. System Structure

The main contributions are as follows:

- Combining the functionality of an Optical Character Recognition (OCR) Program to an UAS. The combined functionality is explored for the two technologies and its limitations and extensions for the purposes of a parking enforcement system.
- Conducting analysis on the feasibility of a drone-based parking enforcement system. A comparison is made between a previously established parking enforcement system utilizing infrared sensors and the possibility of an adoption of or integration with a parking enforcement system utilizing UAS and OCR technologies.
- Inaugurating ideas for future studies of parking enforcement innovation. Ideas include the possibility of automated flight path software, and live feed detection using Internet of Things (IOT) technologies. Other systems, such as an emergency response system, are explored for future research in combined OCR and UAS technologies.
- Developing baseline functionality of a drone-based parking enforcement system. In the baseline functionality includes data and analysis on flight parameters for ideal license plate recognition and OCR.
- Building the foundations of a parking enforcement system utilizing drone technology. Guidelines are established for safety measurements, flight compliance, drone technology limitations, automated flight paths, and drone battery optimization

3. Approach

3.1 Project Schedule

The team began the project by breaking it into major phases, each with subtasks and end dates. These milestones were organized and tracked using an Excel-based Gantt Chart. The Fall 2020 semester work consisted of three phases: Defining, Software, and Gathering Materials; and the Spring 2021 semester consisted of three additional phases: Software Integration, Software Testing, and Execution.

The Defining phase was home to numerous brainstorming sessions, research, meetings, and collaborations. The team met with university TAPS personnel to learn about the current system. Python coding was researched at the request of the project advisor. Finally, the team settled on deliverables and the steps to reach them. The Software phase revolved around the LPR code and autonomous flight program. Researching an LPR code and dissecting its inner workings was a major portion of this phase. Means to accomplish autonomous flight were also explored. The last phase of the fall semester, Gathering Materials, included applying for a budget and purchasing the drone hardware on behalf of the university. A final course report was also compiled for that period of the project.

The spring semester began with further LPR dissection. The referenceable database of license plates was created and research on software development kits (SDK) helped with compatibility and capability struggles. The team is currently working within the fifth stage: Software Testing. Iterative tests and experiments on both the LPR and flight codes are being executed. These will lead to determining the set-up to be most successful. The last phase, called Execution, will see continued tests and expanded tests. Final troubleshooting and user interface modifications will help finalize the system. A comprehensive report will be compiled reviewing the project's procedure and results.

3.2 Methodology and Constraints

Due to the fact that the team did not have access to any active systems or previous progress, the scope of the project varied within its initial phases. As feasibility, and both time and skill constraints, was limiting, the project encountered some challenges. Specifically, lack of access to the official university's LPR program posed the greatest hindrance. Unable to access their program, which also utilized infrared to read the plate characters, the team researched public LPR codes. As such, the team is operating off of a public program from GitHub with a typical desktop computer. Unfortunately, high quality image files taken from the drone are too large for this program to run. Therefore, the team slightly modified its goals with respect to the LPR component, lowering its priority status. Now running lower quality images, it is expected that the LPR accuracy rate will be lower. The team deemed this acceptable because the final system, once operational, will be utilizing the university's licensed program.

Due to the risks involved in airborne activities, the main goal is to establish flight parameter standards. In certain modes, the Mavic 2 Pro has omnidirectional sense-and-avoid technology that will help prevent collisions. In order to get the most successful images, in the safest manner, the team ran a series of tests. These tests were judged on drone position with respect to the environment and the positioning to get centered photos. The drone was positioned at different coordinates with respect to the end of parking spots. This included 1-foot increments from 3 to 10 feet back, and 5 to 9 feet up, creating a matrix of coordinates to test the LPR accuracy at each coordinate.

By gauging the sense-and-avoid technology, establishing forms of autonomous flight, and setting flight parameters, the team should be positioned to run holistic tests and yield promising results. The basis of this project is to develop the fundamentals and standards needed to progress the endeavor further.

3.3 Autonomous Flight System

Achieving autonomous flight can be done in several ways with the Mavic 2 Pro. The first method considered was utilizing the Waypoints 2.0 functionality integrated into the framework of DJI drones, controlled by an interface within the DJI GO application. Waypoints 2.0 is a user interface that displays a satellite image of the user's current location and can be used by adding pins through a screen tap, and the drone will follow a path between multiple waypoints when prompted. Waypoints can be set with a limited level of precision, because the satellite image does not zoom in far enough for the drone to precisely fly a matter of feet away from parked cars. Low altitude flight is required for the project, and the Waypoints 2.0 system contains heightened security safety protocols, preventing flight at lower altitudes, and distance away from obstacles. In the future, different flight modes enabled within the drone that unlock certain abilities and allow varying maneuvers will be explored.

The DJI Developer website offers several open-source resources that increase the capabilities of DJI drones. DJI offers a software development kit (SDK) compatible on all operating systems (i.e., Windows, iOS/MacOS, Linux). The SDK's allow the user to develop their own application with full access to the drone's capabilities, such as flight and taking photographs. The iOS SDK was chosen because it offered an application that is run on mobile devices with an iOS operating system. Staying consistent with the bigger picture of the project scope, the iOS application would be most beneficial in incorporating an autonomous flight system. The iOS application can be run on devices such as an iPhone or iPad, which is a greater convenience for TAPS employees once the system is integrated at Binghamton University. Developing an iOS application is programmed in Objective-C or Swift languages. Objective-C has been used for iOS development since the mid-1980's, while Swift has exclusively been used in the past five years. The DJI iOS Mobile SDK offers equivalent sample applications constructed in Objective-C and Swift. After researching which language would be easier to learn with no background knowledge, experienced individuals recommend learning Objective-C before Swift.

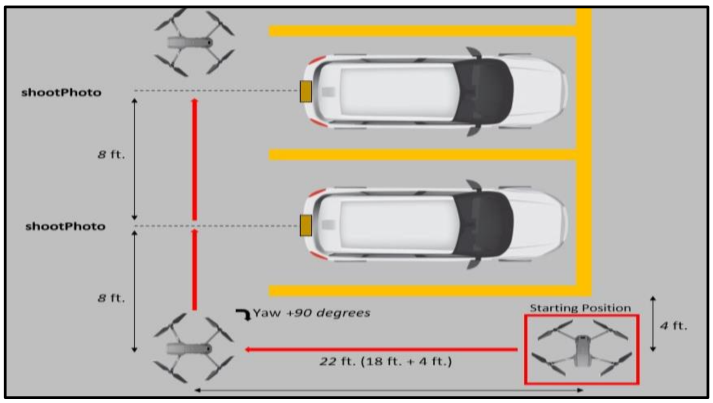

Figure 2. Autonomous Actions

The SDK contains API's (Application Program interface) that allow communication between two applications. The drone and the remote controller can be seen as one entity in the system, and the sample application has abilities to send and receive information from the drone and remote controller. The DJI Developer website contains ample documentation involving the structure of the API's under the categories of the flight controller, gimbal controller, and camera controller. Installing the sample application onto a mobile device can be done through cloning the GitHub repository of the DJI iOS Mobile SDK to a MacOS machine with XCode installed. This sample application is fully open-source and can be altered in any way the user desires. On startup of the application, the main menu searches for a connection to a compatible DJI drone. Once the connection is established, access is granted into the subsequent views of the application. The first menu that is seen is a list of the drone components such as camera, flight, battery, and gimbal components. Each component corresponds to a class within the Objective-C program, which customizable properties, functions, and methods can be declared and called at specified actions (i.e., a button press).

The desired application interface for this project is as simple as beginning with establishing a connection to the drone and selecting a cell within a main menu named "Parking Service" which leads to another application view. This view shall contain a button that triggers the drone to take off, fly along a predetermined path, and shoot photographs behind a small number of parked vehicles. Input parameters include the takeoff height, drone velocity, gimbal pitch (camera angle), and time delay between actions (i.e., The drone will take off, delay, then begin flying in a desired direction). The following schematic represents the parking lot configuration that will be considered when creating the autonomous flight program.

Figure 2 shows a diagram of the proposed autonomous actions that will be implemented into the application. Testing of this program will require the drone to be oriented as shown in the image with respect to the parking space orientation. The dimensions of the parking spaces are measured to be 18 feet by 8 feet, giving us insight on the distances the drone needs to fly. The drone shall takeoff at the position shown in the bottom right, aligned with the end of the parking space, at about 4 feet away from the line. The drone performs directional flight by rotating about the coordinate axis named the roll, pitch, and yaw axis as shown in Figure 3. The camera is facing parallel to the roll axis for reference.

ISBN: 97819384962-0-2

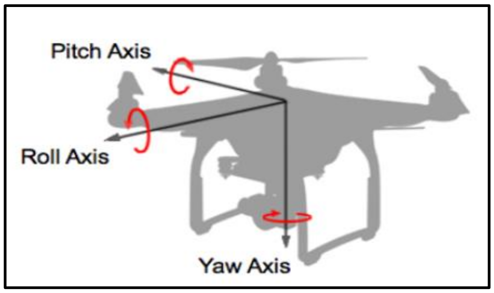

Figure 3. Rotational Movement Axes

Once take off is completed, the drone shall fly to the left by tilting in a negative-roll rotation for approximately 22 feet. Once the roll is completed, a positive 90-degree yaw rotation is initiated to properly orient the camera towards the parked vehicles. Using the body coordinate system within the drone's framework, the drone then repeats the same action where it moves in a leftward direction, defined as a negative roll rotation, traveling parallel to the line of parked cars. After an 8-foot interval, the drone shall be prompted to automatically shoot a photograph of the license plate. Since GPS distance measurement of short distances is unreliable, the photo interval will be expressed as a time interval. This time interval can be calculated by recording the velocity of the drone and the distance it will be travelling. Once photos are taken of two parked vehicles, the drone will be prompted to return to its home (takeoff) location.

4. Results

With testing flight parameters, different trial series were run for different purposes. The first trial was run to test the flight parameters of altitude and the distance from the edge of the parking spot. Measured in feet, the drone listed its own altitude through integrated sensors. The team then measured foot markers extending from the tip of the parking lines to show distance back from the spot. During this trial, it was found that the most ideal altitude and distance, based on the percentage correct and the correct number of characters detected, was at an altitude of 7 feet and a distance of 4 feet away from the parking line. From that parameter, the outputs had an 85.71% correct character detection and a 100% match in the number of characters detected. The total average number of correct characters read, excluding the readings where no license plate was detected, is 3.38 out of 7. While this is lower than desired, this was expected due to the lower quality images processed and public LPR being used. It was also found that the further the drone was from the parking line, there was a higher rate of no detection of licenses plates. This means that the LPR program incorrectly located the license plate area. For example, in an image with too much background noise, the code could perceive a series of trees as the plate characters. Other rectangular or similarly shaped areas in the surrounding environment can be wrongfully identified as a license plate. This is also true with respect to altitude; higher altitudes caused higher rates of no detection outputs. This issue is more attested to the warped vision of the license plate with height. One can envision a two-dimensional rectangle appearing more and more like a line as the height of perspective increases. In the second trial, specific coordinates were not recorded but rather, a free-form UAS flight was used to measure accuracy. It was found that higher qualities of images often resulted in better accuracy of license plate detection and correctness of character readings. In one of the readings, the accuracy of character detection was 100% and overall, the higher image qualities had an average of 4 out of 7 correctly read characters. In the last trial, the highest quality of images from a free form UAS flight were collected. The images were unable to be processed through the LPR code as the image files were too large. As mentioned, this LPR code is not as advanced as the one in action under TAPS. This brings on future issues that will be addressed with the optimal image size and quality based on speed of image processing and accuracy of character detection.

The completed state of the autonomous flight system has not been reached yet due to the requirement of understanding application development, namely in Objective-C. Understanding the dependencies between the different classes and files within the program poses as a complicated task with minimal programming background. However, the editable sample application has been successfully installed onto a mobile device, and the connection to the drone has been shown to initiate properly. Creating new views, customizable input parameters, and autonomous actions are tasks where there is not enough progress to begin testing in the parking lots. The main focus and difficulty of completing this project is debugging existing code, outsourcing for programs on sites such as GitHub and StackOverflow, and researching proper syntax and code structure in Objective-C. Overall operational success of the system is the main objective, which brings minimal sources of data collection and analytics at this stage.

5. Conclusions

The project aims to develop the application of drones, in conjunction with LPR technology, in parking enforcement. The team is currently focused on improving license plate image quality and reading accuracy, LPR code efficacy, and optimizing the drone's autonomous flight capabilities in order to form a system that will act as a rudimentary basis for a larger scale implementation. The product's end goal is not to replace the current parking enforcement system, but rather work concurrently with and enhance it. This endeavor can be justified not only by its applications within the TAPS department, but also by its transferable utility into other projects that currently require manual operation, but could benefit from the assistance of an autonomous, programmable drone. The impact of this project's success will inform future utilization of drones within the TAPS department. The advantages of implementation include reducing the costs of TAPS operations and environmental friendliness, as it would remove the need for circulating SUVs. By developing the functionality of drone-based parking enforcement, this project sets fundamental standards for future success.

6. Future Work

The team will continue to work towards the final deliverables as per the Gantt chart for the remainder of the spring semester. As mentioned, these include an autonomously flying drone, a functioning LPR program, a referenceable database, and a notification system of unregistered vehicles. Many of the deliverables and steps can be tested independently which allows for flexibility. The various flight modes will be explored and utilized. Flight speed and automated actions will be established and expanded. The LPR will be given more images and plates and will provide additional data points and information for calculations.

The project was intended to be passed along to future senior capstone groups. As the first group in the line of succession, this team is responsible for the initial tests and standards settings. The hope is that this project can eventually be expanded to include a fleet of synchronized drones surveying the campus using the official LPR program. With a fleet of drones, parking enforcement will be futuristic and streamlined. There will be measurable improvements in resource use, cost, and allocation. It will also communicate with the full license plate registry housed off site by a third-party database. Some aspects of this eventual system will ideally include drone battery life regulations and monitoring, flight shifts and mobile charging bases, as well as strictly virtually issued citations, that includes referencing DMV records for unknown vehicle plates. This system will also have attributes unforeseen at this point, but that will be developed by future teams.

The current scope of this project is developmental, and its purpose is to initiate the ability for a greater advancement and evolution. Acting as a steppingstone and accomplishing successful functionality, this project will pave the way for absolute implementation within the university's operations.

7. References

Gupta, G., Dua, A., Jain, S., (2019, March 2). OpenCV License Plate Recognition System. Retrieved from https://github.com/Garrydagr8/OPENCV-LICENSE-PLATE-RECOGNITION-SYSTEM

DJI Developer, (2018). Mobile-SDK-iOS. Retrieved from https://github.com/dji-sdk/Mobile-SDK-iOS

DJI Developer, (2019). Windows-SDK. Retrieved from https://github.com/dji-sdk/Windows-SDK

DJI Developer, (2020). Mobile SDK Documentation. Retrieved from https://developer.dji.com/mobile-sdk/documentation/introduction/index.html

ISBN: 97819384962-0-2