Crimson Publishers
Wings to the Research

Mini Review

Uav Surveillance for Urban Crime Prevention

Pushpa B[1], Vignesh P[1]*, Logesh V[2], Om Prakash S[2] and Sham M[2]

[1]Assistant Professor, Kings Engineering College, India

[2]Student, Kings Engineering College, India

ISSN: 2832-4463

***Corresponding author:** Vignesh P, Assistant Professor, Kings Engineering College, Chennai, India

Submission: June 15, 2022
Published: October 14, 2022

Volume 2 - Issue 3

How to cite this article: Pushpa B, Vignesh P*, Logesh V, Om Prakash S and Sham M. Uav Surveillance for Urban Crime Prevention. COJ Rob Artificial Intel. 2(3). COJRA. 000539. 2022.
DOI: 10.31031/COJRA.2022.02.000539

Abstract

The usage of UAV-based monitoring and surveillance has been rapidly increasing in the past few years. The UAV surveillance is quite efficient compared to CCTV cameras or regular patrol by security personnel as the flight reduces the time of the patrol and at the same time covers more area thus increasing not only the number of patrols but also increasing the radius of the patrol. Over the years UAV surveillance has been adopted by many nations but it these mainly used only for military purposes or in disaster spots if we use it in urban cities, we can cover the back alleys that are not under the CCTV coverage and the areas that are hard use CCTV cameras and patrol or even in private properties like mansions or companies that has huge gardens, etc., which greatly reduce many criminal activities also prevents crime such as burglaries, trespassing private properties, etc., it does not only help for police but also can be used in private properties. In this model, we propose a model system for urban UAV surveillance which will be able to patrol the designated area in the designated path and also detects person and weapons such as knife or gun.

Keywords: Deep learning; UAV; Automated surveillance; MAVLink; Open source

Introduction

According to the statics, the worldwide Unmanned Aerial Vehicle (UAV) or drone market is expected to reach 41.3 billion US dollars by the year 2026 [1], this indicates the growth and usage of drones in various fields, these drones changed our lives significantly over the past few years. But still, we are not using these drones for surveillance. The crime rates that happen in the back alleys are still higher than the other places in the urban cities because most of these parts are not under CCTV surveillance also it can patrol the areas which are expected to have more crime rates or private properties to prevent trespassers. It can also increase women's safety at night who work late. GCS is open-source software that can be downloaded from the internet and used for mission management and real-time flight operation. There are eight open-source projects for UAVs, we discussed five autopilots such as processors, gyroscopes, accelerometers, magnetometers and barometers. Introduced flight controllers with various functions to monitor and control UAVs. Their additional functions can be added by extending the open-source code. Mission planners and Q-ground control are known as open-source software packages that deliver UAV flight automation.

It enables UAVs to fly along with advanced predetermined Waypoints and organize real-time flight data. It has the function to return whether the battery is low. The functionality of GCS is convenient due to its wide range. There are some functions such as surveillance and object detection such as people, knives, and guns are neither available. There is too much attention on flight functions and the difficulty of making UAVs for Non-traditional tasks convenient for normal users. The making of customized GCS is a project requirement as mentioned. In real-time, the multiple UAVs are monitored by the development of an efficient and comprehensive guidance system, and the main layout is also discussed. Customized GCS are developed using an open-source library. Here in this research, unmanned helicopters with a two-layer framework that which data are transferred in the background are developed in the real-

time software system. The customized GCS surveillance system is inspired by light radar. The real-time tracking of commercial flights by providing geographical locations and flight data. To receive the flight data and command of the UAV use the flight numbers, speeds and MAV link protocol.

The proposed surveillance system is (1) the monitoring system that provides flight data. (2) Autonomous flight controlling, (3) real-time monitoring, and (4) area monitoring, (5) detection of weapons or trespassers, (6) patrolling at regular intervals. In this system, the #DR telemetry (TX/RX) is employed to attach the monitoring system, and also the Pixhawk supports PX4 firmware and ardupilotmega firmware. The UAV ID in MAVLink is used for the specific system IDs assigned for each autopilot. MAV Link messages and data carried by the telemetry, actual system ID of the XML record contain the data of the flight data and UAV GPS information. At the same time, the UAV interaction in functions like take-offs, landings, and auxiliary servo moments are worked by the commands of the monitoring system. The XML file is also used to send commands to the UAV. The MAV Link C header library is used to generate the XML files. JavaScript package manager provides NodeJS MAV Link to use decoding or encoding. TX and RX are two telemetric and send telemetry messages by (Serial COM6) and receive messages by (Series COM5).

Related Works

Proposed system

The surveillance system we propose here can be separated into two sections (1) Algorithm for object detection, and (2) flight planning of the UAV. The object detection algorithm implemented in the system constantly monitors the area under the surveillance for the trespassers in case it used on private property and weapons such as guns or knives in the ordinary street surveillance it also sent an emergency signal in case it been attacked or damaged to the police or the person in charge. In path planning, the UAV is assigned the path it needs to follow and it also gives it the instruction to take on or take off. Using this path planning the drone follows the path and returns to its station once the patrol is completed.

Path Planning

In this system, we use 3DR telemetry (TX/RX) to attach the Pixhawk autopilot UAV and also the monitoring system. The Pixhawk supports PX4 firmware and also the Ardupilotmega in our system [2-10]. We are using Ardupilotmega in this system. Each autopilot is assigned a specific system ID and they are assumed because of the UAV ID in MAV Link. The messages and data are carried by telemetry in MAV Link, this data contains the UAV GPS information and flight data of the actual system ID of the XML record. In the meantime, the monitoring system sends a command like take-offs, landings, and auxiliary servo movements to the UAV to have interactive functions. The commands are also sent to the UAV through the command XML file. XML files are created by the MAV Link C header library, which can be decoded or encoded using Node.js MAV Link of the JavaScript package manager. Here we are using two telemetries-TTX and RX. One telemetry (serial COM6) is employed to send while the opposite (serial COM5) continues to receive telemetry messages.

Tables 1 & 2 refer to the MAV Link messages and commands that are used to execute the monitoring system. Table 3 & 4 shows us the report status messages (failed or successfully executed) that appear after the commands are sent (Figure 1).

Table 1: MAV Link Message.

	Description/Purpose/Message ID	Field Name and Remark
1	-Geographical location -UAV tacking GPS_RAW_INT[10]	Lat (latitude)
		Long(Longitude)
		Alt(Altitude)
2	-Satellite visibility -UAV communication GPS_RAW_INT[10]	Satillites_visibility (Number of satellites visible(0-20)-small visibility numbers indicate that communication between the UAV and CGS is weak) Fix type (status of GPS-No GPS, 3D GPS and 2D GPS)
3	-Current waypoint -UAV tracking MISSION_CURRENT[10]	Seq (Number of sequence/waypoint to which the UAV is currently heading. We define a sequence as a waypoint before actual flight)
4	-Battery monitoring data -UAV flight safety SYS_STwaypoints	battery_voltage (currently used battery voltage in real-time of flight) battery_current (remaining current in real-time of flight) battery_remainig (remaining percentage in real-time of flight)
5	-Flight data -UAV flight safety ATTITUDE[10]	Pitch (pitch movement in real time of flight) Yaw (yaw movement in real time of flight) Roll (roll movement in real time of flight)
6	-Speed data -UAV flight safety GPS_RAW_INT[10]	Vel (Ground speed- provided by GPS embedded on autopilot)

7	-Heading data -UAV flight safety GLOBAL_POSITION_INT GPS_RAW-INT[10]	High (UAV heading yaw angle in degree) Cog (Course over ground-movement between two waypoints concerning the surface of the Earth)
8	-Flight data -UAV flight mode HEATHBEAT[10]	Base_mode Custom_mode (The combination of base_mode and custom mode can change the flight mode detail in Table 4)

Table 2: MAV Link Command.

	Description/Purpose/Command ID	Remark and Parameters
1	Defining waypoints for a mission/flight. To define waypoint for navigation MISSON_COUNT MISSION_REQUEST MISSION_ITEM MAV_CMD_NAV_WAYPOINT [10]	To define waypoints into autopilot, a set of command are used. Sample waypoint assigning with MAVLink via Nodejs javascript library can be seen in fig

2	Mission clear To clear all waypoint of the autopilot MISSION_CLEAR_ALL [10]	"MISSION_CLEAR_ALL"
		{
		'target_system":1'
		'target_component':1,
		'mission_type':0
		}

3	Mode control For UAV flight mode SET_MODE [10]	"SET_MODE"
		{
		'target_system':1
		'base_mode':89'
		Custom_mode':3
		}

4	Take off/ Landing To take-off and land and COMMAND_LON MAV_CMD_NAV_TAKEOFF MAV_CMD_NAV_LAND [10]	"COMMAND_LONG"
		{
		'target_system':1'
		'target_component':1'
		'confirmation':0'
		'param1':0,
		'param2':0,
		'param3':0,
		'param4':2,
		'param5':37.600785,
		'param6':126.864739'
		'param7':2,
		}

Table 3: Acknowledgment message.

	Description/Message ID	Field Name and Remark
1	Command acknowledgment [10] COMMAND_ACK	Result (MAV_RESULT) 0-accepted 4-fail This message is used after sending a command
3	Mission acknowledgment MISSION_ACK[10]	Type (MAV_MISSION_RESULT) 0-accepted 4-fail This message is used after sending a series of sequences/ waypoints for a mission

Table 4: Flight Mode.

Model Name	Base Mode+ Custom Mode		Field Name and Remark
Stabilize	81	0	UAVs can arm to fly only in the stable state
Alt Hold	89	3	UAV holds a specific altitude to drop the parcel from the air
RTL	89	6	Return to the start flying position after dropping the parcel
Auto	81	9	UAV changes into auto mode to navigate the predefined waypoint

```
m.createMessage(
"MISSION_COUNT",
{
'target_system' : 1,
'target_component' : 1,
'count' : 2,
// For 2 waypoints defining
'mission_type' : 0
},

function(message) {port.write(message.buffer) ; } ) ;

m.on( 'MISSION_REQUEST', function(message , fields) {
console.log ("seq is " + fields.seq);
} ) ;
m.createMessage(
"MISSION_ITEM",
{
        'target_system' : 1, //UAV ID
        'target_component': 1, //autopilot
        'seq' : 0, // Waypoint 1
        'frame' : 3, // default value for quadrotor
        'command': 16, //default value, MAV_CMD_NAV_WAYPOINT
        'current' : 0, // 0: before mission /// 1: during mission
        'autocontinue' : 1, //default value
        'param1' : 0,
        'param2' : 0,
        'param3' : 0,
        'param4' : 0,
        'x' : 37.601057, //waypoint 1 latitude
        'y' : 126.864401, //waypoint 1 longitude
        'z' : 1, // altitude (meter)
        'mission_type' : 0 //default
},

 function(message) {
  port.write(message.buffer);
 });
m.createMessage(
"MISSION_ITEM",
{
        'target_system' : 1, //UAV ID
        'target_component' : 1, //autopilot
        'seq' : 1, //waypoint 2
        'frame' : 3, // default value for quadrotor
        'command' : 16, //default value, MAV_CMD_NAV_WAYPOINT
        'current' : 0, // 0: before mission /// 1: during mission
        'autocontinue' : 1, //default value
        'param1' : 0,
        'param2' : 0,
        'param3' : 0,
        'param4' : 0,
        'x' : 37.600785, //waypoint 2 latitude
        'y' : 126.864739, //waypoint 2 longitude
        'z' : 2.68, // altitude (meter)
        'mission_type' : 0 //default
        },
function(message)
 { port.write(message.buffer);});

m.on ('MISSION_ACK', function(message, fields) {
        console.log("MISSION_ACK  is " + fields.type);
} ) ;
```

Figure 1: Assigning two waypoints with Node.JS MAV Link.

Weapon Detection

The weapon detection algorithm was designed using the application BagOffeatures and the category Classifier system. The analysis of the weapons is made into a combined input and fed into the system This phase receives the raw images from the drone, extracts the features, trains, and tests the system with the data. fist need to image pre-processing. The image set used in this research contains three sets of images: Airplanes, Guns, and Knives The different sets contain different amounts of images the image sets contain images whose object of interest seems to have a conflict with the background object The obtained results then, can be compared favorably with those of electromagnetic wave images. the electromagnetic waves when passed through a closed space, it acts as a MOM (Method of Moment), whose functions are unable to analyze metallic objects. Feature extraction. The next phase of image processing which is the taking out of peculiar features of the various sets is done using the BagOfFeatures function. Matrix laboratory (MATLAB) 2020 functions are used to flawlessly extract the features of the different groups. it identifies some identical features of the group which it considers strong features [11-17].

The strongest features consist of those characteristics that distinguish one category from the other. r it selects 80% of the strongest features, and this increases its sensitiveness when objects in such categories are tested with the system. The 80% strongest features extracted also help to overcome the challenges that may arise from blurry images from drone cameras. BagOfFeatures goes further to harmonize the number of these strongest features amongst the three groups. For the proposed system, the least number of strongest features have been gathered from the gun's category, with 12,446 strongest features. Hence the 12,446 strongest features of all three image sets are chosen and saved in the BagOfFeatures functions. K-means clustering is then implemented to create a vocabulary consisting of 100 visual words. This is a method of vector quantization. For the proposed system, a total number of 37,338 features were applied in this stage, and 100 clusters were created. The iteration in the stage was covered after the 14 iterations, at a 0.08 s/iteration rate. For each pixel of any image, it computes the Euclidean distance (d), between the center of the image and each pixel of an image using the function in the equation below. $d = p(x, y) - Ck$ (1) where, k is the number of clusters, x and y are the dimensions of the resolution of the image, and Ck represents clusters center the visual word occurrences (VWO) created for the three different groups. The guns category has a fair amount of VWO, due to its range of unique features.

Training and Result

The image category classifier function is used to classify images. In this stage, the output of the BagOfFeatures becomes an input. The function then trains a Support Vector Machine classifier (SVM) using the BagOfFeatures input. The parallel computing capabilities of this function make it possible for the system to train with as many features as are fed into the system.

$f(x) = \text{sgn } I \propto i\alpha jk(x, xi) + b$ Using $\alpha i \ldots, \alpha n$

where x is an observation (corresponding to a row of X). b is the bias term (corresponding to MDL. Bias). the trained system is evaluated using three sets of test categories. The test sets are a combination of images of different objects. The ability of the system to detect and correctly identify the number of each object in each of the sets displays the sensitivity of the system to any objects that may be introduced into the system. the results of single image test sets are easy to read. Hence we applied the same matrix order in introducing the test sets. From the final result, we see that 88 images were tested for each of the test sets and the confusion matrix shows a 95% accuracy for knives, 80% accuracy for guns, and 73% accuracy for airplanes (Tables 5 & 6). The resulting confusion matrix shows an average accuracy of 94.67% for the overall system. Hence, we can consider the system to be reliable to the nearest 0.01 error rate

Table 5: The results of the first evaluation and test.

Known	Predicted		
	Airplanes	Guns	Knives
Airplanes	0,73	0,16	0,11
Guns	0,8	0,8	0,13
Knives	0,01	0,03	0,95

Table 6: The results of the evaluation and test validation.

Dataset Partition	No. of Images for Training	No. of Images for Training	Average Accuracy
1:09	79	716	94%
2:08	159	636	94%
3:07	239	556	93%
4:06	318	477	93%
5:05	398	397	94%
6:04	477	318	94%
7:03	556	239	94%
8:02	636	159	93%
9:01	716	79	94%

Conclusion

The UAV is capable of monitoring and surveillance the urban streets without causing any major changes to the lifestyle of the people. Though this method is widely used during the crisis of COVID-19 to monitor the implementation of lockdown all are operated using operating personnel. But this method reduces the operating personnel as well as increases the safety of many citizens. An average commercial drone can fly for about 15-20 minutes when fully charged we use this time for patrolling and recharging once it reaches the station it can recharge itself and continues its patrol at regular interval. At the present study, our surveillance system can monitor around 300m of radius without any major issues

References

1. https://www.statista.com/statistics/1234521/worldwide-drone-market/
2. Lim H, Park J, Lee D, Kim (2012) Build your quadrotor: Open-source projects on unmanned aerial vehicles. IEEE Robotics & Automation Magazine 19(3): 33-45.
3. Khalid A, McFall K (2014) Aerial robotic autonomous patrol and surveillance system. 14th AIAA Aviation Technology, Integration, and Operations Conference 1-10.
4. Perez D, Maza I, Caballero F, Scarlatti D, Casado E, et al. (2013) A ground control station for a multi-uav surveillance system. J Intelligent Robot System 69: 119-130.
5. Dong M, Chen BM, Cai G, Peng K (2007) Development of a real-time onboard and ground station software system for a uav helicopter. Journal of Aerospace Computing, Information, and Communication 4: 933-955.
6. http://refhub.elsevier.com/S0045-7906(19)30393-3/sbref0006
7. https://www.npmjs.com/package/mavlink
8. http://mavlink.org/messages/common#MAV_CMD_ DO_SET_SERVO
9. http://mavlink.org/messages/common
10. http://qgroundcontrol.org/mavlink/waypoint_protocol
11. Giyenko A, Palvanov A, Cho Y (2018) Application of convolutional neural networks for visibility estimation of CCTV images. In: International Conference on Information Networking, Chiang Mai, Thailand, pp. 875-879.
12. Gupta BB, Quamara M (2018) An overview of internet of things (IoT): Architectural aspects, challenges, and protocols. Concurr Comput Pract Exp 32(21): 1-2.
13. Nur Ibrahim AW, Ching PW, Gerald Seet GL, Michael Lau WS, Czajewski W (2010) Moving objects detection and tracking framework for uav-based surveillance. 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, Singapore.
14. Ali S, Shah M (2006) Cocoa-tracking in aerial imagery. Airborne Intelligence Surveillance Reconnaissance (ISR) Systems and Applications III 6209: 62090D-62096D.
15. Harris C, Stephens M (1988) A combined corner and edge detector. Alvey Vision Conference 15: 147-151.
16. Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6): 381-395.
17. Bradski G (2000) The openCV library. Doctor Dobbs Journal 25: 120-126.