2021 10th International Conference on Advances in Computing and Communications (ICACC) | 978-1-6654-3919-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICACC-202152719.2021.9708269

YOLOv5 based Open-Source UAV for Human Detection during Search And Rescue (SAR)

Sruthi M S
Dept. of Electrical Engineering
Govt. Engineering College, Thrissur
Thrissur, India
sruthimangala2000@gmail.com

Manal Jaleel Poovathingal
Dept. of Electrical Engineering
Govt. Engineering College, Thrissur
Thrissur, India
mannujaleel@gmail.com

Nandana V N
Dept. of Electrical Engineering
Govt. Engineering College, Thrissur
Thrissur, India
nandanavn99@gmail.com

Lakshmi S
Dept. of Electrical Engineering
Govt. Engineering College, Thrissur
Thrissur, India
lakshmisoff@gmail.com

Mohamed Samshad
Research Assistant
ICFOSS
Thiruvanathapuram, India
samshadnc@icfoss.org

Sudeesh V S
Research Assistant
ICFOSS
Thiruvananthapuram, India
sudeeshvs@icfoss.org

***Abstract*—Floods and earthquakes are among the most frequently occurring natural disasters. They account for high mortality rates due to their rapidness and uncertainty of occurrence. Inundated lands require a quick response for rapid evacuation, arresting fatalities, and consequential economic losses. People tend to seek shelter at dry and open lands at times of calamity. The manual Search and Rescue (SAR) operations have their shortcomings due to the difficulties in identifying the human presence. It requires a longer time for evacuation and therefore increased mortalities. This paper proposes a quadcopter for real-time monitoring of isolated places and automatically detecting stranded humans during floods using image processing techniques at affordable rates. Live video streaming is possible with a camera and a video transmission system attached to the quadcopter. The rescue centers automatically receive the location of humans in case of human detection. Our model integrates an Open-Source autopilot system model, APM 2.8 multicopter flight controller that efficiently stabilizes the flight, and a YOLOv5 object tracking convolutional neural network algorithm for faster detection of human beings. The model is trained using a dedicated dataset of more than 1000 images and attains 0.954 mAP. We have developed a drone using open-source hardware and software tools, conducted test flights to check its stability and the efficiency of the object detection algorithm. We also conducted a mini-survey of one of the most flood-prone areas of Thrissur district in Kerala, using Mission Planner open-source software to evaluate how quickly our drone can assess the entire area. The aim is to save more human lives by quick and efficient aerial assessment in the most cost-efficient manner.**

***Index Terms*—Open-Source Drone, YOLOv5, Search And Rescue(SAR), Human detection**

I. Introduction

Aerial Surveillance at times of natural catastrophes like floods is gaining relevance in recent times due to the increased technological advancements in drone technology, computer vision, and image processing. One of the significant barriers for the rescue personnel is to accurately track the location of survivors and victims at the earliest [1]. Implementing Unmanned Aerial Vehicles for Situational Assessment and identifying the stranded people at isolated places is quick and promising in reducing the heavy mortality rates witnessed during floods due to the delay in evacuation [2], [3]. NASA's drone, Finder device with state-of-the-art sensors identified survivors under the rubble in the 2015 heavily intense 7.8-magnitude Nepal earthquake that caused widespread destruction [4]. Finder device was also deployed in Haiti Earthquake in 2010 for SAR [5]. In Uttarakhand flash floods in June 2013, four UAVs were deployed for SAR. UAVs identified around 170 stranded people from places where quick human intervention was not possible [6]. These scenarios reflect the necessity of aerial surveillance in SAR processes. The fast pace of technological growth demands us to switch to automation and replace the manual search for quick and efficient detection. The first 72 hours following the disaster are the most significant in carrying out SAR operations for saving human lives. High speed of response is necessary during this timeline [3]. In grave situations, outside community resources undertake manual SAR operations where the rescue personnel outside the flooded community reaches out only once they know the location of stranded people [7]. This process is time-consuming and error-prone at isolated places where communication facilities are weak. Also, human detection robots are likely to get vandalized while they search for humans under harsh conditions. Unlike the existing methods, flying machine like drone with a camera and video-transmission system can speedily cover large flooded areas by flying at low altitudes [1]. Thereby, it can quickly detect trapped humans by using an image processing algorithm and pass their location to rescue centers for a speedy response.

A brief working of our model is as follows: A camera mounted on the quadcopter captures a live video feed of the flooded region under inspection. The video is monitored in real-time in the ground control station, with the help of a video transmission system and a radio telemetry module. YOLOv5 object detection algorithm trained over a dedicated dataset of over 1000 images is used for human detection. It functions on the live video for human detection. The live coordinates of the quadcopter are also simultaneously obtained using a GPS module. Our database stores the video footage and GPS coordinates of stranded people for quick access and these locations are mailed to the rescue team. In addition, the location of detected humans is visualized on a map to make it easier for rescue personnel for future references. Open source autopilot system model, Ardupilot Mega (APM) 2.8 multi-copter flight controller is the main component that efficiently stabilizes and controls the flight. APM is one of the free, open-source, full-featured and reliable autopilot systems that can control any vehicle system from traditional helicopters to autonomous multi-copters [8] [9], which adds to the advantages of our model.

YOLO outperforms other real-time object detection methods like DPM, R-CNN and has more than twice the minimum average precision when compared to other real-time object detection methods. It also has a lesser number of background errors when compared to Fast R-CNN [10]. Existing models use algorithms like YOLO3-tiny based on the darknet framework for human detection [11]. Our system uses the latest version of the YOLO series, YOLOv5. YOLOv5 is built on the Py-Torch framework and makes it very user-friendly and faster to train datasets [12] [13]. Also, the False Positive Rates and False Negative Rates of the YOLOv5 algorithm are considerably lesser than the YOLOv3 algorithm in object detection making it ideal for fast and robust object detection [14]. We have used open-source hardware and software tools which makes our detection system more cost efficient.

The contributions of this paper is mentioned below:

1) Practical implementation of Open-Source Drone for SAR during natural calamities.
2) Quick and efficient real-time monitoring of the calamity-affected areas.
3) Detection of stranded human beings using the YOLOv5 object detection algorithm.
4) Sending the GPS coordinates of detected people by mail to the respective authorities for a speedy response.
5) Visualisation of the location of detected human beings on a map for quick and easy reference for the rescue organizations.
6) Mini-survey results of a flood-prone area of Kerala reflecting on the total time taken by the drone to assess the area.

II. Methodology

Fig. 1 shows the block diagram of the setup. The primary component of the system is a quadcopter which is used to carry out the missions. It is made to fly over the area where human detection is to be carried out. It carries an FPV (First Person View) camera from which we receive the live video feed required for live detection which is transmitted by the FPV

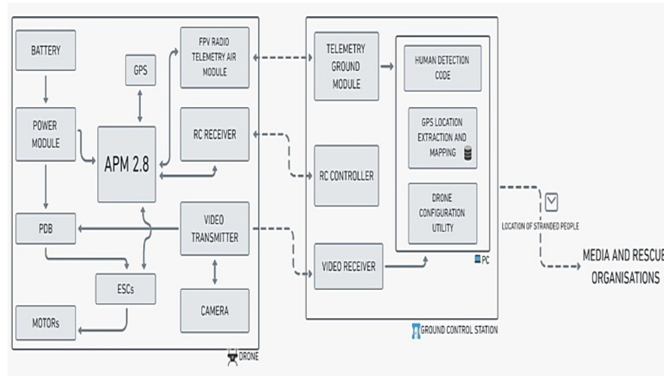


Fig. 1: Block diagram of the system

video transmitter. The open-source APM 2.8 flight controller controls the motors and interfaces the external and internal sensors [8]. A Li-Po battery is used to power the quadcopter and provide sufficient flight time. The external GPS module provides the live coordinates required to carry out the rescue operations.

In the Ground Control Station, image processing takes place and the software application Mission Planner monitors the quadcopter. To transfer data for drone monitoring, a 3DR Telemetry radio link is connected to the computational unit and the APM. The telemetry set allows one to observe the vehicle performance parameters and its current position. The operator in the ground station uses a 6 channel RC controller to manually control the quadcopter's manoeuvre, throttle and orientation (roll, yaw and pitch). It has been calibrated to fly in 3 modes– Stabilize, Loiter, and Alt Hold. The control commands from it are mapped into PWM signals which are provided to the flight controller to control the motors [8]. The output video signals from the video receiver is processed to detect stranded humans using real-time object detection.

After a connection between the quadcopter and the ground control station is established, the video feed is obtained. It is analysed and frames are extracted to carry out detection. OpenCV is used for processing the image for object detection. Images used to create the dataset have been trained using the YOLOv5 image detection algorithm. Each grid cell in the frame predicts bounding boxes with confidence scores and their parameters [13]. Hence people are detected and the coordinates of their location are stored as a database in the computational system. This database is mailed to the designated mail ID to carry out timely rescue missions. At the end of the program execution, a map of the location is generated where the detected human is pinpointed for better visualization.

III. Experimental Setup

A. System Hardware

The system uses a quadcopter as shown in Fig. 2 with a payload of 100g and a flight time of 10 minutes. The quadcopter uses an F450 frame that is built from glass fiber and polyamide nylon. The frame is designed to accommodate the motors and Electronic Speed Controller (ESC)s. The main component that functions as the brain of the quadcopter is the flight controller- APM 2.8 Multicopter Flight Controller. It is a complete open source autopilot system which has a variety of built-in sensors like, compass, 3-axis gyrometer, barometer which collect information about the height, direction and speed of the drone. It uses ATMEGA32U-2 and ATMEGA2560 chips for USB functions and processing respectively. The controller is used to vary the speed of the motors and monitor battery levels, control the GPS module, radio and video telemetries.

Fig. 2: Front view of the quadcopter

All the BLDC motors are connected to the flight controller via ESCs. Simonk 30A BLDC ESC is chosen for the drone to control the speed and direction in which the motor rotates. Given that each brushless motor requires an ESC, a quadcopter will require four ESCs. The ESC also includes protection on the 5V receiver line and backward polarity protection. This means that if you attach a battery backward by accident, it won't destroy the controller or affect the ESC.

A2212 - 1000kV BLDC out runner type motor with a thrust value of 800gwt at 3S is chosen for the drone with a 1045 propeller. The motor has superior speed vs torque characteristics, high efficiency with noiseless operation and a longer life. The propellers have a 15° angle design to avoid whirlpool multi-copter flying.

The telemetry chosen for establishing communication between quadcopter and the ground control station is 3DR 433 MHz 500 mW Radio Telemetry with a range of 2500m. This radio telemetry set has separate Air and Ground modules. Telemetry set allows communication of the flight controller with a USB or UART equipped device such as a tablet, computer or a laptop that supports a USB connection. The telemetry lets you see live information like GPS coordinates overlaid on a map, the system voltage and way point navigation.

The radio controller used to maneuver the drone is FlySky CT6B 2.4 GHz 6CH Transmitter that is accompanied by an FS-R6B Receiver with a bandwidth of 500 kHz. The transmitter and receiver pair is used to establish communication link between the controller and the quadcopter and it also has switches to control flight modes and directions.

For obtaining live coordinates, Neo 7M GPS module which includes HMC5883L digital compass is a low power, highly sensitive GPS module. We use Foxeer Razer Mini 1200 TVL CMOS $1/3''$ PAL / NTSC Switchable 2.1mm FPV Camera with 5.8GHz 48 Channels 600MW TS832 Transmitter (TX) Module and FPV Mini 5.8GHz 150 CH receiver for video transmission and reception respectively to get the live video feed.

A detailed connection diagram of the drone hardware is presented in Fig. 3.

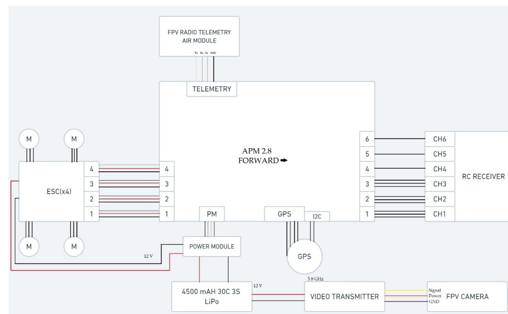


Fig. 3: Connection diagram of the quadcopter

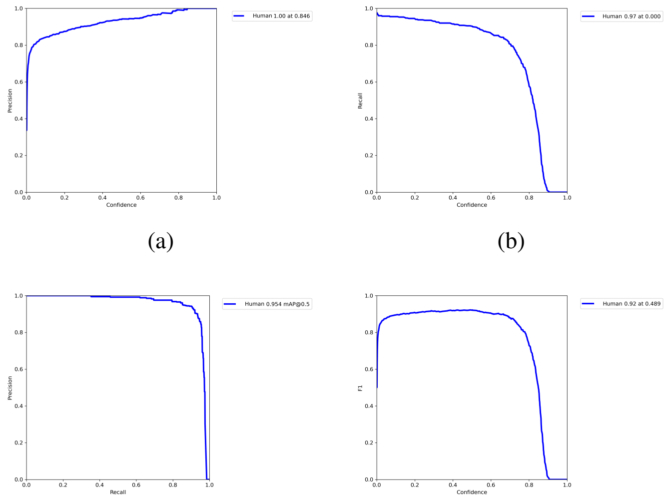
B. Dataset development

The custom dataset proposed for close range aerial human detection for our project contains 1028 distinctive images with 2794 human instances of different actions. The data was collected from various locales using a drone equipped with an analog FPV camera at a height of 8m. The footage provided videos with 640 x 480 pixel format at 60 fps. The dataset comprises of scenarios of a group of humans carrying out different actions at different angles and orientations to make it compatible for real time applications. The dataset initially recorded in the form of a video was converted to images by extracting frames using OpenCV and every 10 frames were skipped to avoid repetition. Image annotation was done using a browser based online software called Roboflow and was annotated for only one class titled "Human" as the main objective is human detection.

C. Human Detection

The Human detection and GPS data acquisition at the instant of identification is performed at the ground control station on a PC. With the help of a camera mounted on the quadcopter, a live video feed is obtained, that is used to detect stranded people using a human detection program that uses a model that is trained using YOLO v5 algorithm.

YOLO is a state-of-the-art, open-source real-time object detection system [10]. We have used the latest version of



(a) (b) (c) (d)

Fig. 4: Results of YOLO v5 training. a) precision vs confidence score graph. b) recall vs confidence score graph. c) precision -recall graph. It can be seen that the model has a mAP of 0.945. d) F1 score graph. The maximum F1 score is 0.92 at a confidence score of 0.489

YOLO algorithm to train our detection model [12]. This version has unique features employed like Mosaic augmentation, 16 bit floating point precision and adaptive anchor boxes which improve the model generalisability, inference speed, accuracy and overall robustness of the model. We used the version YOLOv5s in this project. The training details of the model used is given in Table I. The results of tr

TABLE I: TRAINING DETAILS

GPU	Tesla V100
Batch Size	1
Number of Epochs	300
Training Time	5.935 hours
GFLOPS	16.4

The setup establishes a connection with the drone, loads the trained model and utilizes the trained weights to detect humans and limit boxes for them. Then it extracts frames from the video feed obtained from the drone camera and proceeds to detecting humans. After detection, the predicted bounding boxes are drawn onto the image and the footage containing the detected results as shown in Fig. 5 is stored in a directory.

IV. Mission Planning

A. Speed of the UAV

The speed of the drone is a chief determinant in measuring flight time and detection accuracy. The maximum speed at which the drone can travel roughly depends on the type of the camera used for detection, the image processing time of the

Fig. 5: Snapshot of real time human detection

ground station and the percentage of overlap. The processing time t_p of the ground station depends upon the frame rate of the video feed as well as the image detection time. Overlap o is the common area covered by two successive photos of the same flight line in percentage as shown in Fig. 6c. For a smooth video, the overlap must be around $50-60\%$ [15]. The first step in determining the maximum possible speed is to find the ground footprint of the drone, i.e, to find the dimensions of the area captured by the camera in an instant. Since the camera is fixed at an angle to the vertical, the area captured by it would be trapezoidal.

Fig. 6a shows the FOV of the camera. θ_y is the vertical FOV of the camera, ϕ is the angle at which the camera is tilted from the normal to the ground, D_N and D_F are the nearest and furthest distances along the vertical FOV of the camera on the ground, h is the altitude of the drone. Fig. 6b shows the area covered by the camera. The edges of the trapezium would distort the image owing to the large horizontal FOV. Hence the area to be considered is a rectangle after removing the shaded parts as shown in Fig. 6b W_N and W_F are the widths of the trapezium due to the horizontal FOV θ_x of the camera. D_N and D_F can be given by [16];

$$D_N = h \tan\left(\phi - \frac{\theta_y}{2}\right) \tag{1}$$

$$D_F = h \tan\left(\phi + \frac{\theta_y}{2}\right) \tag{2}$$

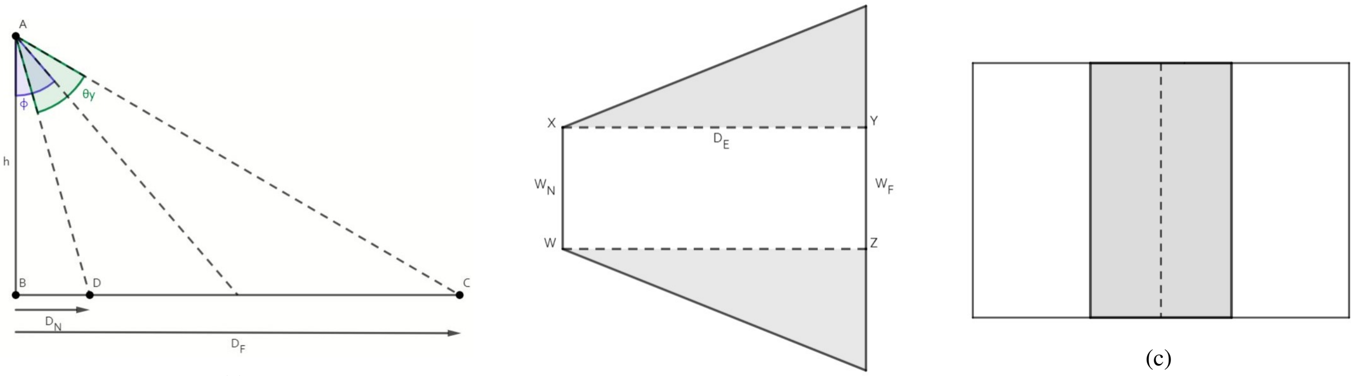
The height of the projected trapezoidal shape would be;

$$D_F - D_N = h\left(\tan\left(\phi + \frac{\theta_y}{2}\right) - \tan\left(\phi - \frac{\theta_y}{2}\right)\right) \tag{3}$$

The projected horizontal directions W_N and W_F can be given by [16];

$$W_N = 2\sqrt{{D_N}^2 + h^2}\tan\left(\frac{\theta_x}{2}\right) \tag{4}$$

$$W_F = 2\sqrt{{D_F}^2 + h^2}\tan\left(\frac{\theta_x}{2}\right) \tag{5}$$



(a)

(b)

(c)

Fig. 6: a) FOV of the camera on the ground. b) De-projected area of the FOV of the camera. The rectangle $XYZW$ shows the actual part of the area that can provide images without distortion. c) overlap between two successive images. The shaded area denotes the percentage of overlap.

The camera used for this project has a horizontal FOV $\theta_x = 125°$ and a vertical FOV of $\theta_y = 76°$. The camera is installed at an angle of $\phi = 45°$ to the vertical. Computing with these parameters gives the value of D_N, D_F, W_N and W_F as 0.982, 65.154, 30.966 and 252.199 meters respectively. The value of the height of the trapezoid, $D_F - D_N$ is 64.172m.

The processing time of the ground station t_p can be given as;

$$t_p = \frac{1}{fps} + t_d \tag{6}$$

Where fps is the number of Frames Per Second (FPS) and t_d is the time taken for detection. The system has a FPS of 60. The average detection time is 0.45 seconds. The processing time t_p was calculated to be 0.47 seconds.

Overlap only affects the length of the area. For the sake of simplicity, since the UAV is not autonomous, side lap, i.e, the common area covered by two adjacent flight lines is not considered for the study. The effective length of the area after considering overlap is given by;

$$D_E = \left(1 - \frac{o}{100}\right)(D_F - D_N) \tag{7}$$

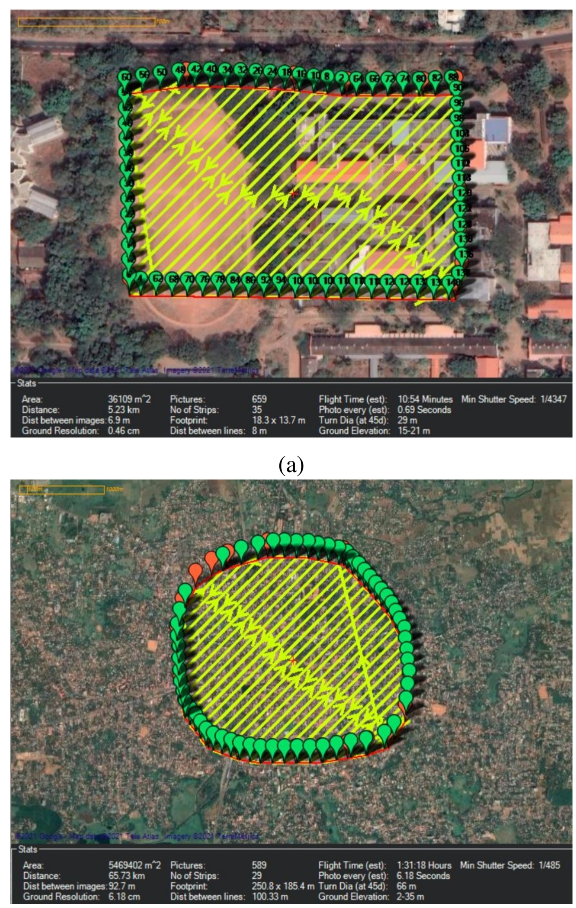
To equation for the maximum flight speed v_m of the drone can be given by;

$$v_m = \frac{D_E}{t_p} \tag{8}$$

On taking an overlap percentage of $o = 80$, the maximum speed possible is 27.31m/s. A speed of 10m/s was fixed for the project as a safe value.

B. Flight Time

Mission planner is a free and open-source software that helps to plan and analyse missions. The survey feature in the software can be used to calculate the flight time of the drone. A portion of the area of our college, Government Engineering College, Thrissur, was considered for the survey (Fig. 7a). According to this, the current setup would take approximately



(a)

(b)

Fig. 7: a) ground area surveyed using Foxeer Razer Mini at a height of 8m. The area covered is part of Govt. Engineering College Thrissur. b) Part of the town of Thrissur surveyed using the same setup but with a Go Pro Hero 4 camera at an altitude of 100m.

10 minutes and 54 seconds to fully cover an area of 8.92 acres when flying at an altitude of 8m and a speed of 10m/s. This was validated by actual flight which yielded similar results.

A survey was then done on a more extensive area covering a considerable portion of the town of Thrissur, which is a flood prone area. A better camera, GoPro Hero 4 Wide was used and the flight altitude was modified to 100m. A flight speed of 15m/s was set. The area covered was calculated to be 1351 acres with a flight time of 1 hour and 31 minutes (Fig. 7b). The area can be divided into smaller parts with flight times of $10-15$ minutes each and as such would only take a few flights to cover a large area thus making SAR operations faster and more efficient.

V. Conclusion

This paper presented a drone-based approach for human detection for SAR operations during disasters. The YOLOv5 algorithm was used on a custom dataset created from footage taken by an on-board camera on the drone. The system can be used for real-time applications. A live video feed of the disaster area is obtained via radio telemetry. The location of detected human beings are shown on a digital map and emails are sent to the respective authorities for quick action. The project uses open-source hardware that makes it economical and easy to build and reproduce.

For future improvements, a better camera, such as a go-pro, can be added to widen the search area by flying the drone at higher altitudes. Such a setup will also aid in creating better datasets which will further improve SAR operations. The drone can also be made fully autonomous for faster and more accurate detection.

Acknowledgement

We would like to thank International Centre for Free and Open Source Software (ICFOSS), Trivandrum for providing laboratory facilities and guidance. We would also like to extend our sincere gratitude towards Mr. Srinivasan Ravindran, Programme Head, ICFOSS, Mr. Mohamed Samshad and Mr Sudeesh, research assistants and project guides at ICFOSS.

References

[1] R. Tariq, M. Rahim, N. Aslam, N. Bawany, and U. Faseeha, “Dronaid : A smart human detection drone for rescue,” in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT IoT (HONET-ICT)*, 2018, pp. 33–37.

[2] Z. Zaheer, A. Usmani, E. Khan, and M. A. Qadeer, “Aerial surveillance system using uav,” in *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, 2016, pp. 1–7.

[3] A. Valsan, P. B., V. D. G. H., R. Unnikrishnan, P. K. Reddy, and V. A., “Unmanned aerial vehicle for search and rescue mission,” in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 684–687.

[4] “Finder search and rescue technology helped save lives in nepal,” May 2015. [Online]. Available: https://www.nasa.gov/jpl/finder-search-and-rescue-technology-helped-save-lives-in-nepal (Accessed: May. 29, 2021).

[5] “Nepal earthquake: How drones, sensors came to people’s rescue,” May 2015. [Online]. Available: https://www.indiatoday.in/world/story/nepal-earthquake-drones-sensors-nasa-team-rubicon-halodrop-252008-2015-05-08 (Accessed May. 24, 2021).

[6] C. Chauhan, “Uttarakhand: Drones in rescue act,” June 2013. [Online]. Available: https://www.hindustantimes.com/india/uttarakhand-drones-in-rescue-act/story-aoVZX6cd76javXFpm9COtO.html (Accessed May. 27, 2021).

[7] “Search and rescue manual.” [Online]. Available: http://www.sdmassam.nic.in/download/searchandrescuemanual.pdf (Accessed: May. 29, 2021).

[8] E. Ebeid, M. Skriver, and J. Jin, “A survey on open-source flight control platforms of unmanned aerial vehicle,” *2017 Euromicro Conference on Digital System Design (DSD)*, pp. 396–402, 2017.

[9] “Ardupilot mega.” [Online]. Available: https://www.ardupilot.co.uk/ (Accessed: May. 25, 2021).

[10] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[11] C. Liu and T. Szirányi, “Real-time human detection and gesture recognition for on-board uav rescue,” *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2180

[12] G. Jocher, A. Stoken, J. Borovec, NanoCode012, A. Chaurasia, TaoXie, L. Changyu, A. V, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, Marc, Y. Kwon, oleg, wanghaoyang0106, Y. Defretin, A. Lohia, ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, Doug, Durgesh, and F. Ingham, “ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations,” Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4679653

[13] D. Thuan, “Evolution of yolo algorithm and yolov5: The state-of-the-art object detection algorithm,” Bachelor’s Thesis, Information Technology, Oulu University of Applied Sciences, 2021. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/452552/Do_Thuan.pdf

[14] A. Kuznetsova, T. Maleva, and V. Soloviev, “Detecting apples in orchards using yolov3 and yolov5 in general and close-up images,” in *Advances in Neural Networks – ISNN 2020*. Cham: Springer International Publishing, 2020, pp. 233–243.

[15] F. Flores-de Santiago, L. Valderrama, F. Flores-Verdugo, and R. Rodriguez-Sobreyra, “Assessing the effect of flight altitude and overlap on orthoimage generation for uav estimates of coastal wetlands,” *Journal of Coastal Conservation*, vol. 24, p. 35, 05 2020.

[16] C. Burke, M. Rashman, S. Wich, A. Symons, C. Theron, and S. Longmore, “Optimizing observing strategies for monitoring animals using drone-mounted thermal infrared cameras,” *International Journal of Remote Sensing*, vol. 40, no. 2, pp. 439–467, 2019. [Online]. Available: https://doi.org/10.1080/01431161.2018.1558372