BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Project Report

Course No: EEE 318
Course Name: Control System I Laboratory

Submitted to

Dr. S. M. Mahbubur Rahman
Professor, Department of EEE, BUET

Tasnima Afsana
Adjunct Lecturer, Department of EEE, BUET

Submitted by

Group 1
EEE 17, Section A (2)
Level 3, Term 2

Members

Saleh Ahmed Khan – 1706053
Adib Md. Ridwan – 1706055
Imtiaz Ahmed – 1706062
Tamim Ahmed – 1706063
Sadat Tahmeed Azad - 1706064

Submission date:

27 February, 2022

Post Disaster Surveillance and Response Using Human Detection From Real Time Aerial Imagery

Saleh Ahmed Khan
Department of EEE
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
1706053@eee.buet.ac.bd

Tamim Ahmed
Department of EEE
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
1706063@eee.buet.ac.bd

Adib Md. Ridwan
Department of EEE
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
1706055@eee.buet.ac.bd

Sadat Tahmeed Azad
Department of EEE
Bangladesh University of Engineering and Technology
Dhaka,Bangladesh
1706064@eee.buet.ac.bd

Imtiaz Ahmed
Department of EEE
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
1706062@eee.buet.ac.bd

Abstract—Disaster management in Bangladesh has improved a lot in recent years with the advent of technological advances in early predictions and countermeasures to minimize damages. But unfortunately still we can't completely eradicate the loss due to these disasters and that's why post natural disaster management is one of the significant topics to reduce the loss in human lives. Aerial imagery using Drones can be used in this case to identify human locations to rescue, provide with medicine and rations and to manage crowds in emergency shelters. With that in mind, this document proposes a drone based real time human detection system which will enable identifying humans and disaster surveys of the affected area for better decision taking and management.

Index terms —Disaster Management, Aerial Imagery, Human detection system, disaster survey

Ⅰ. INTRODUCTION

Bangladesh is one of the most disaster-prone countries on the face of earth. Within over 200 natural disasters in the past 40 years due to geographic location, most of them were cyclones.[2] Currently, multipurpose cyclone shelter (MPCS) provides short term safety for cyclone affected people, but for those who can reach in time. Even those people are not fully safe from the effects of cyclones. Several methods are used to identify individuals to rescue and preserve lives in an efficient way, such as Wi-Fi scanner and smartphones to detect people.[2] Although this types of method are very good for human identification and rescuing, an aerial survey identifying situation in real life will boost the damage reduction and rescue process even further along with the existing framework as it will give a full overview of the situation to make educated guess for the authority.

Disaster risk reduction, preparedness, response, recovery, relief and rehabilitation. These are the building blocks of disaster management. Drones help reinforce and amplify the impact of each of these blocks by 3x, 10x or even 25x. Disaster response, in particular, has seen major strides and improvement with these Unmanned Aerial Vehicles (UAVs) or drones.[1] Although drone usage in disaster management is a fairly new concept, in research time various types of research and application works are implemented in this field. In US drone based survey system has proved to be useful in many cases, specially in for damage assessment purposes during the 2017 southeastern US Hurricanes Harvey and Irma.[4] Based on research articles identified from 2009 to 2020, drone applications in disasters are classified into four categories; (1) mapping or disaster management which has shown the highest contribution, (2) search and rescue, (3) transportation and (4) training. But still there is limited discussion to address the post-disaster healthcare situation especially with regards to disaster victim identification. So, human identification from post natural disaster footage to identify, rescue and rehabilitate victims is a very important aspect of addressing disaster management effectively.[3]

So, we propose a drone imagery based post natural disaster survey and rescue system that will help save life in the coastal regions of Bangladesh. Here, along with the existing system, our proposed system will give an extra edge to the authorities as the exact location and condition of victims can be identified. We'll control the drone from a headquarter, live footage will be sent from the camera via the help of satellite internet. In real time damage assessment will be possible through implemented Deep Learning algorithms.

Ⅱ. SYSTEM ARCHITECTURE

To implement our real life design scenario, currently we have implemented a real time aerial imagery based human detection algorithm that will help in real life detecting victims in the disaster affected areas. The system can be divided into hardware and software implementations.

A. Hardware

This section includes the hardware setup of our pipeline. Our system pipeline design consists of 5 modules: base station, radio communication module, sensors, PID controller and drone itself.

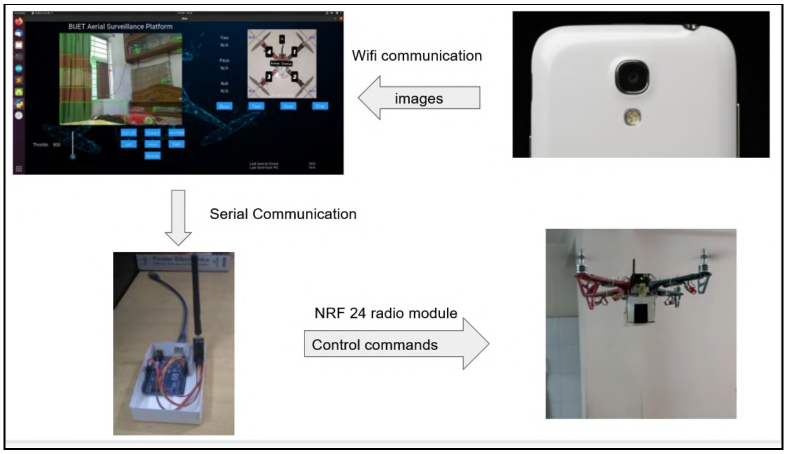


Fig: Complete system architecture

1. **Base Station:** The role of a base station in our architecture is to monitor the whole architecture, respond to any anomaly in the pipeline, monitor and extract insights from the incoming data. All the modules in this system will be connected using satellite internet and in real time video with inference will be sent to the base station. Identifying victim position and proper route can be done using this footage to conduct damage assessment. In our case we have used a laptop as the main base station. A transceiver radio communication transceiver module is used to connect the base station with the drone. As we have used a mobile phone as our camera, Droidcam software is used to livestream the feed using serial communication.

Fig: base station

2. **Radio Communication Module:** This module is used to connect the drone with the base station so that the direction, speed can be controlled and real time surveys can be conducted. It consists of 2 nrf24L01 transceiver modules, one at the base station and another

mounted on the drone. This works in 2.4 GHz worldwide ISM frequency bad and GFSK modulation for data transmission with an operating voltage of 1.9-3.6 V but can work with any 5V logic microcontroller like Arduino.[5] So, nrf24l01 module was chosen because of low cost, high stability and low power consumption and has a broad application prospects.

3. **Sensors :** We have used gyroscope and accelerometer sensors to give the PID controller the information needed to adjust its speed and direction. The accelerometer detects that it has been thrown in the air and is falling back to the ground. The gyroscope will then stabilize its orientation within a split second. Next, the distance sensor stabilizes the drone to a particular pre-programmed height from the ground. Then the drone locks to its current position.[6]

4. **PID controller:** PID controllers are often used for controlling a drone. This is a program in the drone, which evaluates the sensor data and drives the motors accordingly via the motor controller. PID stands for: proportional–integral–derivative. The characteristics of a good controller are:- stable impression in the air, minimal deflection by a gust of wind and long flight time and low power consumption during flight time. [7] We have used an arduino uno mounted on the drone with the code needed to act as the PID controller which runs code uploaded from our laptop depending on experimentations.

5. **UAV module:** All the modules that were mentioned to be mounted on with some other components will complete our flying part. The main parts are- the quadcopter frame, one BLDC motor for each arm, one ESC controller for each motor, one propeller for each arm, radio transceiver mounted on drone, Battery and power distribution cables, camera mounted for first person video capture.

Frame: Available DJI450 frame has been used for the body of the frame, to hold the motors/ the controller board and batteries.

Propellers: 8” and in some cases 10” propellers have been used to provide lifting force for the drone.

BLDC motors: 1400 kV motors have been used to power the propellers. Which generates enough lift to get our heavy drone in the air.

Electronic Speed Controllers: BLHeli 30A ESCs have been used to provide steady high current required to operate the motors.

Batteries: 2200mAh 3S lipo battery has been used to provide continuous power to both the ESCs and the Arduino MCU.

Radio Module: NRF24 Radio module provides continuous communication between base station and drone.
Gyroscope: GY 521 6 Degree of Freedom gyroscope and IMU is the main sensor of the drone and provides closed loop feedback for the drone to maintain its stability and control.

Flight Controller: An Arduino Uno serves as the brain of the drone. Which processes the PID algorithm and keeps the drone stable, as well as receives and interprets commands given by the user.

Drone Circuit Board: All necessary components are soldered and minted sturdily on the frame so that they don't get damaged while the drone is in operation.

Mounted Camera: A camera holder has also been mounted on the drone to provide live feed back to the base station FPV/Mobile camera will be used)

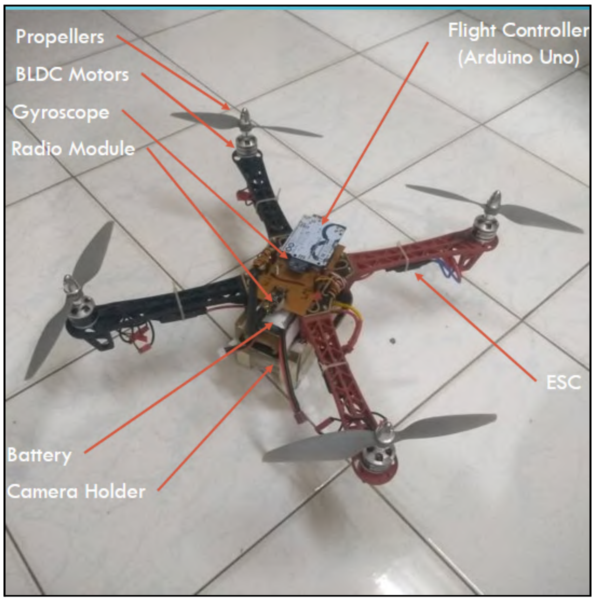


Fig: Drone with all necessary components

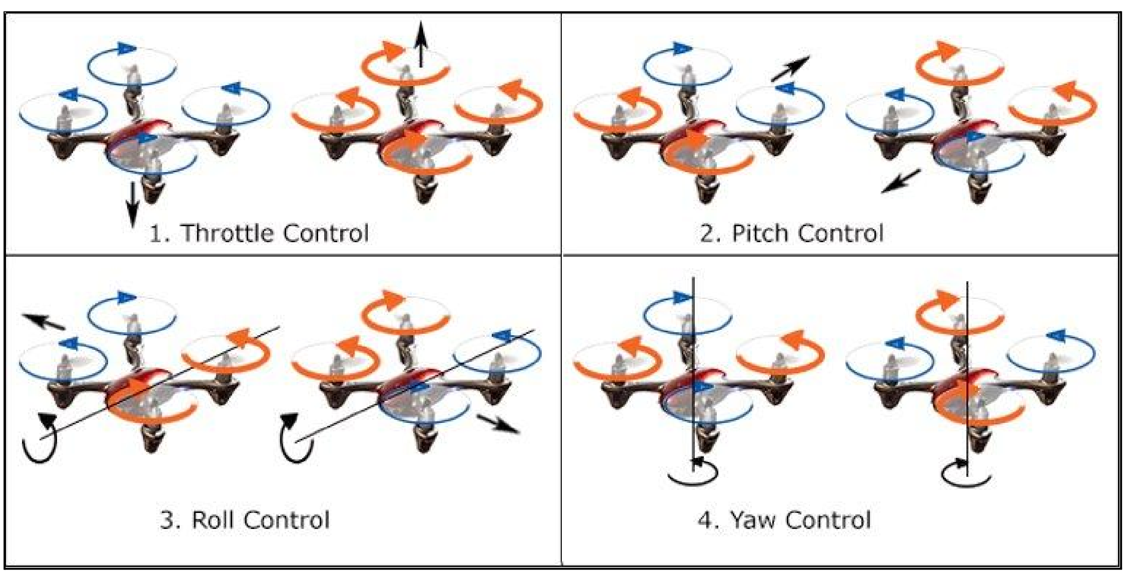
B. Software

The software section of this portion can be divided into mainly 3 modules, basic setup, PID controller, Human Detection algorithm. These 3 modules are implemented using arduino and python.

1. **Basic Setup :** We have used a **kivy(** python based framework) based GUI to control the throttle and observe the speed of each motor ,values from sensors and live feed from the camera. This GUI is connected with the radio transceiver module via arduino uno,which works as the controller, to transmit and receive signal from the drone. The communication uses a Python-ROS-Arduino framework. The python program streams the live video, and it is connected to a ROS node which provides instant communication to the base station arduino, which in turn, sends data to the drone. The python program has a keyboard function, so it can be controlled conveniently using the WASD buttons.

Fig: GUI(kivy based)

2. **PID controller :** The feedback control loop is implemented using this PID controller. An Arduino uno is used for this and so the controlling code is implemented using arduino code. In order to stabilize the drone mid air, 3 dimensions of stability are needed (Roll, Pitch and Yaw) ,each with their own kp, ki, kd values. The Gyro takes these angles as input to the PID controller, applies the PID algorithm and outputs the speed to the 4 motors on top of the base throttle specified by the base station. As seen in the diagram, to control pitch, the two forward motors are adjusted, in order to roll, the two side motors and finally for yaw, the two diagonal motors. The altitude of the drone is controlled using the throttle.



3. **Human detection:** For human detection various algorithms perform satisfactorily including YOLOv3-5, PedNet, multiped, SSD MobileNet V1, SSD MobileNet V2, SSD inception V2.[8] The deciding factor in these types of emergency response situations is **detection speed and accuracy**. Among these models **YOLO** models are comparatively **faster** and the latest **YOLOv5** algorithm is the state of the art in this field in terms of **optimized accuracy and latency**.(Inference speed of **140 FPS**, yolo v4 had 50 FPS with 90 percent reduction in size) Among different versions of YOLOv5 the **v5s** is one of the fastest and better performing. So, initially we started with the pretrained YOLOv5s version.

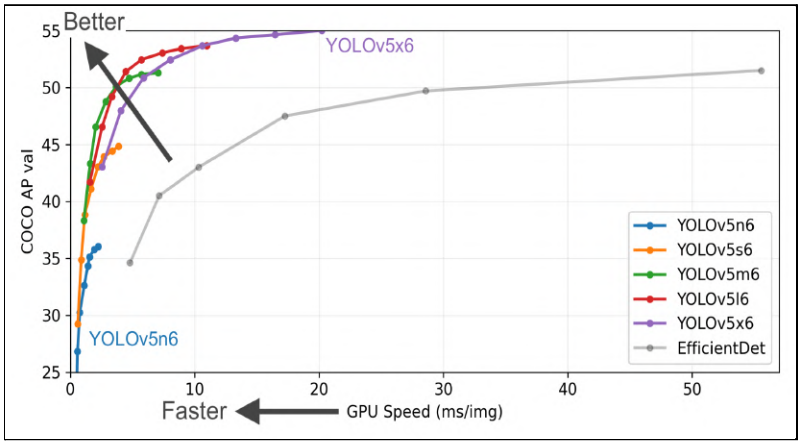


Fig: performance comparison among different yolo models[9]

YOLOv5 is mainly a CNN based architecture suitable for object detection tasks with the help of bounding boxes. It consists of three parts: (1) Backbone: CSPDarknet, (2) Neck: PANet, and (3) Head: Yolo Layer. The data are first input to CSPDarknet for feature extraction, and then fed to PANet for feature fusion. Finally, Yolo Layer outputs detection results (class, score, location, size). Although pretrained model works pretty good, in almost all scenarios custom training boosts overall performance

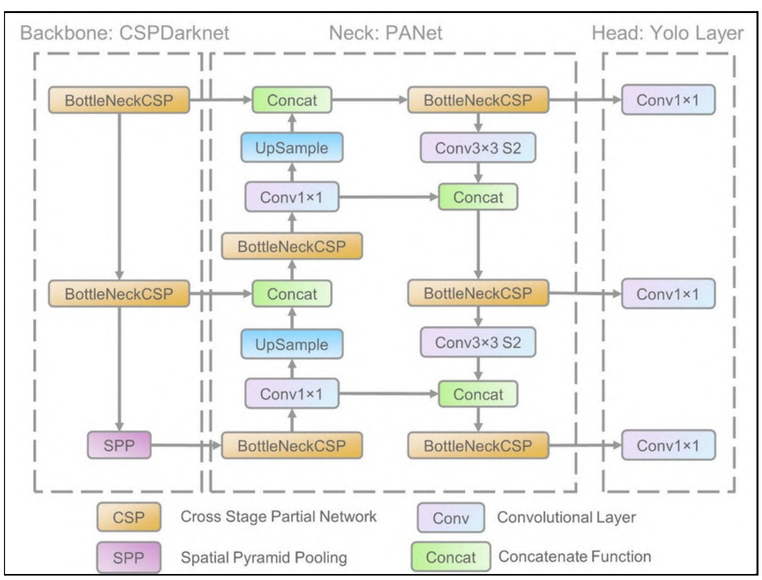


Fig: YOLOv5 architecture[13]

Ⅲ. METHOD

A. PID controlling method

PID tuning is done by holding the drone in one hand while the operator uses the controller and increases the throttle, afterwards, by tilting the drone to any one side, we can see whether or not the drone responds accordingly and tries to stabilize. Once the drone is sufficiently stable, it is placed on the ground and attempted to fly. The operator must control it from the base station at all times, otherwise, it is difficult for the drone to hover on its own completely.

B. Human Detection

At first, pre pre-trained YOLOv5s model was used, but it had a diminishing result with higher heights. In order to make our drone suitable to fly at different levels according to our task we wanted a more robust detection approach. So, to increase detection accuracy we custom trained the model. In our case, a post disaster dataset would have been the perfect fit. But due to scarcity of suitable datasets of that sort, we have compared among various datasets including CrowdHuman[10], VisDrone[11], Okutama-Action[12] to train our model. Among these the VisDrone dataset is chosen to train our human detection algorithm as it covers most of the cases with height difference and background variation which will be crucial in identifying humans from post natural disaster footage.

Dataset	Dataset Size	Strength	Weakness
Okutama-action	43 minute video with 12 action classes at 30 FPS and 77365 frames in 4K resolu- tion. It captured from **10-45 meters** and with cam- era angle of 45 or 90 degrees	dynamic transition of actions, significant changes in scale and aspect ratio, abrupt camera movement, as well as multi-labeled actors.	Clean background, doesn't represent disaster affected area that much
VisDrone	Train : val : test = 6471 : 548 : 1610 images consisting of 288 video clips formed by 261,908 frames and 10,209 static images ranging from 14 cities in China.	the dataset was collected using various drone platforms (i.e., drones with different models), in different scenarios, and under various weather and lighting conditions, which significantly realizes the messy background.	Less lower shots of images and dynamic transitions are not included.
CrowdHuman	Train: val: test respectively 15,000, 4,370 and 5,000 images.	The total number of persons is also noticeably larger than the others with ~340k	Only lower level and closeup images of humans which doesn't represent

		person and ~99k ignore region annotations in the CrowdHuman training subset.	disaster condition.

Fig: Dataset comparison

Considering all the factors, the **VisDrone(image) dataset** is used for custom training our model.

Dataset	Image Size	Batch size	epochs	Class number
Train:val:test = 6471:548:1610	640 x 640	16	10,30	10

Fig: Training Description

Hyperparameters	Learning rate initial(lr0)	Learning rate final(lrf)	Momentum	weight decay
Values	0.01	0.01	0.937	0.0005

Fig: Hyperparameters used

Fig: sample training images

Ⅳ. RESULTS

A. System Implementation

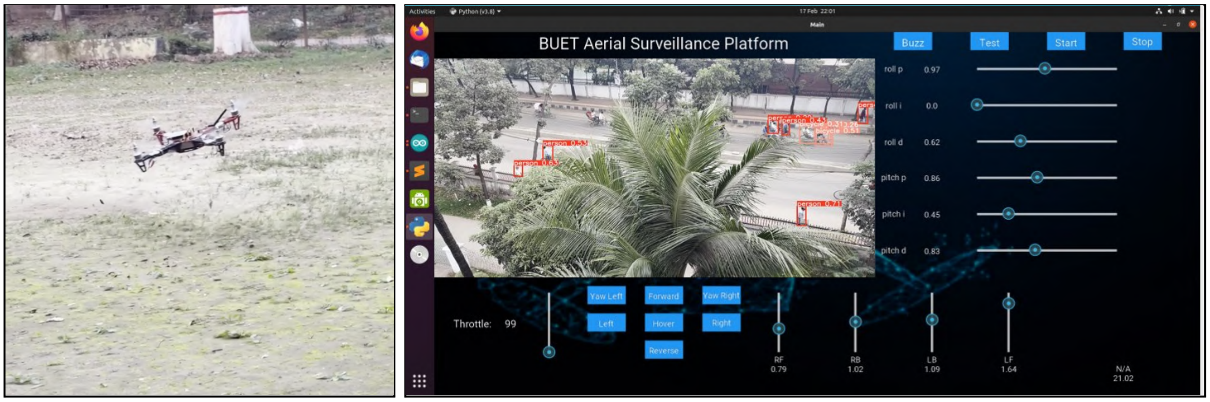


Fig: a. Drone during flight,b.detection from the feed.

B. Human Detection Model

From the detected images it can be seen that the model performs pretty well in close range detection even with the pretrained weights. But training for 10 epochs and 30 epochs has shown significant improvement in the performance.

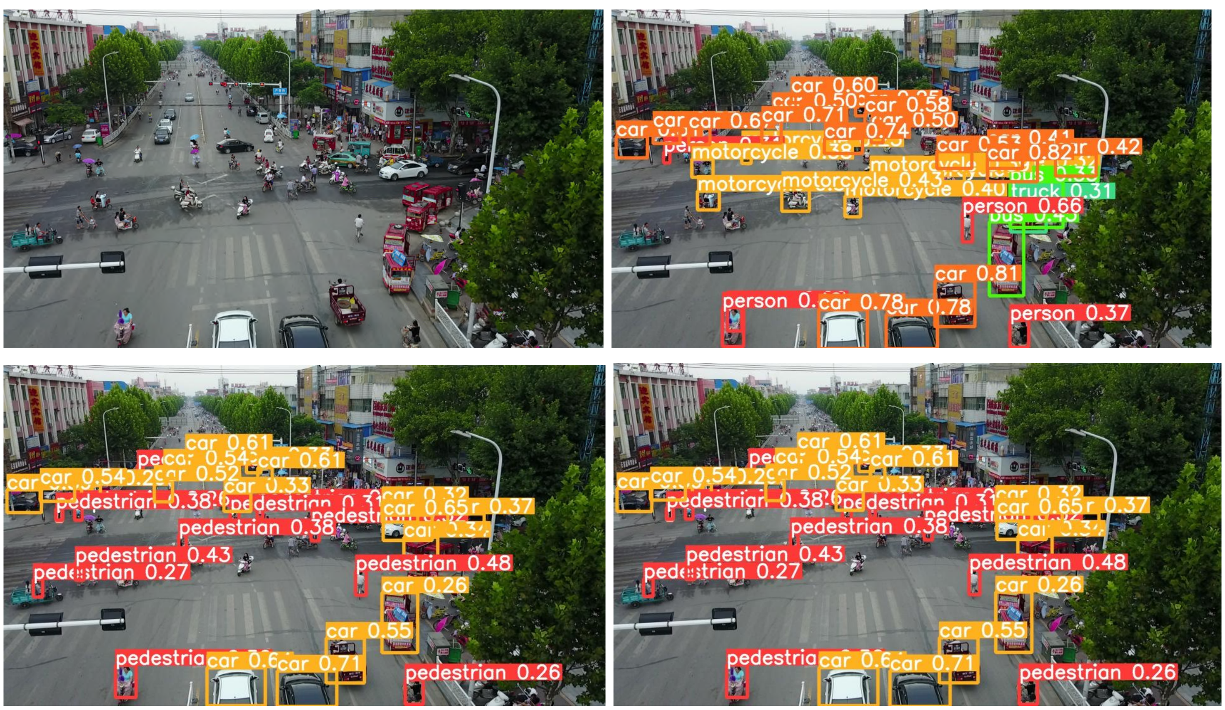


Fig: test image 1 a) given image,b,c,d) pretrained,10 epochs, 30 epochs

Fig: test image 2 a) given image,b,c,d) pretrained,10 epochs, 30 epochs

But, unfortunately for vertical/ images taken from very high altitude the accuracy tends to drop a lot, in those cases training for more epochs will prove to be useful as with more epochs we see improvement in performance.

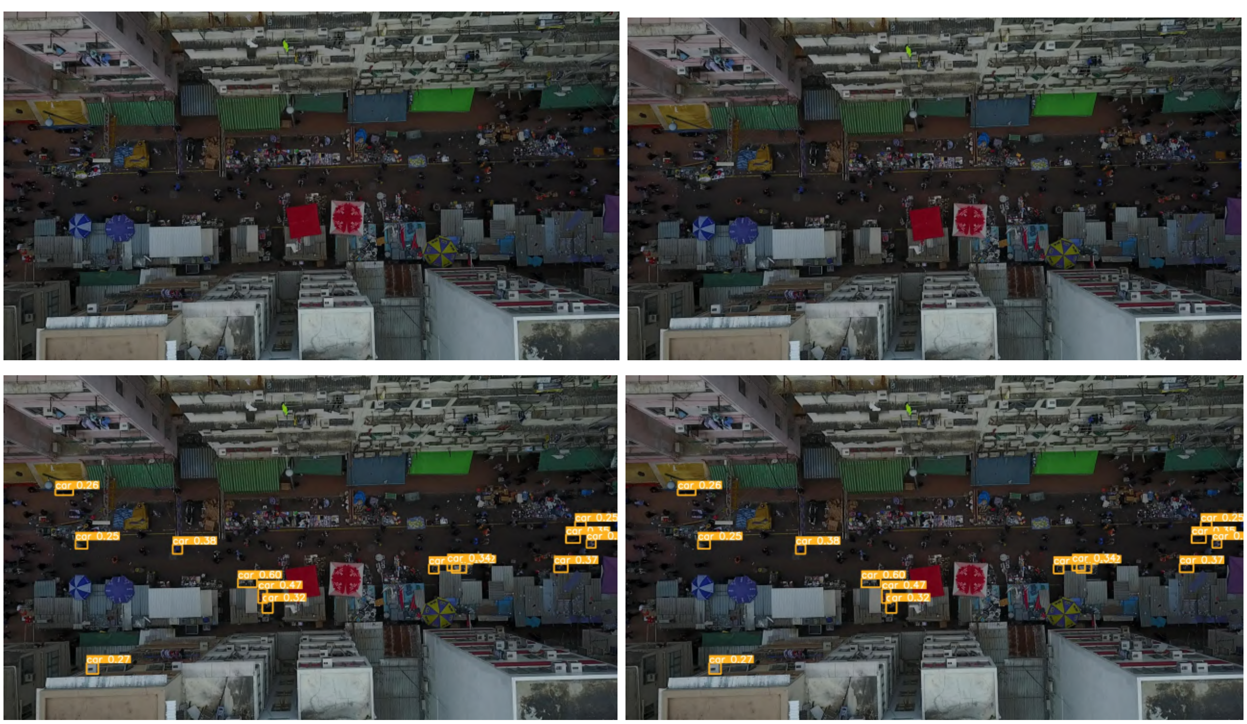


Fig: test image 3 a) given image,b,c,d) pretrained,10 epochs, 30 epochs

Moreover, Validation and runtime results of the yolov5 model run for 10 and 30 epochs are shown below. From here we can see precision, recall and Mean Average Precision, all increase with the number of epochs. Results are shown below:

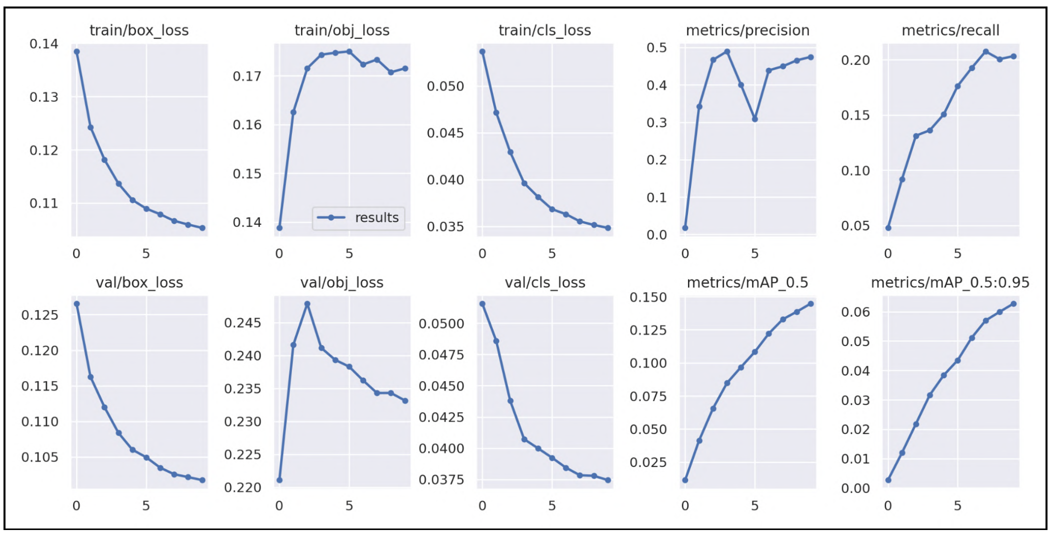
1. 10 epochs

Model Summary: 232 layers, 7270791 parameters, 0 gradients, 16.8 GFLOPs

val: Scanning '/content/drive/MyDrive/control_project/Control/VisDrone_Images/Visdrone_image

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100%
all	548	38759	0.464	0.202	0.139	0.0605
pedestrian	548	8844	0.139	0.383	0.21	0.0667
people	548	5125	0.271	0.261	0.166	0.0446
bicycle	548	1287	1	0	0.00998	0.00297
car	548	14064	0.29	0.696	0.58	0.32
van	548	1975	0.0963	0.233	0.0666	0.04
truck	548	750	0.286	0.0813	0.075	0.0343
tricycle	548	1045	1	0	0.0194	0.00715
awning-tricycle	548	532	1	0	0.00173	0.00104
bus	548	251	0.247	0.0956	0.0574	0.0299
motor	548	4886	0.315	0.274	0.203	0.0591

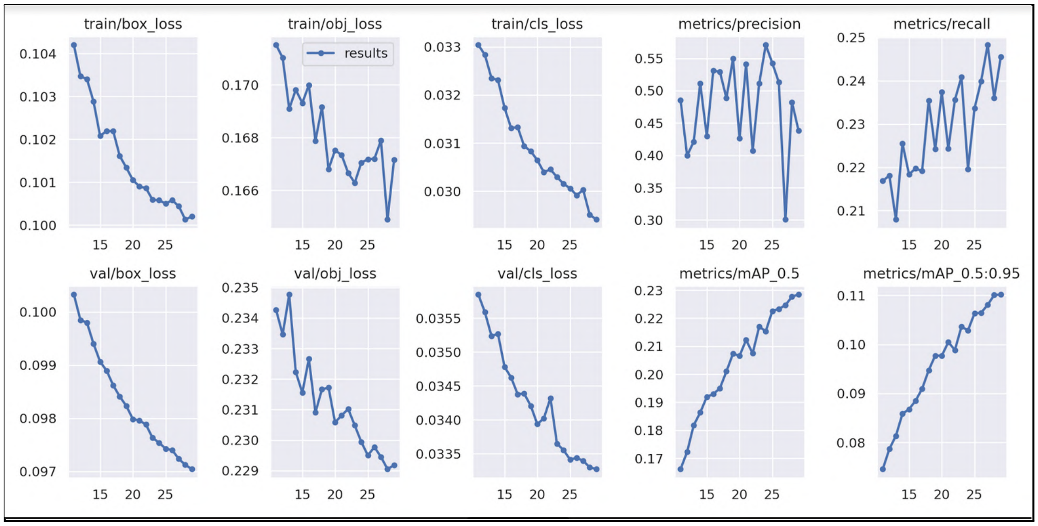
Speed: 0.1ms pre-process, 3.1ms inference, 30.6ms NMS per image at shape (32, 3, 640, 640)



2. 30 epochs

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100%
all	548	38759	0.422	0.248	0.223	0.108
pedestrian	548	8844	0.31	0.383	0.321	0.119
people	548	5125	0.338	0.294	0.238	0.0737
bicycle	548	1287	0.372	0.000777	0.0306	0.0115
car	548	14064	0.454	0.725	0.677	0.413
van	548	1975	0.27	0.203	0.163	0.107
truck	548	750	0.27	0.216	0.161	0.0876
tricycle	548	1045	0.48	0.0153	0.0812	0.0389
awning-tricycle	548	532	1	0	0.0401	0.0234
bus	548	251	0.356	0.267	0.207	0.104
motor	548	4886	0.367	0.375	0.306	0.104

Speed: 0.1ms pre-process, 3.0ms inference, 20.5ms NMS per image at shape (32, 3, 640, 640)



Ⅴ. DISCUSSIONS

A. Challenges Faced

One of the crucial challenges was to have stable hovering in our drone, it needed several steps of fine tuning the gain values(kv,kp,ki) to find a suitable value. As we didn't know the exact transfer function of our physical model, we couldn't calculate the exact values. Still, some instability persisted due to uneven weight distribution of the drone. As a result, we couldn't attach our mobile with the drone as it increased the instability. We had also faced problems with the radio communication module, as it could only send one bit at a time. We think other modules would have performed better.

In the software part, the main problem was lack of disaster survey dataset using drone imagery. As a result we had to use drone based human detection dataset (VisDrone) which gave pretty good results but it doesn't completely capture the modalities of post disaster situations so some problems in accuracy persists. Moreover as the height increases we can see sharp decline in accuracy as the model can't completely capture human features from that short pixel density.

B. Future Recommendations

To solve the hardware issues, fine tuning the controller more and if possible modeling the system using software will prove to be very useful. The modeling should be done considering the weight of the camera. Moreover for higher altitude go pro or other professional camera is recommended. We recommend using better radio modules like **TX16S** which are suitable for FPV drones.

For the human detection module, to increase accuracy transformer heads(TPH yolov5)/attention based heads are proving to be very useful. But the most important factor will be building a robust dataset for scenarios in Bangladesh. Moreover we have seen model performance

improving along with a rise in epochs, but we couldn't train for more than 30 epochs due to hardware limitations, higher computational power will yield increase in performance.
Also, along with human detection, simultaneous use of anomaly detection algorithms will improve in rescue work, which will be crucial to conduct a survey and form a suitable route for the authority to the victim.

Ⅵ. CONCLUSION

This study demonstrates the usage of drones for post natural disaster rescue and survey work. Here we have implemented a human detection algorithm(yolov5) to detect humans from live feed. Necessary fine tuning was done so that the system can work satisfactorily in post natural disaster scenarios. The drone can fly upto 1-2 stories height, where further modification is needed, but the system can perform real time human detection satisfactorily.

ACKNOWLEDGEMENT

We thank Dr. S. M. Mahbubur Rahman Sir, Professor, Bangladesh University of Engineering and Technology (BUET) for his precious guidance. We would also like to thank our teacher Tasnima Afsana (Adjunct Lecturer, department of EEE) for her support.

REFERENCES

0. Documentation(github repository)
1. How drones boost disaster response, search and rescue
2. Disaster Resilient Rescue of Coastal Community on Cyclone Warning
3. Sharifah Mastura, Syed Mohd Daud,Mohd Yusmiaidil Putera,Mohd Yusof,Chong Chin Heo, Lay See Khoo , Mansharan Kaur Chainchel Singh, Mohd Shah Mahmood, Hapizah Nawawi,"Applications of drone in disaster management: A scoping review"
4. Faine Greenwood, Erica L. Nelson, P. Gregg Greenough,"Flying into the hurricane: A case study of UAV use in damage assessment during the 2017 hurricanes in Texas and Florida"
5. How nRF24L01+ Wireless Module Works & Interface with Arduino
6. Fintan Corrigan, "Drone Gyro Stabilization, IMU And Flight Controllers Explained"
7. Atheer L. Salih, M. Moghavvemi, Haider A. F. Mohamed, and Khalaf Sallom Gaeid,"Flight PID controller design for a UAV quadrotor"
8. Wahyu Rahmaniar and Ari Hernawan Department of Electrical Engineering, National Central University, Zhongli, Taiwan, Department of Computer Science and Information Engineering, National Central University, Zhongli, Taiwan, "Real Time Human Detection Using Deep Learning on Embedded Platforms: A Review"
9. YOLOv5 official documentation
10. Shuai Shao*, Zijian Zhao*, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, Jian Sun,"CrowdHuman: A Benchmark for Detecting Human in a Crowd"

11. Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, Qinghua Hu,"Vision Meets Drones: A Challenge"
12. Mohammad amin Barekatain, Miquel Martí, Hsueh-Fu Shih, Samuel Murray, Kotaro Nakayama, Yutaka Matsuo, Helmut Prendinger,"Okutama-Action: An Aerial View Video Dataset for Concurrent Human Action Detection"
13. Renjie Xu, Haifeng Lin,Kangjie Lu,Lin Cao and Yunfei Liu,"A Forest Fire Detection System Based on Ensemble Learning "
14. Project YMFC-AL - The Arduino auto-level quadcopter
    A modified version of this code is used in our drone with support for NRF24 radio module and Python
15. https://bit.ly/37LrJ27 - resources used for the creation of the python GUI
16. Hardware - Software intercommunication using Python-ROS-Arduino framework for latency free instant communication.
17. Connecting ROS to Arduino || 5 Minutes