Design a border Surveillance System based on Autonomous Unmanned Aerial Vehicles (UAV)

Saif S. Abood*, Prof. Dr. Karim Q. Hussein, Prof. Dr. Methaq T. Gaata*****

* Department of Computer Science, Mustansiriyah University, Iraq
saifsalahabood@gmail.com
https://orcid.org/0009-0002-4938-5784

** Department of Computer Science, Mustansiriyah University, Iraq
karimzzm@yahoo.com
https://orcid.org/0000-0002-0890-4691

*** Department of Computer Science, Mustansiriyah University, Iraq
dr.methaq@uomustansiriyah.edu.iq
https://orcid.org/0000-0002-1769-6521

Abstract

After the spread of autonomous driving systems in cars, the next step is the development of autonomous drone systems. This will have an application in many military and civil fields, including surveillance and photography instead of satellites, as well as in search and rescue, dissemination, aid and distribution of goods, as it can cover large areas and save effort and human cost.

The goal of this paper is to discuss using autonomous drones to design a border surveillance system. Where the current surveillance systems were studied to discuss the possibility of using drones as an alternative to long-range cameras, and a design proposal was presented that works on three different levels of surveillance systems. The possibility of controlling the drone through a single-board computer that is installed on the control tower with the base of the charging station for the drone was also discussed.

In order to achieve the required balance between the speed of controlling the drone and the accuracy of object detection and classification, the concept of parallel processing was applied with the fast algorithms for tracking and detection (contour detection) and the high accuracy deep learning model (YOLOv8n) for detection and classification , which was trained on a customized dataset to improve the accuracy of object detection and classification in images taken from different view angles of the drone.

After testing the experimental model of the system, the results showed a good enhancement in the quality of the captured images, as well as a solution to the problem of occlusion or blurring of objects in the image.

Also, the results confirmed the possibility of relying on a single-board computer in processing images and controlling the drone to track moving targets in a way that was not possible by relying on transferring images to remote central processing.

Keywords- Autonomous Drone, border surveillance system, Unmanned Aerial Vehicle (UAV), YOLO algorithm.

I. INTRODUCTION

This section provides an overview and background of Border surveillance systems, Unmanned Aerial Vehicles (UAV) Systems, and Object Detection and Tracking. This background will help in understanding and evaluating the related works.

1) Border surveillance systems

One of the most crucial duties in the field of national defense and security is border surveillance. A country's borders must be actively monitored round-the-clock in order to preserve peace and guarantee the protection of the populace. This monitoring has become more important due to the increase in activities such as terrorist infiltration and illegal movement.

In [1] current border surveillance approaches have been classified into five main categories, namely:

i. Wired sensors-based technologies (such as fiber optic sensors);
ii. Wireless Scalar Sensor-based technologies;
iii. Wireless Multimedia Sensor-based technologies;
iv. Radars-based technologies;
v. Combined technologies (hybrids).

However, due to safety concerns, full automation of border control is not yet possible, but these systems certainly can help and work in cooperation with military forces to protect a country's borders. [2]

2) Unmanned Aerial Vehicles (UAV) Systems

The rapid developments in the capabilities of (UAV) in terms of increased flight time, processor power, as well as high-speed communication has contributed to the increasing reliance on it not only in the military field but also in the civilian fields. They have been used in a wide range of applications such as search and rescue, surveying, mapping and many other applications has led to a variety of shapes, sizes and specifications.

One of the most important of these new applications is to support surveillance systems in open and wide areas such as military battlefields, borders, external highways, electric power transmission lines, or oil and gas pipelines.

Previously, fixed-wing drones were preferred in surveillance systems because of their ability to fly for longer periods of time and longer distances, but with the advancement of technology, it became possible to use quadcopters drones and benefit from their high ability to maneuver, avoid obstacles and stay in a specific place. [3]

3) Object Detection and Tracking

The Object Detection is the use of artificial intelligence to locating different objects in pictures or videos. As for Object Tracking is the process of determining the new location of object in a frame based on knowing either its location in a previous frame from the same video (called Single Camera Object Tracking) or knowing its distinctive feature in another video (called Multi Cameras Object Tracking). [4]

There are several techniques of detection and tracking, including thermal, optical, and radar-based systems. Each type uses a different method to detect and track objects, such as using sensors to measure heat, light, or radar signals.

Detection and Tracking are used in a variety of applications, such as surveillance, navigation, autonomous vehicles, and robotics. It is can be used to identify and track people, vehicles, and other objects in real-time. Detection and tracking can provide numerous benefits, such as improved safety, increased efficiency, and better decision making. Also, it can be used to improve accuracy and reduce human error.

There are several types of object detection algorithms, including convolutional neural networks (CNNs), region-based convolutional neural networks (R-CNNs), You Only Look Once (YOLO), and Single-Shot Detectors (SSDs). Each of these algorithms has its own advantages and disadvantages.

There are two different categories within Detected Objects. Figure 1-1 shows these categories (i) Traditional detectors and (ii) Deep learning detectors, which also branch into 'One-stage detector' and 'Two-stage detector'. [4]

The Section 2 will discuss and summarize, some related research. Section 3: covers the design the proposed system and implement an experimental prototype. Section 4: lists all the algorithms. Section 5: discuss the training steps of YOLOv8n classification model with details of the datasets. Section 6: describes the details and information for each hardware parts in the prototype. Section 7: discuss the testing results of apply the proposed concepts in the experimental system and the results of training YOLOv8n model. Finally: presented the conclusions of this paper and suggested some future works.

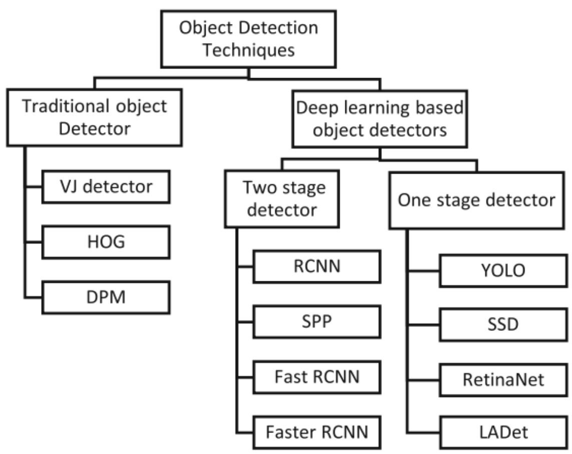

Figure 1-1: The various categories of object detection techniques [4]

II. RELATED WORKS

This section will discuss and summarize, some related research:

1) "Real-Time Object Tracking on a Drone with Multi-Inertial Sensing Data", [5]

This paper proposed a lightweight object-detection algorithm on a drone that uses Oriented FAST and Rotated Binary Robust Independent Elementary Features ORB algorithm for feature extraction, Local Difference Binary LDB for feature binary descriptors, and K-Nearest Neighbors KNN to match the image descriptors. To determine the relative position between the coordinate systems of the drone and the object, they introduced an object tracking approach integrating object recognition results with Inertial Measurement Units (IMU) data, Euclidean space equations, and Global Positioning System (GPS).

The drone is built on Qualcomm's Snapdragon flight board to achieve the highest energy efficiency. The real-time performance has been achieved by using the proposed object detection and tracking method with this drone.

2) "Smart Border Surveillance System using Wireless Sensor Network and Computer Vision" [2]

They proposed a design for an autonomous border surveillance system that does not require human assistance, it was able to take warning actions and issue alert messages for the human controllers. The idea was to deploy Multiple pyro-electric infrared sensors (PIR) along the borders and connected them to a Raspberry Pi board, which controlled (Pan Tilt Zoom (PTZ) camera movement by two stepper motors, to capture the moving object closely, then the images are analyzed using Open source TensorFlow-Object detection API to detect the object and identify the threat, The detection accuracy obtained was not mentioned.

3) "Neural Network Control for Active Cameras Using Master-Slave Setup" [6]

This research developed a new master-slave camera control method that defines the relationship between fixed cameras and PTZ cameras using neural network technology. Moreover, proposed an experimental setup to make a fair comparison between the different object detection and tracking methods. In this new experimental setup two PTZ cameras were placed side by side with very similar vision of being able to execute two different algorithms simultaneously. After that, metrics Target to Center Error (TCE) and Track Fragmentation (TF) were computed as (93.9 ± 54.7 pixels) and (97.9%)

4) "Developing Smart COVID-19 Social Distancing Surveillance Drone using YOLO",[7]

They propose a surveillance system that uses a drone to detect and identify the crowd using YOLO-v3-tiny and give social distancing warnings in an effort to prevent transmission of Covid-19. They have implemented route segmentation on the IRIS PX4 drone in the Gazebo simulator and Robot Operating System ROS. The system detected the crowds with an accuracy of about 90%.

5) "An Effective Drone Surveillance System Using Thermal Imaging", [8]

In this paper, they proposed a smart thermal imaging surveillance system by using a drone that can detect people with the help of YOLOv4 algorithm. The drone uses Radio Frequency Identification (RFID) technologies to distinguish between local military personnel and strangers. Difficulties related to security in critical areas of guarded buildings were addressed by combining the drone with long-range surveillance capability and secure data transmission.

6) "Cost-Effective Real-Time Aerial Surveillance System Using Edge Computing", [9]

In this paper, the authors suggest a low-cost method for an aerial surveillance system that keeps limited computation (Background subtraction) on board the UAV (Raspberry Pi 3), while moving heavy computation for real-time object detection tasks (YOLO and SSD) to the cloud. They got 71% accuracy for YOLO and 76.4% accuracy for SSD. But SSD works at 13 FPS whereas YOLO works at 30 FPS.

7) "Real-Time UAV Trash Monitoring System", [10].

In this study, the researchers developed a marine trash detection and real-time monitoring system. Also, the YOLOv4-Tiny-3l model was trained and implemented in the NVIDIA® Jetson Xavier NX board computer. The YOLO model could process 22 FPS (frames per second) and identify trash objects with more than 70% AP50 (Average Precision). The system has been designed to be able to manage a swarm of UAV.

8) "Solutions for the implementation of a sensors network for border surveillance", [11]

In this thesis, the researcher proposed a border control system with a multi-layer framework that detects and tracks any cross-border entry with minimal human intervention based on a combination of several technologies such as multimedia sensors, radars, and drones, and presented a detailed deployment scheme for all layers of the proposed structure. To save energy, load balancing and eliminate redundancy, with an activation scheduling strategy, the contributions of this thesis were evaluated through a simulation process.

Summary of Related works

From summarizing previous work in Table 1.1 in terms of key features and limitations, it is clear that it is difficult to achieve a balance between cost and efficiency. In some works, a high-cost drone is used, that is able to process images using lightweight algorithms, as in (1), (4), and (5). Others used a low-cost drone and relied on transferring images to be processed using high-precision algorithms at a ground station or in the cloud, as in (2), (3), and (6). The performance was not optimum in both of these cases, because of either the slow response or inaccuracy, so each method of them is suitable for different kind of applications. While the surveillance system needs a combine the high response, high accuracy, and low cost per unit distance.

Table 1-1: Related Works Summary

No.	Key Features	Limitations
1)	• Lightweight algorithm on-board • Calculated the relative position coordinate between drone and object	• Lew resolution • High-cost per unite
2)	• Deployed infrared sensors • Fast frequent response due to no recharging time	• Single viewpoint • Affected by blurring and occlusion • High cost per unit distance
3)	• used a neural network to define the correspondence between fixed and PTZ cameras • Fair comparison approach	• Single viewpoint • Affected by blurring and occlusion

4)	• Autonomies UAV • Use two cameras on the drone	• Simulator
5)	• Used thermal camera • Used RFID technologies to identify the enemy	• High-cost per unite
6)	• Moving heavy computation to cloud. • Using edge computing technique on UAV to decrease the network delay	• Need a constant connection
7)	• Autonomies UAV • Implemented YOLOv4-Tiny model in the NVIDIA® Jetson Xavier NX board on UAV.	• Manually flight was used to obtain a clear view of the different trash sizes. • High-cost per unite
8)	• Combine several technologies • Provide a detailed deployment scheme	• Humanly classification of the objects

III. DESIGN AND IMPLEMENT THE PROPOSED SYSTEM

The basic idea of the proposed system depends on two concepts, each one solves a part of the problem. The first concept is the use of drones instead of the long-range cameras. This concept solves the problem of distance and image blurring, or the problem of obstruction and not showing the whole object. The second concept is to use speed tracking algorithms in parallel with slow classification algorithms to achieve fast control of the drone in real-time and high classification accuracy. The single Board computer (such as the Raspberry Pi) controls the drones automatically instead of controlling the PTZ camera. The drones will provide closer and high-resolution images, and thus the process of recognizing and classifying the moving object will be easier and more accurate.

However, there will be some new challenges as the detection and tracking algorithms applied to moving camera images are more complex and require more computation than the algorithms applied to fixed camera images.

A. General Operations of the Proposed System

The Figure 3-1 shows the basic operations performed by the single-board computer when using the drone, for example, with a wide-angle camera. It begins with processing and analyzing camera images to detect any moving object, calculate its location, and try to classify it. Then, if the moving object is not classified because of the distance and blurring of the image, or because of the occlusion and not showing the whole object, the drone is automatically sent toward the location of the moving object.

Here, the Single Board Computer will process and analyze the image captured by the drone to control the drone to arrives near the moving object and take a close-up image of it for classification. Then the drone is controlled to return and land at the charging station in the tower.

Finally, the pictures and information are sent in the form of a report to the control and monitoring center to determine the importance and seriousness of the situation and issue the appropriate warning.

In the case of using the drone with motion sensors or radars, the drone will take off immediately after detecting the movement, and the single-board computer will complete the rest of the operations as mentioned previously.

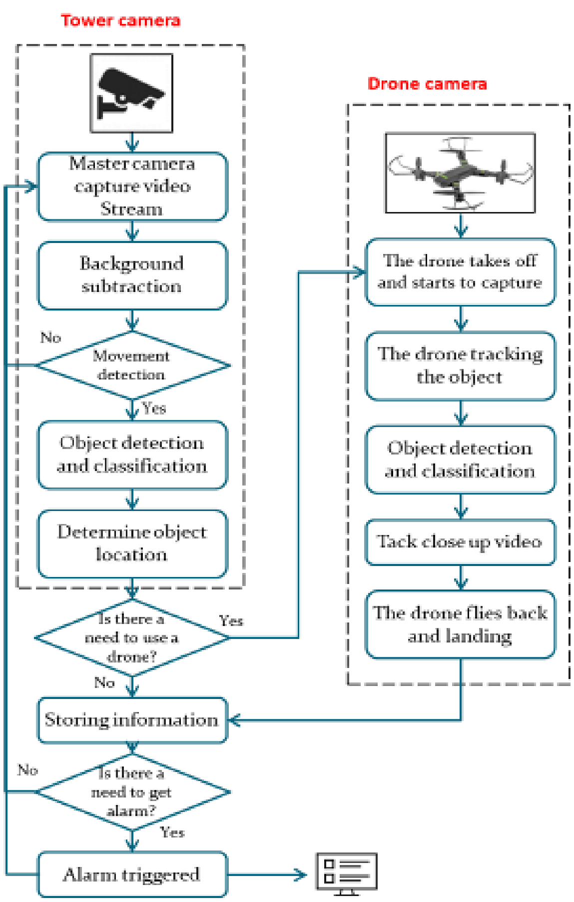

Figure 3-1:General block diagram of using the drones with the wide-view angle camera

B. Structure of the Proposed Method

The proposed method depends on running the algorithms in two parallel loops, a fast main loop to control the drone by using the contours detection algorithm to tracking the target, and the second loop which is a slow loop to analyze the image and classify the object by run the YOLOv8n algorithm for objects detection and classification.

The main loop begins with motion detection, whether by using a wide-angle camera, motion sensors, or radar, where the coordinates of the target are determined on the ground, and the drone is sent to this location, and begin imaging with the camera of drone.

A copy of the current frame (cf) is sent to the second loop thread if it is idle or has finished processing the previous frame to detect and classify the target. then, the current frame (cf) is cropped using the minimum and maximum coordinates in the (target box) to get the Region of Interest (RoI).

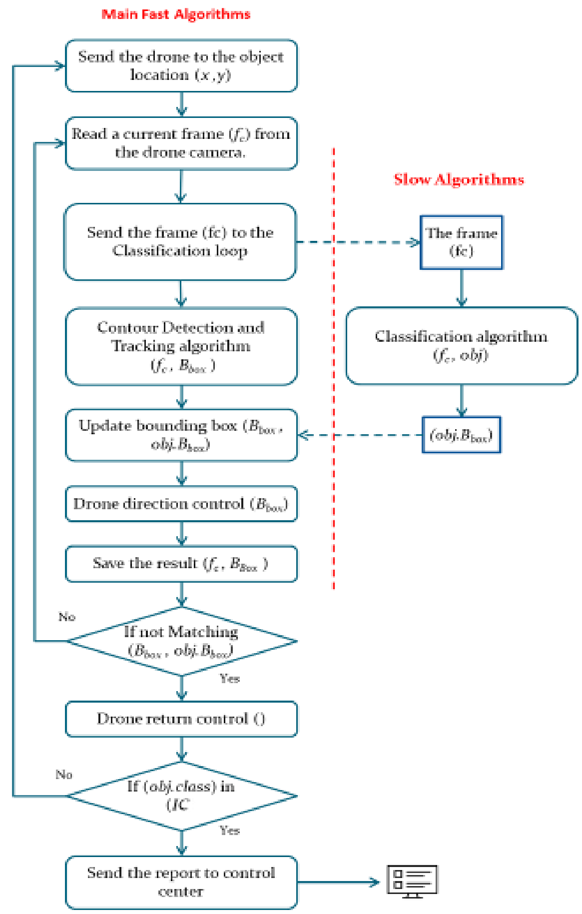

Figure 3-2: flowchart of proposed method

In the classification loop, after receiving the size of the current frame (cf) must be changed to match the input of the classification model, e.g., in the YOLO model a padding is added to make the height and width are multiple of 32. After applying the YOLO and get the result as a list of object bounding boxes with classification of each one, a quick comparison of the bounding boxes (Bbox) is made with the bounding box of the target object (obj.Box) by calculated the (IOU) for them, and update the coordinates of the target object.

In the end of classification loop, re-calculate the coordinates of (obj.Box) with respect to (cf) and return the classified object (obj.clas) to the main loop and receive a new frame.

After first time detect the target, the contours detection algorithm begins processing the (RoI) in new frames to tracking the target, and the second loop thread still running to detect and classify the target and continuously update the (RoI) coordinates.

In the Contours detection algorithm, the (RoI) is converted to grayscale and smoothing by using a Gaussian filter. Then, apply Binary Thresholding filter and Morphological filter before using the Canny filter to detect edges, and found contours by compressing horizontal, vertical, and diagonal edge segments, leaving only their end points to reduce their size in memory.

At the same time in the step of update and matching the bounding boxes (Bbox) from the contour detection algorithm are compared and matching with the bounding box (Obj.Bbox) from the classification algorithm. The (IOU) with each bounding box is calculated and update the coordinates of the moving object.

The drone is directed by receiving a control command containing four movement variables, (left/right, forward/backward, up/down, yaw), these four variables represent the percentage of the movement speed in each direction and their values range from 100 to -100, so it's similar to the joystick commands.

The values of movement variables are calculated through the Proportional-Integral-Derivative PID Control equation, which calculates the error ratio between the center of the camera image and the center of the bounding box of the target.

After the drone arrives near the target, it begins to fly around the target to imaging it from all sides. This is done by moving the drone to the right or left with re-orientating the drone to the target based on the location of the bounding boxes in the image. And to complete the 360-degree rotation, the angle of the drone is continuously compared the current angle with the angle when start rotation.

Finally, if the drone has finished photograph the target from all sides and the target is classified, the images are saved and the drone is returned to the base station. If the detected object is in the list of classes of interest (IC), a warning report is sent to the control center.

C. Autonomous Flying Strategy

The flying strategy of the autonomous drone can be written in these rules:

Rule 1: If the drone is in tracking and it is far from the target:
- Moving forward toward the target.

Rule 2: If the drone is in tracking and it is close to the target and the rotation has not started:
- Start the rotation.
- Record the direction of start rotation.

Rule 3: If the drone is in a rotation and the direction of the drone is not equal to the direction of start rotation:
- Take a picture of the target.
- Move around the target counterclockwise.

Rule 4: If the drone is in a rotation and the direction of the drone is equal to the direction of start rotation:
- Finish tracking.
- Finish rotation.
- Start fly-back to base.

Rule 5: If the drone is in fly-back and it is far from the base:
- Move forward towards the base

Rule 6: If the drone is in fly-back and it is above the base:

ISSN: 2710-2165

- Landing.
- wait for a new task

The Figure 3-3 shows a screenshot of the simulation of the drone flight strategy in the proposed system, where the tower, radar, and drone are located on the left of the screen, and when clicking the mouse in any location on the screen, a top-view image of the object appears in this place, that object can be moved in all directions through the keyboard. and immediately the drone flies towards this object in order to tracking and photograph it from all sides, and return to the tower. where the object images are displayed on the right side of the screen, and report information such as the object's class, location, time and date are written in the upper left side of the screen.

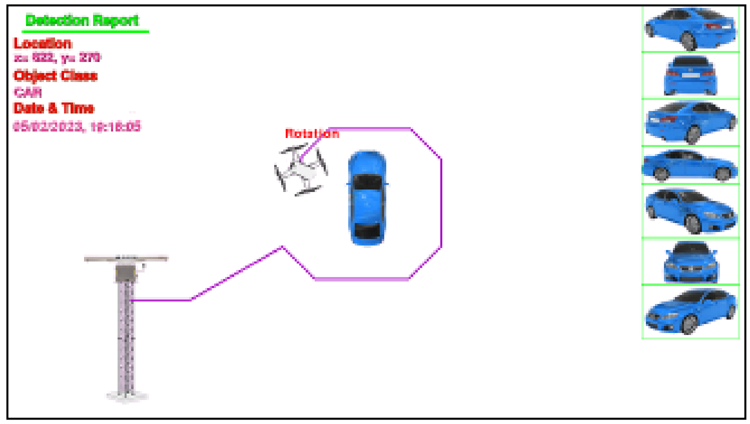

Figure 3-3:Simulation of Flying Strategy

IV. THE ALGORITHMS

The Main Loop Algorithm (4.1)
Input: Detected object location (x, y), tower base station location (x_o,y_o) , list of Interest Classes (*IC*). **Output:** Report about the classified object (*obj*).
Begin **Step1:** Send the drone to the object location (x, y) **Step2:** Read a current frame (*cf*) from the drone camera. **Step3:** If the Classification loop is idle Then Call the Classification loop algorithm (cf,$B._{box}$,*obj*) in new thread. (4.2) **Step4:** Call the algorithm of Contours Detection (cf, B_{box}) (4.3) **Step5:** Update bounding box (B_{box}, $obj.B_{box}$) **Step6:** Drone direction control (B_{box}) **Step7:** Save the result (cf, B_{Box}). **Step8:** If not Matching (B_{box}, $obj.B_{box}$) then go to step2. **Step9:** Drone return control (x_o,y_o) **Step10:** If (*obj.class*) in (*IC*) Then Send the report to control center **Step11:** Repeat steps (1...10) until stop **End**

The Classification Loop Algorithm (4.2)
Input: current frame (*cf*), list of bounding boxes (B_{box}). **Output:** list of classified objects (*obj*).
Begin **Step1:** Receive the current frame (*cf*) from the main loop, **Step2:** Resize the (*cf*) to fit the inputs of the classification model. **Step3:** Use YOLOSv8 model to detect and classify the objects in (*RoI*). **Step4:** Recalculate the coordinates of ($obj.B_{box}$) relative to (*cf*). **Step5:** Return ($obj.B_{box}$) to the main loop **Step6:** Repeat steps (1...5) until stop **End**

The Algorithm of Contours Detection (4.3)
Input: current frame (*cf*), **Output:** list of bounding boxes (B_{box}).
Begin **Step1**: Crop the region of interest (*RoI*) from the (*cf*). **Step2**: Convert (*RoI*) to grayscale **Step3**: Smoothing (*cf*) using Gaussian filter. **Step4**: Apply binary Threshold filter and morphological filter. **Step5**: Edge detection by applying Canny filter. **Step6**: Find Contours by compresses edge segments. **Step7**: Calculate the area of each contour and remove small ones. **Step8**: Calculate the bounding boxes (B_{box}) of each remaining contour.

Step9: Return (B_{box}) list
End

V. YOLOv8

In January 2023, Ultralytics (the same company that previously released YOLOv5) released the YOLOv8 on GitHub sit, without publishing any research paper on YOLOv8, it will be compared to previous versions of YOLO based on the code and its documentation.

YOLOv8 offers five different scaled versions: YOLOv8x (extra-large), YOLOv8l (large), YOLOv8m (medium), YOLOv8s (small), and YOLOv8n (nano). When evaluated using the 2017 test-dev MS COCO dataset, with an image size of 640 pixels, YOLOv8x achieved an AP of 53.9%. compared to YOLOv5's 50.7% on the same input size. It also has a fast speed.

Figure 5-2:Different view angles of the drone

YOLOv8 includes numerous improvements and architectural changes. it can be used to perform detection, segmentation, classification, tracking, and pose estimation.

Also, YOLOv8 comes with features that make it easy for developers to use, such as a command line interface (CLI) and a well-structured Python package.

YOLOv8 is anchor-free, predicting fewer bounding boxes, so the Non-maximum Suppression (NMS) process becomes faster. During YOLOv8 training mosaic augmentation is used, however, it is turned off for the last 10 epochs since continuous usage of this method might be harmful.

The bottleneck is the same as that of YOLOv5, which means that YOLOv8 is starting to revert to the ResNet block defined in 2015.

In the neck, the count of parameters and the overall size of the tensor were reduced by directly connecting the features without imposing the same channel dimensions.

YOLOv8 enhances images during training. Whereas in each epoch the model presents a slightly different set of images. one of these enhances are Mosaic augmentation where the model is compelled to learn things in new places, In partial occlusion, and against various surrounding pixels as a result of stitching 4 images together.

However, it has been demonstrated empirically that if this augmentation is used throughout the whole training regimen, performance will suffer. It is beneficial to disable it during the last ten training epochs.

With the distribution of the ultralytics package comes a CLI. Many YOLOv5 users would be familiar with this because CLI was also used for the core training, detection, and export interactions.

1) Training YOLOv8n Classification model

When choosing a detection algorithm, a balance must be achieved between accuracy and speed to achieve real-time operation requirements, as the control of the aircraft will depend on tracking the detected object.

The proposed algorithm to use is YOLOv8n, which is the lighter version of the latest version of the YOLO algorithm for deep learning.

As in Figure 5-2, the angle of view of the drone differs from the horizontal angle of view, and the difference increases according to its height and distance from the target. Therefore, although a pre-trained neural network model for the YOLOv8n algorithm was available. the neural network had to be trained on a customized set of images from different view angles of the drone.

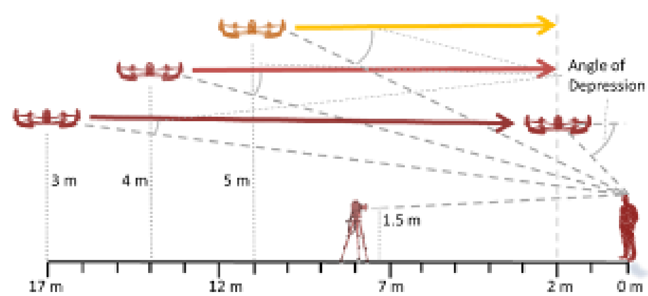

Figure 5-2: YOLOv8 model tasks

2) YOLO training steps

The training of the YOLOv8n object detection model on custom data can be summarized in these steps:

Step 1: Collect Data

The training data was collected by selecting 9.840 image frames from the UAVDT dataset consisting of 100 video sequences, captured using the UAV platform at a number of locations in different regions. The videos are recorded at 30 fps, at a resolution of 1080 x 540 pixels. Approximately, one frame was selected from each second of video. and the frames are cropped to a resolution of 540 x 540

Step 2: Label the data

The images were labeled using the Roboflow website, which is an integrated platform for image labeling and exporting them in different formats for machine learning models. There are also other programs and tools that can be used. such as the Labeling tool. The image labeled in (5) classes (car, bus, truck, motorcycle, person).

Step 3: Split Data

Before training a computer vision model on custom data, the data must be divided into a Training set, Validation and a Test set. Training set: is the largest set from the dataset that is used to teach the model how to make predictions. As a default, it is recommended to allocate 70% of the dataset to the training set. Validation set: it is the set of data that is used during training to test how well the model performs and to choose when to stop training, and it is not included in the training. It is recommended to keep 20% of the data set for the validation set.

Test set: It is the set of data that is used to run evaluation metrics to evaluate the accuracy of the model. It is recommended to allocate 10% of the data set to the test set

The percentage for division is depending on the size of the data set and the task set for it.

Step 4: Create configuration files

A helpful way for organizing and storing all significant parameters of training a computer vision model is by create a custom configuration file. which includes the correct path to the dataset folder, the number of classes, and their names.

Step 5: Start Training

After completing the previous steps, training YOLOv8 on the custom data can be started using the command below (for example) in (Command Prompt) or in the terminal.

yolo task=detect mode=train model=yolov8n.pt data=/data.yaml epochs=80 imgsz=640
task = detect (It can he segment or classify)
mode = train (It can be predict or val)
model = yolov8n.pt (It can yolov8s/ yolov8I/ yolov8x)
epochs = 80 (It can be any number)
imgsz = 640 (It can be 320, 416, etc., but make sure it needs to be a multiple of 32)

VI. THE HARDWARE OF THE SYSTEM

This section reviews the details of the hardware requirements for implementing an experimental model of the proposed system (Figure 6-1) and the diagram of the connections (Figure 6-2), Which are as follows:

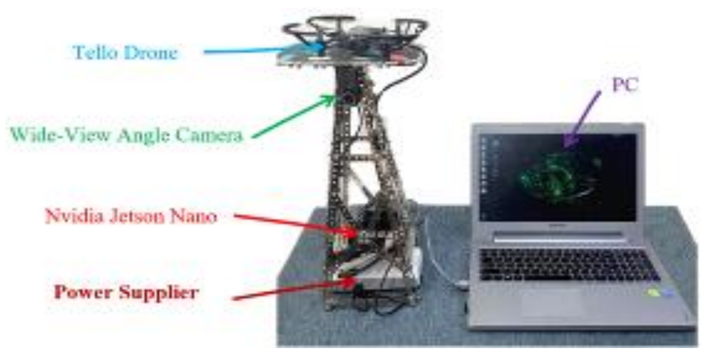

Figure 6-1: Hardware requirements

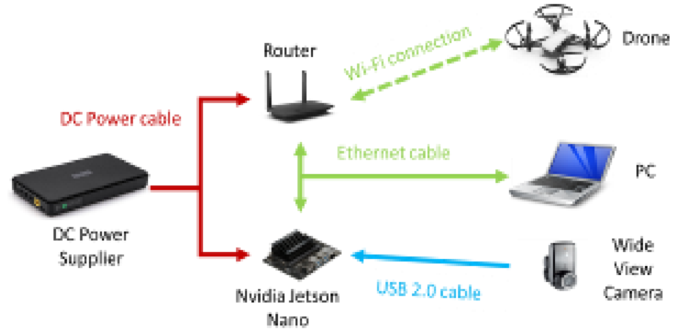

Figure 6-2: hardware connections diagram

1) Single Board Computer

When choosing the appropriate Single Board Computer, the complexity of the algorithms used in image processing must be taken into account, in addition to the algorithms for controlling the flight of the drone, where the required processor power and the amount of random memory must be calculated. It should also provide a fast Wi-Fi connection with the appropriate range. And through research, we found two devices can meet these requirements.

The first device is an Nvidia Jetson Nano (Figure 6-3), which is the best, so it was chosen in this research. It has 4GB of memory and a 1.34GHz Octa-core processor with a 64GB SD card, and a 2.4GHz wireless LAN. It is a microcontroller dedicated to image processing and implementation of deep learning algorithms, but it is not widely known.

The second device is the fourth version of the well-known Raspberry Pi4 series, with 4 GB of memory and a 1.5 GHz quad-core processor with a 64 GB SD card, and 2.4 / 5.0 GHz wireless. LAN. It is a less powerful device, but it is easier to use.

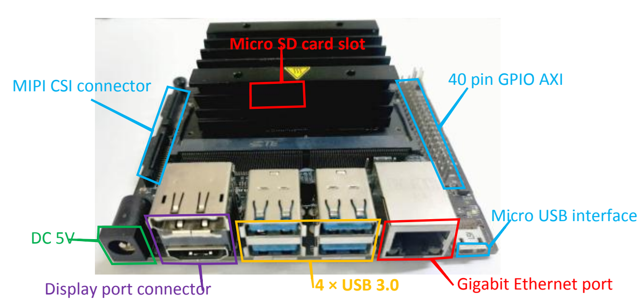

Figure 6-1: Nvidia Jetson Nano

2) The drone

A drone that fits this proposal system must be a lightweight (quadcopter) drone so that its uses battery power only for flying, imaging, and sending images to the tower. It must be fast and maneuverable and have a base station for charging.

In the practical experiments of this research, the educational drone Tello.edu from DJI was used, which is a small, lightweight plane that has Intel Processor and can be programmed using the Tello SDK library.

The weight of this drone is 87g (including propellers and battery), dimensions are 98 × 92.5 × 41mm with 3-inch propellers, It can fly precision hovering for 13 minutes at a maximum speed of 28.8 km/h or 8m/s, It also has a Swarming Capabilities, Range Finder, Barometer, System Vision, 2.4 GHz Wi-Fi Port: Micro USB Charging Port.

The drone's camera captures JPG Photos in 5MP (2592×1936) and HD 720P MP4 Videos with 82.6º Field of View (FOV) and Electronic image stabilization (EIS)

Figure 6-2: Tello.edu DJI drone

VII. THE RESULTS DISCUSSION

1) Results of using the Drones instead of the Long-Range Cameras.

When applying the first concept in the main idea of the proposed system, which is the use of drones instead of long-range cameras, the problem of distance and image distortion or occlusion problem was solved.

Also, the results showed a significant improvement in the ability to track a moving object, as well as reducing the probability of misclassification, where it became possible to photograph the object from several different angles of view and from a close distance, so that the image is not affected by weather conditions.

In addition, the images which are attached to the reports are more

useful to the operator in the control center because of the high amount of details they contain, where it is possible to extract additional information from the images that were not previously available, such as facial recognition or the license plate number. In (Figure 7-) it is difficult to identify the person or read the license plate number in the distant images even with a zoom-in, while in (**خطأ! لم يتم العثور على مصدر المرجع.**) the close-up image clearly shows the details of the person and the license plate number.

Figure 07-1: distant images with a zoom-in

Figure 07-2: close-up image clearly shows the details

2) Results of Parallel use of Detection and Classification Algorithms.

When applying the second concept in the main idea of the proposed system, which is using the speed of the contour detection algorithms in parallel with the accuracy of the YOLOv8 classification algorithms, the required balance was achieved between the speed of controlling the drone in real time with the high classification accuracy of the objects in the images.

In the (Figure 7-), it can be clearly seen the significant difference between the speed of the contour detection algorithm and the YOLOv8 classification algorithm.

It can also be noted that the speed of the contour algorithm decreases with the increase in the number of objects in the image, while the speed of the YOLOv8 algorithm is not affected by that.

The other thing is that the accuracy of the YOLOv8 algorithm is higher in determining the bounding box, as the contour algorithm does not distinguish between an object and its shadow.

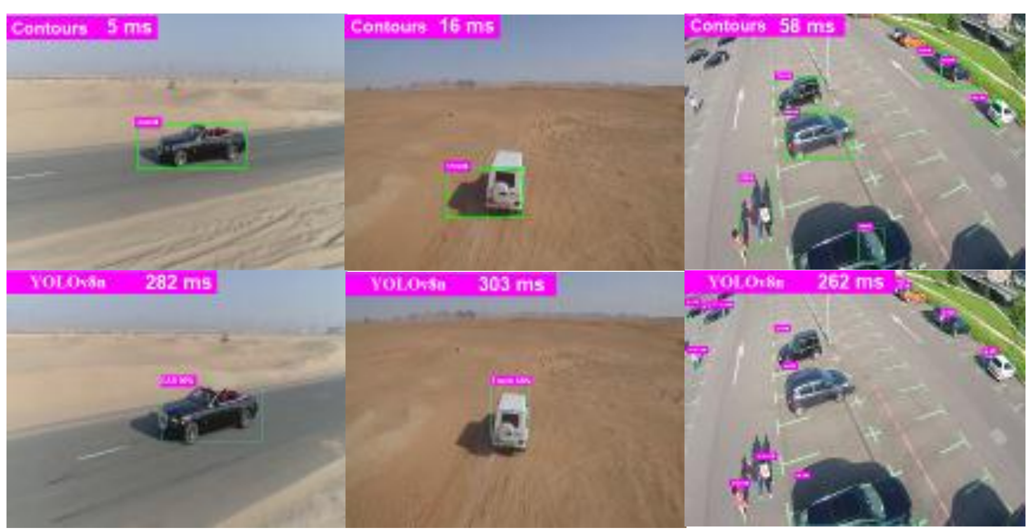

Figure 7-3: The different speed between detection and classification algorithms

3) **The results of training YOLOv8n model.**

The (Figure 7-4) and (Figure 7-5) present the metrics and performance measures of training the YOLOv8n model on the UAV imaging dataset. It contains 9840 images labeled into (5) classes (person, car, motorcycle, bus, truck).

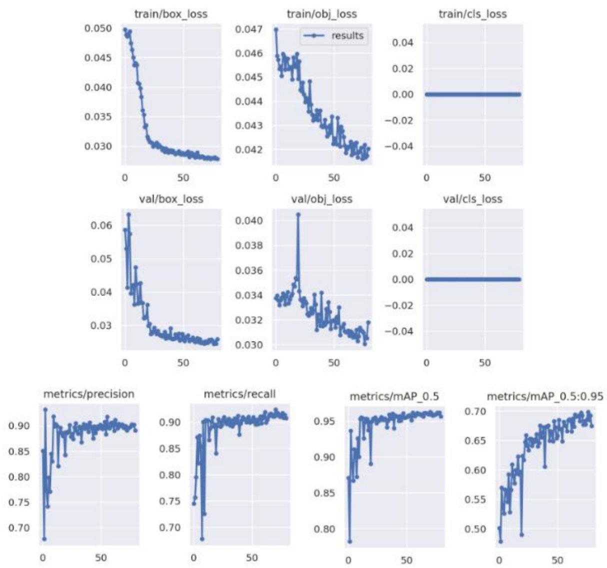

Figure 7-4:The training and validation set's metrics

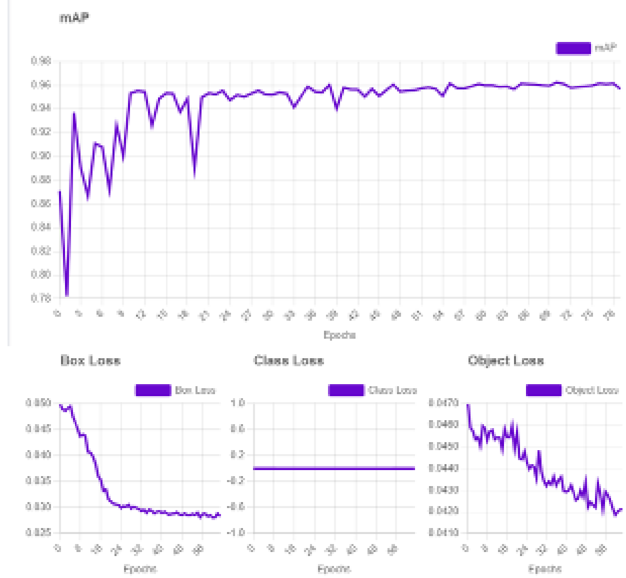

Figure 7-5: The performance measures of the trained

The train loss functions are represented by the three graphs in the first row, all of which are decreasing over time. This is similarly true for the graphs of the validation loss functions in the second row, with the exception that the object loss function had a small peak after about 20 epochs before returning to a decreasing trend.

Additionally, it can be said that the precision and recall functions are both improving, which indicates good performance. After roughly 80 epochs, the mAP 0.5 metrics value is already moving in the direction of one. This implies that a larger value will be obtained the more over-detection happens.

Figure 7-6: Comparing the custom-trained model with the pre-trained model

Despite the use of a large dataset in training the model, the results of the training did not differ much from the results of the pre-trained model provided by Ultralytics on its account on the GitHub site. Where a slight improvement appeared in the classification accuracy of the images taken from top view. While reducing the number of classes did not lead to an increase in processing speed.

VIII. CONCLUSION

In this paper, a border control system based on autonomous drones instead of long-range cameras is designed and implemented. It also adopts a single-board computer to control the drone. The efficiency of the proposed system was tested in real time, simulation and by using the dataset. The result was achieving targets in terms of execution time and accuracy. The conclusions drawn from the results analysis are summarized in the following notes:

• The use of drones is a radical solution to the problems of long-range cameras, but it increases the difficulty of image processing due to changing the angle of view.

• The use of parallel processing of two different types of algorithms achieves a balance between speed and accuracy.

IX. FUTURE WORK

This research achieved an important step that paves the way for the development of more complex systems. The following is a set of proposed future works:

- Developing systems capable of operating swarms of drones, where cooperation, information exchange, task distribution, and high reliability and fault tolerance are achieved.
- Training deep learning models to detect viewing angle, direction and distance of moving vehicles
- Developing a drone control system capable of overcoming obstacles
- Developing a platform with an automated or wireless charging station for the drone.

REFERENCES

[1] M. Lamine Laouira, "*Solutions for the implementation of a sensors network for border surveillance*." [Online]. Available: https://theses.hal.science/tel-03735980

[2] N. Bhadwal, A. Shukla, V. Madaan, A. Kakran, and P. Agrawal, "*Smart Border Surveillance System using Wireless Sensor Network and Computer Vision.*" In 2019 international conference on Automation, Computational and Technology Management (ICACTM) (pp. 183-190). IEEE.

[3] A. Aabid et al., "*Reviews on design and development of unmanned aerial vehicle (drone) for different applications*," Journal of Mechanical Engineering Research and Developments (Scopus) , vol. 45, No. 2, pp. 53–69, Accessed: Apr. 22, 2023. [Online]. Available: https://jmerd.net/Paper/Vol.45,No.2(2022)/53-69.pdf

[4] J. Kaur and W. Singh, "*Tools, techniques, datasets and application areas for object detection in an image: a review*," Multimedia Tools and Applications (Scopus), vol. 81, No. 27, pp. 38297–38351, Nov. 2022, doi: 10.1007/s11042-022-13153-y.

[5] P. Chen, Y. Dang, R. Liang, W. Zhu, and X. He, "*Real-Time Object Tracking on a Drone with Multi-Inertial Sensing Data*," IEEE Transactions on Intelligent Transportation Systems (Scopus), vol. 19, No. 1, pp. 131–139, Jan. 2018, doi: 10.1109/TITS.2017.2750091.

[6] R. O. Reis, I. Dias, and W. Robson Schwartz, "*Neural Network Control for Active Cameras Using Master-Slave Setup*." [Online]. Available: https://www.axis.com/en-us/support/developer-support/vapix

[7] P. Somaldo, F. A. Ferdiansyah, G. Jati, and W. Jatmiko, "*Developing Smart COVID-19 Social Distancing Surveillance Drone using YOLO Implemented in Robot Operating System simulation environment*," in IEEE Region 10 Humanitarian Technology Conference, R10-HTC, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/R10-HTC49770.2020.9357040.

[8] M. Sneha et al., "*An effective drone surveillance system using thermal imaging*," Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2020, pp. 477–482, Oct. 2020, doi: 10.1109/ICSTCEE49637.2020.9277292.

[9] M. Shahzad Alam and S. K. Gupta, "*Cost-effective real-time aerial surveillance system using edge computing,*" in Lecture Notes in Civil Engineering, (Scopus) Springer, 2020, pp. 289–299. doi: 10.1007/978-3-030-37393-1_25.

[10] Y.-H. ; Liao, J.-G. Juang, Y.-H. Liao, and J.-G. Juang, "*Real-Time UAV Trash Monitoring System*," Applied Sciences 2022 (Scopus) , Vol. 12, Page 1838, vol. 12, no. 4, p. 1838, Feb. 2022, doi: 10.3390/APP12041838.

[11] Mohamed Lamine Laouira, "*Solutions for the implementation of a sensors network for border surveillance. PhD Thesis. USTHB-Alger | Request PDF,*" 2022. https://www.researchgate.net/publication/362325635 Solutions_for_the_implementation of_a sensors network_for_border_surveillance (accessed Aug. 02, 2023).

[12] J. Terven and D. Cordova-Esparza, "*A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond*," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.00501

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "*You Only Look Once: Unified, Real-Time Object Detection*", In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). Accessed: Jul. 03, 2023. [Online]. Available: http://pjreddie.com/yolo/

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "*You Only Look Once: Unified, Real-Time Object Detection*," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, pp. 779–788, Jun. 2015, doi: 10.1109/CVPR.2016.91.

[15] D. Thuan, "*Do Thuan EVOLUTION OF YOLO ALGORITHM AND YOLOV5: THE STATE-OF-THE-ART OBJECT DETECTION ALGORITHM*" Bachelor's Thesis. Oulu University of Applied Sciences 2021.

[16] B. Xu, N. Wang, T. Chen, and M. Li, "*Empirical Evaluation of Rectified Activations in Convolutional Network*," May 2015, [Online]. Available: http://arxiv.org/abs/1505.00853

[17] J. Redmon and A. Farhadi, "*YOLO9000: Better, faster, stronger*," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017. doi: 10.1109/CVPR.2017.690.

[18] J. Redmon, A. F. ön baskı ArXiv:1804.02767, and U. 2018, "*YOLOv3: An Incremental Improvement*," Arxiv.Org, 2018, Accessed: Jul. 18, 2023. [Online]. Available: https://click.endnote.com/viewer?doi=arxiv%3A1804.02767&token=WzQwNzM1MTUsImFyeGl2OjE4MDQuMDI3NjciXQ.lSgGK5sHlR34I_uXzlEKQFsXoKk

[19] K. He, X. Zhang, S. Ren, and J. Sun, "*Deep Residual Learning for Image Recognition*" In Proceedings of the IEEE conference on computer vision and pattern recognition, Accessed: Jul. 18, 2023. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "*YOLOv4: Optimal Speed and Accuracy of Object Detection*," Apr. 2020, Accessed: Jul. 18, 2023. [Online]. Available: https://arxiv.org/abs/2004.10934v1

[21] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "*CSPNET: A NEW BACKBONE THAT CAN ENHANCE LEARNING CAPABILITY OF CNN A PREPRINT*," In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 390-391) 2019, Accessed: Jul. 18, 2023. [Online]. Available: https://github.com/WongKinYiu/CrossStagePartialNetworks.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "*Densely connected convolutional networks*," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017. doi: 10.1109/CVPR.2017.243.

[23] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection," China, Information Sciences, 522, 241-258. 2017. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1903/1903.08589.pdf

[24] Jonathan Hui, "*YOLOv4*." , Medium. https://jonathan-hui.medium.com/yolov4-c9901eaa8e61 (accessed Jul. 19, 2023).

[25] P. Y. Chen, J. W. Hsieh, M. Gochoo, C. Y. Wang, and H. Y. M. Liao, "*Smaller Object Detection for Real-Time Embedded Traffic Flow Estimation Using Fish-Eye Cameras,*" in Proceedings - International Conference on Image Processing, ICIP, IEEE Computer Society, Sep. 2019, pp. 2956–2960. doi: 10.1109/ICIP.2019.8803719.

[26] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "*Path Aggregation Network for Instance Segmentation*," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018. doi: 10.1109/CVPR.2018.00913.

[27] J. Solawetz, S. S. JUL 1, and 2020 6 Min Read, "*Train YOLOv4-tiny on Custom Data - Lightning Fast Object Detection*," Roboflow Blog. 2020.

[28] G. Jocher et al., "*ultralytics/yolov5: Initial Release*," Jun. 2020, doi: 10.5281/ZENODO.3908560.

[29] C. Li et al., "*YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.02976

[30] Chu Yi et al., "*YOLOv6: The fast and accurate target detection framework is open source*," Jun. 23, 2022. https://tech.meituan.com/2022/06/23/yolov6-a-fast-and-accurate-target-detection-framework-is-opening-source.html

[31] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "*OTA: Optimal Transport Assignment for Object Detection*," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 303–312, Mar. 2021, doi: 10.1109/CVPR46437.2021.00037.

[32] Z. Gevorgyan, "*SIoU loss: More powerful learning for bounding box regression*." arXiv preprint (2022) arXiv:2205.12740.

[33] L. Xie et al., "*Elan: Towards Generic and Efficient Elastic Training for Deep Learning*." In 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS) (pp. 78-88). IEEE.

[34] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "*YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*," Jul. 2022, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7464-7475) [Online]. Available: http://arxiv.org/abs/2207.02696

[35] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[36] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[37] J. Wang, "*Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)*," IEEE J. Quantum Electron., submitted for publication.