SWAYAM NPTEL COURSE ON

MINE AUTOMATION AND DATA ANALYTICS

By

Prof. Radhakanta Koner

Department of Mining Engineering

Indian Institute of Technology (Indian School of Mines) Dhanbad

Module 06

Automated tracking and VR systems

Lecture 13 B

Automated communication and tracking technologies: Image processing

CONCEPTS COVERED

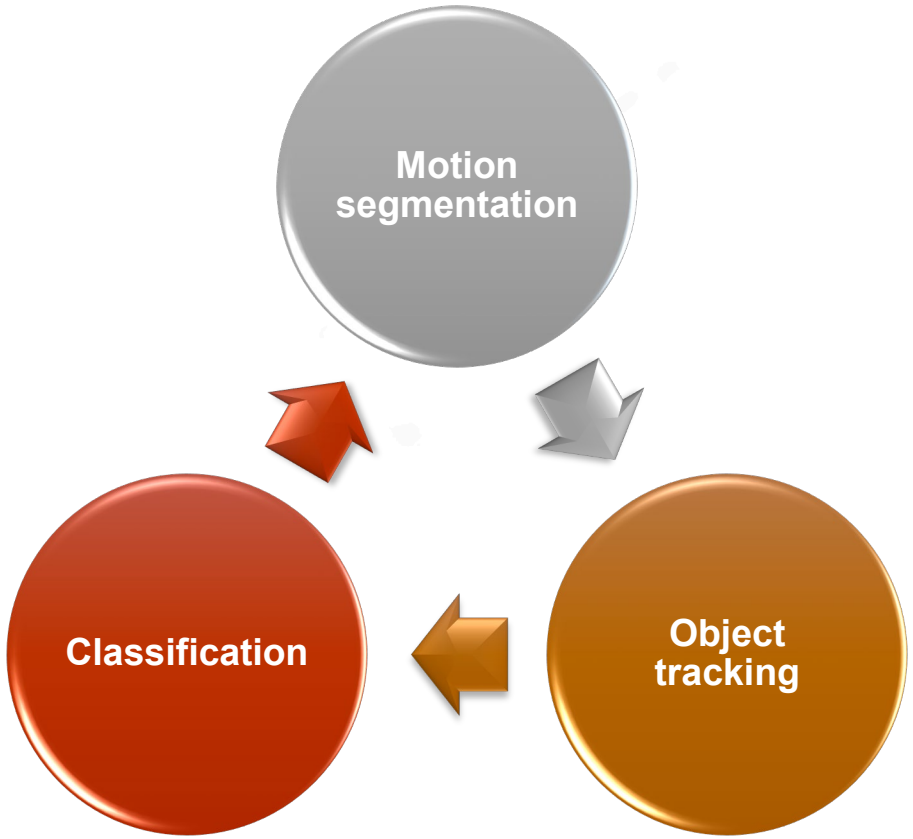
- Introduction to Automated Communication and Tracking Technology
- Case study: Real-Time Object Detection and Tracking for Unmanned Aerial Vehicles Based on Convolutional Neural Networks

Automated Communication and Tracking Technology

- Image processing is a form of processing with input as image such as photograph or video frame and output can be characteristics or parameters related to image.
- Computer vision is an area that consists of methods for incorporating, analyzing and visualizing images.
- Surveillance stands for monitoring the behaviour, activities, and other changing information, usually of people for the purpose of influencing, directing and protecting them.

- The process of locating moving object using a camera is video tracking.
- In simple terms, tracking means associate target objects in consecutive video frames.
- Difficulties arise especially when objects are moving rapidly as compared to frame rate or when the tracked object changes direction over time.
- A sequential flow of object detection, object tracking, object identification and its behavior completes the process framework of tracking.

Object tracking in video surveillance is a very important aspect of computer vision and pattern recognition. The common architecture of classification consists of three main steps are



Step-1 Motion segmentation

- Object detection is a computer vision technology that deals with identifying instances of objects such as humans, vehicles, animals or birds and other moving objects. Object detection is one of the initial steps for object tracking.
- A video surveillance system for stationary cameras generally includes some part of motion detection.

Background Subtraction

Temporal Differencing

Optical Flow

Step-2 Object classification

Classification is a process in which individual items like objects, patters, image regions, pixels, etc. are grouped based on the similarity between the item and the description of the group. In general, object classification in video surveillance are

- Shape-Based Classification
- Motion-Based Classification
- Color-Based Classification
- Texture-Based Classification

Step-3 Object tracking

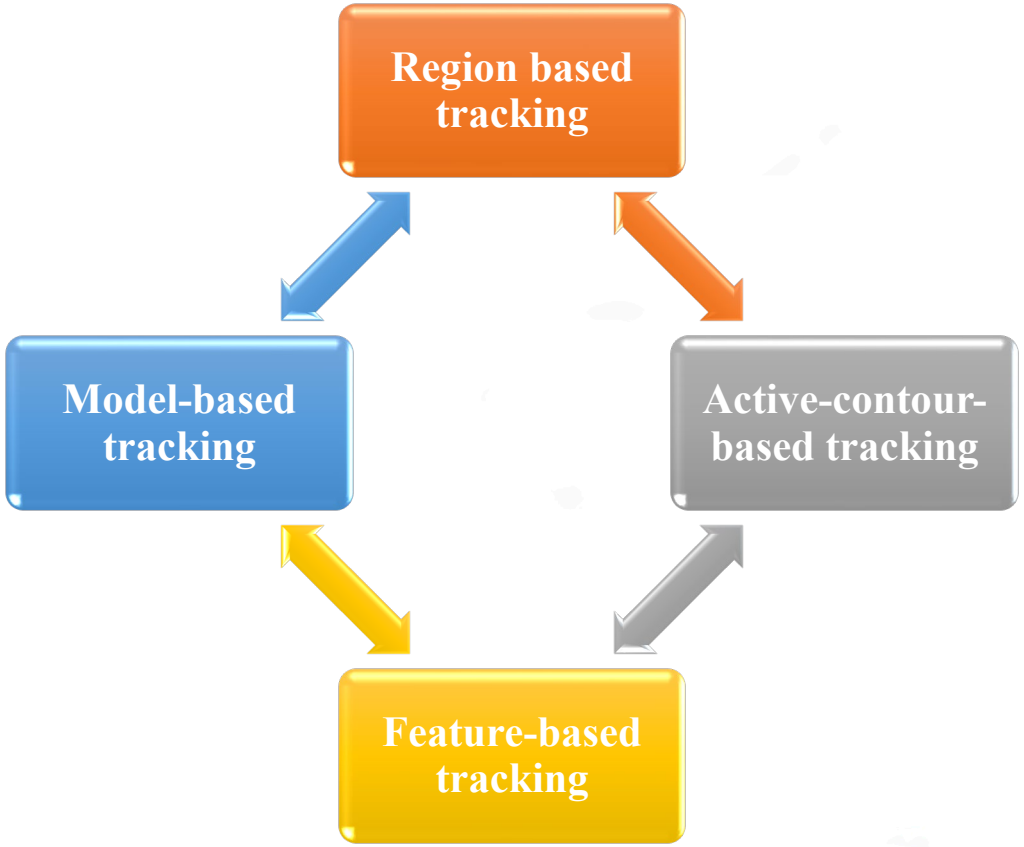
- In simple terms tracking is the problem of estimating the trajectory of an object in the image plane as it moves around a scene.

- Depending upon tracking domain, method and algorithm, a tracker can also provide object centric information like area, orientation and shape of an object. Once objects are detected, the next task in video surveillance process is to track the objects from one frame to another.

- Tracking objects can be complex due to complex object shapes, object motion, non-rigid nature of object, scene illumination changes, partial or full object occlusions, etc.

Step-3 Object tracking

Tracking procedures are mainly divided into four types



Real-Time Object Detection and Tracking for Unmanned Aerial Vehicles Based on Convolutional Neural Networks

- In this work, the target object for detection is a person.
- In this study, utilize the ROS (Robot Operating System) to implement image detection and tracking for controlling UAVs.
- Hardware required laptop, lightweight models.
- For the object detector, train a convolutional neural network based on the YOLOv4 architecture.

- This study employ the pruned version of the YOLOv4 object detector and the SiamMask monocular object tracker to detect and track the target person captured by the camera of the drone.

- This object detection system consists of four main components:

I. Object detection

II. Target tracking

III. Proportional Integral Derivative (PID) control

IV. The UAV driver package

- This study utilize the Tello drone for implementing the object detection and tracking system.
- During the tracking process, the UAV control parameters include the roll, pitch, yaw, and altitude, all of which are controlled using PID controllers.
- These PID controllers take the position and distance of the target object as inputs.
- The position and distance are calculated using the monocular front-facing camera of the UAV.

Approach

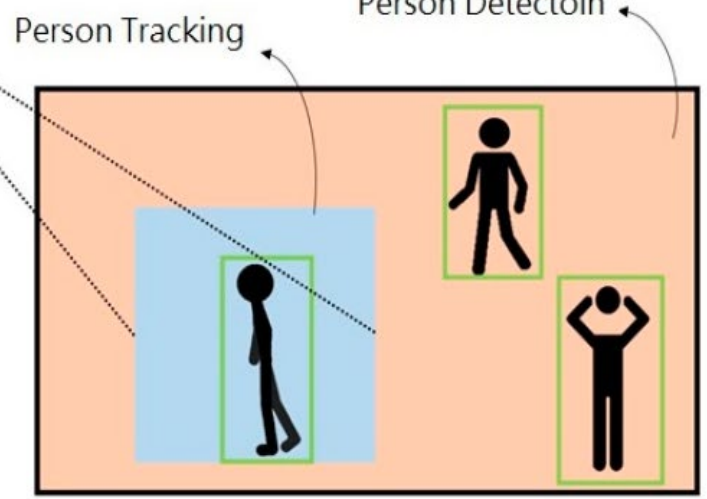
Introduction to Framework Methods

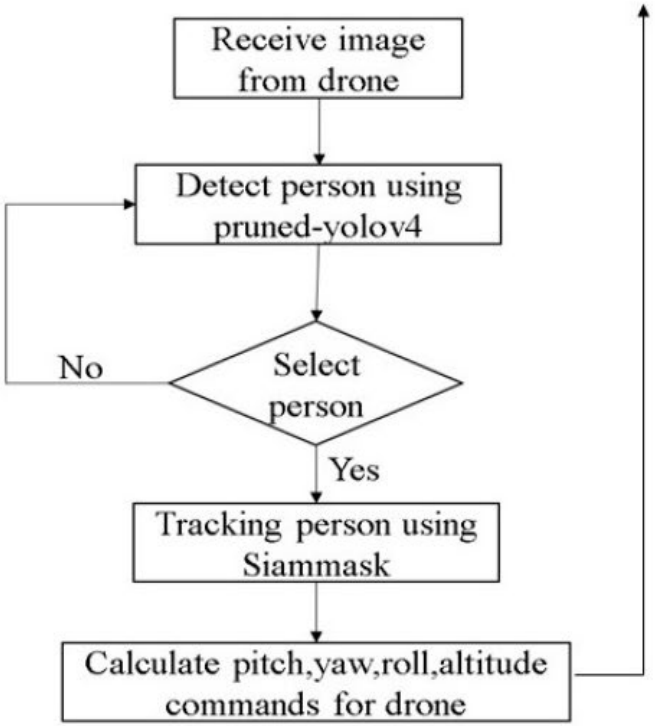
- **Proposed framework- object detection, model pruning, and visual tracking.**

System Setup and Communication

- **A laptop computer (PC) communicates with the Tello drone via Wi-Fi.**
- **The drone transmits images at a constant frequency of 30 Hz, processed using a pruned version of the YOLOv4 algorithm for object detection.**


Drone
Laptop


Person Tracking
Person Detectoin


Receive image from drone
Detect person using pruned-yolov4
Select person
No
Yes
Tracking person using Siammask
Calculate pitch,yaw,roll,altitude commands for drone

Object Detection and User Interaction

- Users can select bounding boxes based on their requirements for object detection.
- Pruned-YOLOv4 is utilized for person detection, with detected bounding boxes displayed on the screen.

Object Tracking with SiamMask

- The system employs the Siamese network, SiamMask, for object tracking.
- A tracking algorithm based on a PID controller estimates roll, pitch, yaw, and altitude based on the tracked object's position and distance.

User Interaction and Object Selection

- Users can select a specific object of interest by clicking on its bounding box.
- The system extracts the person within that bounding box as a template frame for the SiamMask network to enable subsequent tracking.

Flight Commands Generation

- The tracking algorithm calculates the error between the target and the center of the frame.
- This error serves as input for the PID controller to generate flight commands for yaw, roll, and altitude adjustments.

Handling No Target Detection

- If no target is detected, the drone maintains its position until a target appears in the image feed.
- This ensures stability and prevents unnecessary movements when no object of interest is present.

Hardware Specifications

Introduction to DJI Tello Drone

- The DJI Tello drone is a small and easy-to-control consumer-grade drone suitable for both indoor and outdoor use.
- It has dimensions of approximately 98 × 92 cm and weighs around 80 g.

Drone Features

- Equipped with various sensors including a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis magnetometer, a pressure sensor, and an ultrasonic altitude sensor.
- Features a front-facing camera with a resolution of 1280 × 720, capable of capturing video at 30 frames per second.

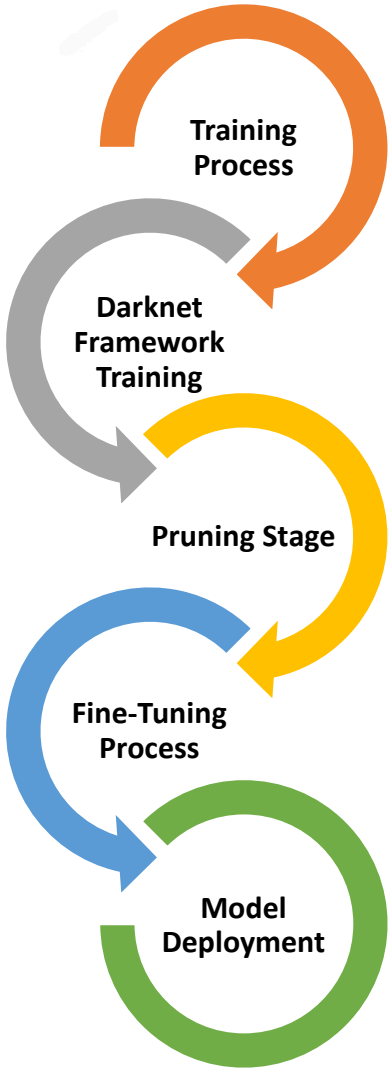
Communication and Connectivity

- The Tello drone can communicate with other devices such as smartphones or laptops via a Wi-Fi network.

Usage in Study

- In this particular study, a PC was used for communication with the Tello drone.
- This setup allows for data exchange and control of the drone's functions during experiments or applications.

Detection Model Pruning and Object Detection



The object detection process begins with the training of the model

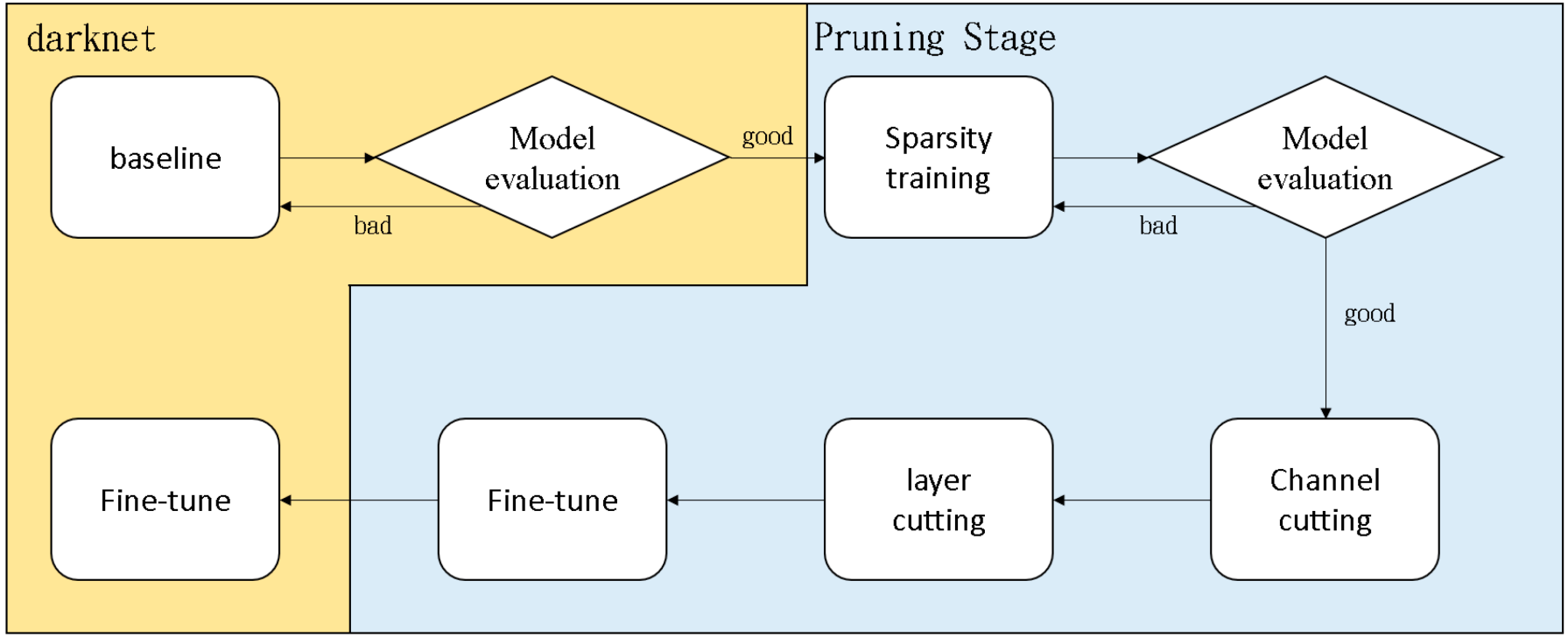
Initially, the Darknet framework is employed to train the YOLOv4 base model

Various techniques such as sparse training, channel pruning, layer pruning, and fine-tuning are applied to the base model using the Darknet framework.

Fine-tuning helps in optimizing the model's performance and accuracy further.

This deployment allows for real-world applications of the trained model in detecting objects efficiently.

Detection Model Pruning and Object Detection



Darknet Training

Training with Darknet Framework and YOLOv4

- The Darknet framework is utilized for training the YOLOv4 model, with adjustments made to various hyperparameters to enhance accuracy and performance.

Adjusting Input Size

- One crucial hyperparameter adjusted is the input size of the network, impacting the model's ability to detect small objects.
- Increasing the input size aids in detecting small objects but may slow down inference speed and consume more GPU memory.

- The YOLOv4 network downsamples the input size by a factor of 32, necessitating input width and height to be multiples of 32.
- In this study, the input size is set to 416 × 416 to ensure compatibility with the network.

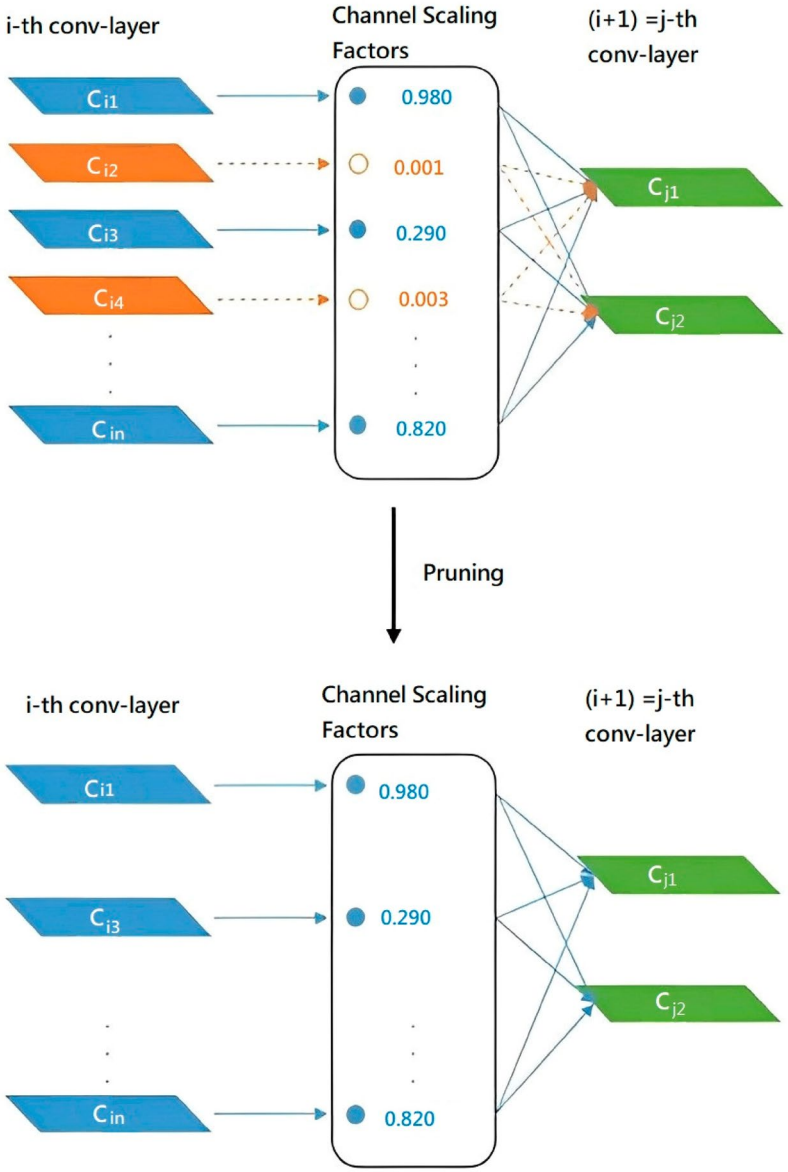
Batch Size and Subdivisions

- The batch size and subdivisions hyperparameters are adjusted based on GPU performance.
- Batch size represents the number of images loaded during training, typically set to 64.
- If GPU memory is insufficient, each batch is subdivided into smaller sub-batches to fit into memory.

- In this study, the batch size is set to 64, with subdivisions set to 8, optimizing GPU memory usage.

Number of Iterations

- Training in the Darknet framework is measured in iterations, not epochs.
- Each object class should ideally have at least 2000 iterations for effective training.
- With only one class, the number of iterations is set to 2200 to ensure sufficient training for higher accuracy and performance.



Pruning Stage

Basic Training and Pruning Strategy

- Before pruning, weights from the Darknet framework undergo basic training.
- The pruning strategy is employed for model pruning, focusing on achieving channel sparsity in deep models.

Sparse Training and Channel Pruning

- Sparse training is conducted on the model, with L1 regularization applied to scaling factors associated with each channel in convolutional layers.

- This regularization helps identify unimportant channels, which are subsequently pruned based on their scaling factor values.

Compact Model Generation

- After pruning, a compact model is obtained, potentially sacrificing some less important channels for reduced model size and complexity.
- This compact model is then fine-tuned to achieve comparable or even higher accuracy compared to the fully trained network.

Sparsity Training

Introduction to Batch Normalization (BN) Layer

A Batch Normalization (BN) layer is added after each convolutional layer in YOLOv4 to expedite convergence and enhance generalization.

Normalization Process

The BN layer normalizes convolutional features using batch statistics, represented by the equation

$$y = \gamma \times \frac{x - \bar{x}}{\sqrt{\sigma^2 + \varepsilon}} + \beta$$

Here, $\bar{X}$ and $\sigma 2$ represent the mean and variance of the input features in the mini-batch, respectively. γ and β represent the trainable scale factor and bias in the BN layer.

Indicator of Channel Importance

The scale factor (γ) in the BN layer is utilized as an indicator of channel importance. L1 regularization is applied to γ to facilitate channel-level sparse training, distinguishing between important and unimportant channels effectively.

Sparse Training Loss Function

The loss function for sparse training incorporates L1 regularization on γ and is expressed as

$$L = loss_{yolo} + \alpha \sum_{\gamma \in \Gamma} f(\gamma)$$

Here, f(γ) represents the L1 norm applied to γ, and α is the penalty factor balancing the two loss terms.

Benefits of Sparse Training

- **Pruning effectiveness relies on the sparsity of the model.**
- **Sparse training compresses most γ values in the BN layer towards zero, leading to two benefits:**

    1) **Improved model efficiency through network pruning and compression, reducing computational complexity.**
    2) **Identification and pruning of parameters with minimal impact on network performance by sparsifying weights close to zero**

Channel cutting

- Following the completion of sparse training, the process of channel cutting is initiated to further optimize the model.
- The total number of channels in the backbone is computed to establish the basis for channel cutting.
- Corresponding γ values are extracted and stored in a variable, then sorted in ascending order.
- The decision of which channels to retain and which to prune is based on a predefined pruning rate.
- The pruning rate, typically a value between 0 and 1, determines the proportion of channels to be pruned. A higher pruning rate signifies a greater degree of pruning.

Fine-tuning

Necessity of Fine-Tuning

- In cases where pruning adversely affects model accuracy, fine-tuning becomes essential to restore the pruned model's accuracy.
- Fine-tuning plays a crucial role in mitigating accuracy loss caused by pruning, thereby enhancing the overall performance of the pruned model.

Importance of Fine-Tuning

- Fine-tuning allows for the adjustment of model parameters and optimization of the pruned model for improved accuracy.

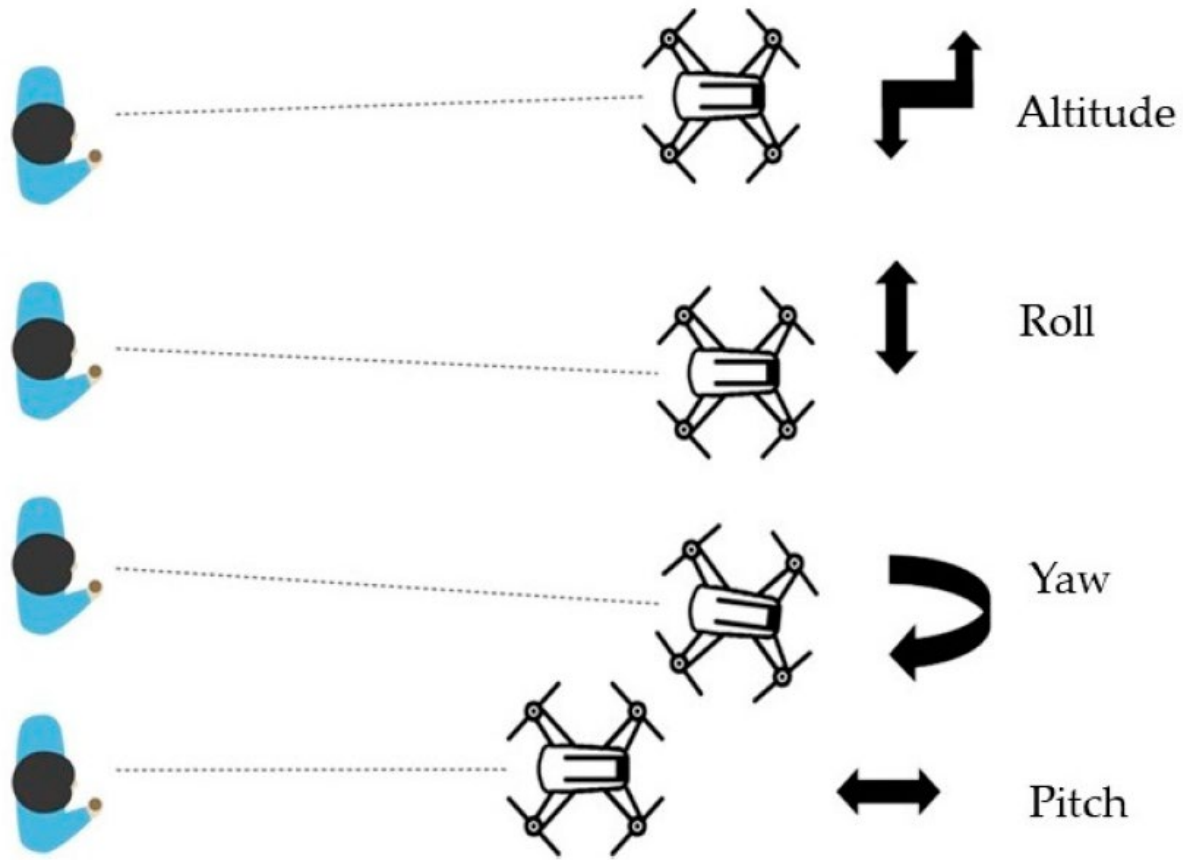
- It ensures that the pruned model maintains its effectiveness in performing its intended task, such as object detection.

- **Experimental Approach**

- In the conducted experiments, the Pruned-YOLOv4 model was retrained using the same training hyperparameters as the normal training process for YOLOv4.

- This approach ensures consistency and facilitates comparison between the pruned and original models, enabling a comprehensive evaluation of their performance.

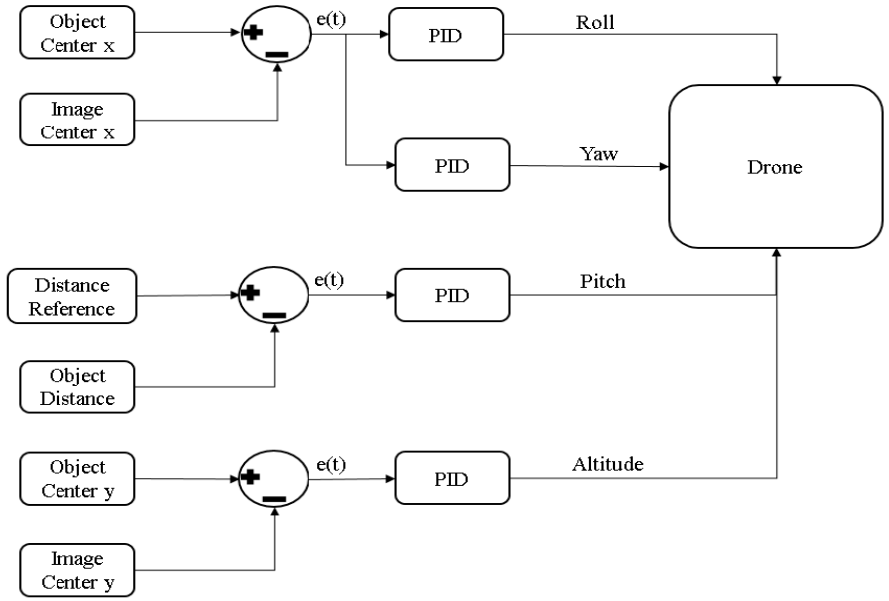
Fundamental maneuvers of a drone



Control scheme for the drone

Error Calculation for X-Axis

- By comparing the center point of the tracked object with the center point of the screen, the error in the X-axis is determined.
- This error corresponds to the drone's roll for lateral movement and yaw for clockwise or counterclockwise rotation.



X-Axis Error Handling

If the drone detects lateral movement of the object, adjustments can be made to the drone's heading to face the object or perform lateral movements to maintain alignment.

Error Calculation for Pitch Axis

- The pitch axis involves forward and backward movements.
- By comparing the distance between the drone and the real object with the desired ideal distance, the distance error is calculated to control the drone's forward or backward movements accordingly.

Pitch Axis Error Handling

Adjustments are made based on the calculated distance error to control the drone's forward or backward movements and maintain desired proximity to the tracked object.

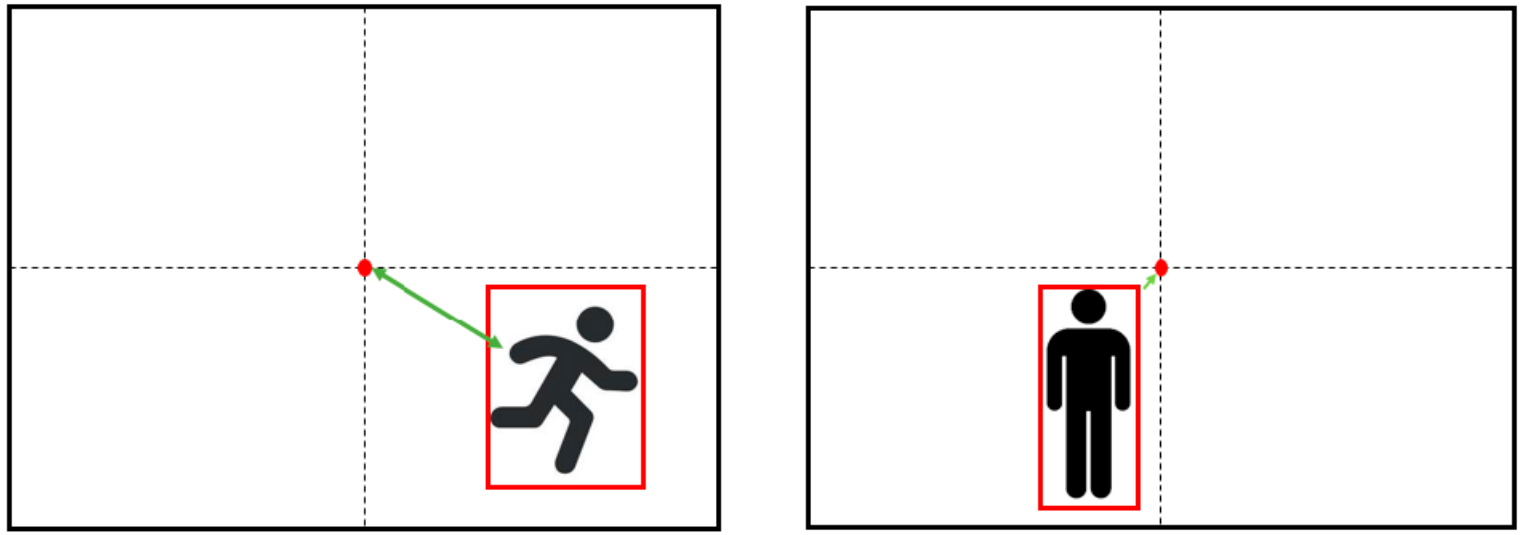
Error Calculation for Y-Axis

- The error in the Y-axis is obtained by comparing the Y-coordinate of the tracked object with the Y-coordinate of the screen center.
- This error is used to calculate the necessary altitude adjustments for the drone's vertical ascent or descent

Y-Axis Error Handling

Altitude adjustments are made based on the calculated error to control the drone's vertical movement and maintain the desired altitude relative to the tracked object.

If the tracked object does not exhibit significant movement, the yaw option is selected, which only requires adjusting the drone's heading to follow the target, represented by the green arrow illustrated in Figure.



Examples of object movement (a) Roll (b) Yaw

Detection Model Pruning and Object Detection

- This study trained the baseline YOLOv4 model using the coco2014 dataset.
- The Tiny version of YOLOv4 is specifically designed as a lightweight variant for devices with lower computational resources.
- To achieve this goal, a series of experiments to simulate the exploration needs of drones in real environments and require the drones to successfully track target objects automatically.

To evaluate the performance of the object detector, we applied the following four metrics:

(1)Precision:

- It measures the proportion of true positives among all the detections made by the system. A higher precision indicates that the system can accurately identify target objects, reducing the likelihood of false alarms.

(2)Recall:

- It measures the proportion of true positives among all the actual target objects. A higher recall indicates that the system can successfully detect a larger portion of the target objects, reducing the risk of missed detections.

(3)BFLOPs

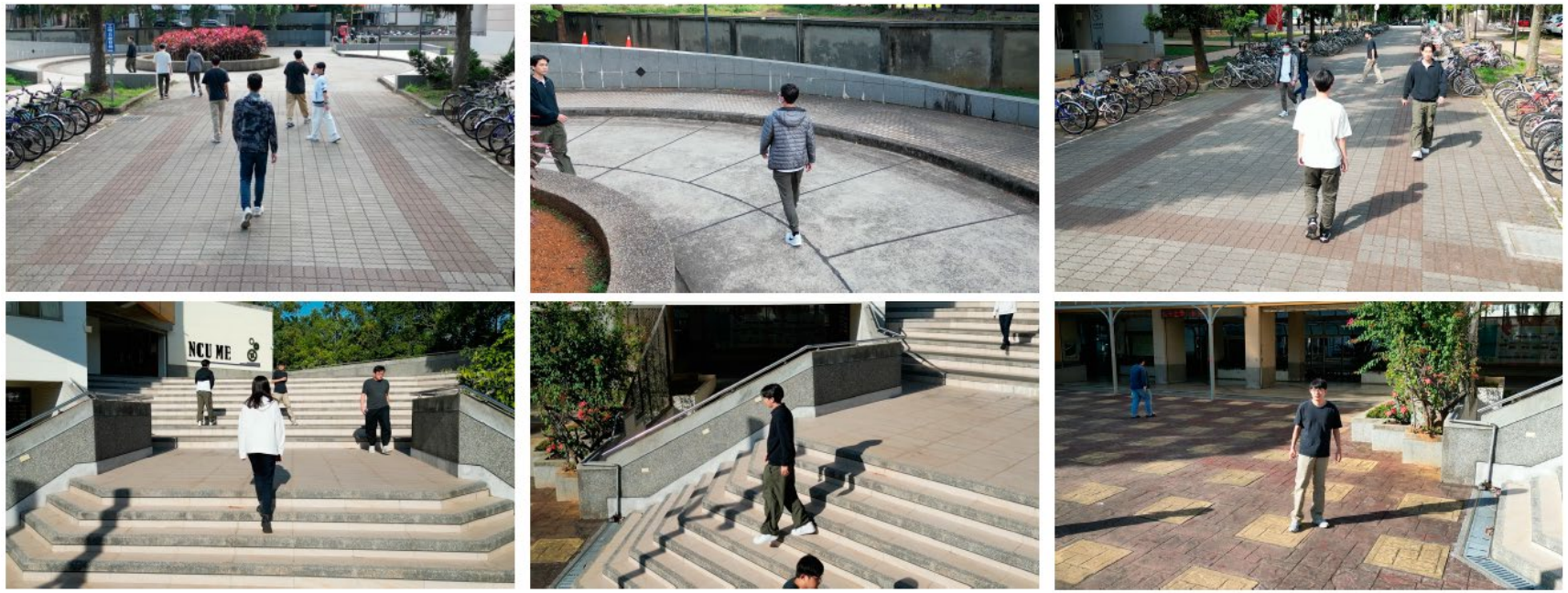
- **BFLOPs is a metric used to measure the computational efficiency of a computer system or machine-learning model. It is a commonly used metric for evaluating the computational efficiency and speed of systems or models.**

(4) mAP@0.5 (mean Average Precision at IoU 0.5)

- **mAP@0.5 is a commonly used evaluation metric in object detection. It measures the average precision at an Intersection over Union (IoU) threshold of 0.5 across different classes.**

Subject Tracking and Drone Control

- Figures below are the selected outdoor scenes and indoor scenes in the experimental videos, respectively.
- The subjects being tracked include ten different people.
- Each person is tracked for 50 to 90 s in outdoor and indoor environments five times.
- During the tracking process, a random number of 0 to 7 other people would appear as passersby in the scene.



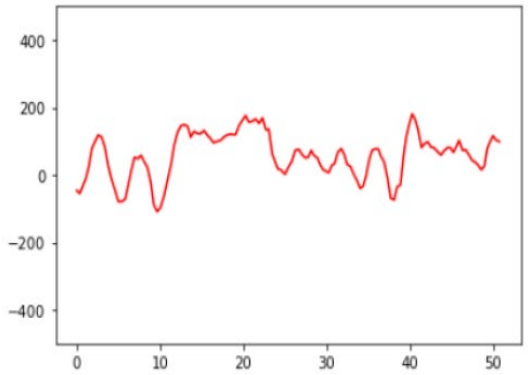
Selected scenes in outdoor environments

Selected scenes in indoor environments.

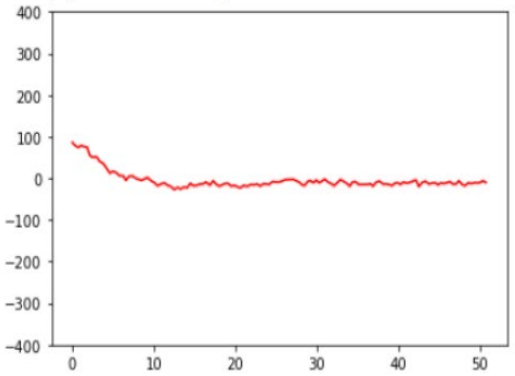
B. Analysis of PID control for drone

- In the video-1, the tracked subject walks on a flat surface, as shown in figure (a). The four directions that the subject moves are represented as the red, blue, yellow, and purple arrows in figure (a)

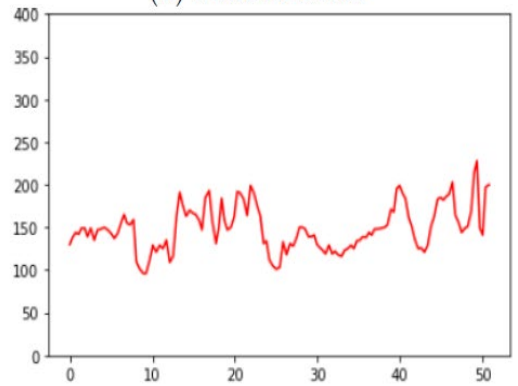
(a) Scene of experimental video 1



(b) Error in x-axis



(d) Error in y-axis



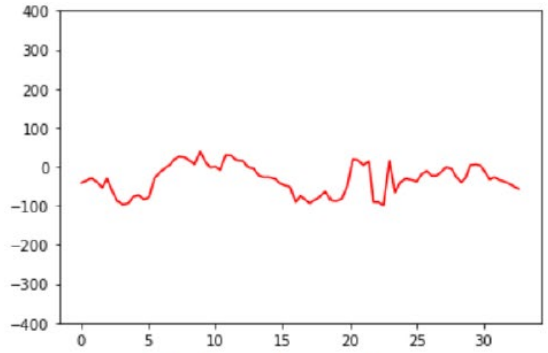
(c) Distance in z-axis

- The response of the PID control to the error in the *x*-axis position of the tracked object in video-1 is plotted in figure (b).
- For the error in *y*-axis, since the subject in video-1 does not undergo significant changes in height, we can observe from Figure (c) that there is not a significant variation in the y-axis error.
- Figure (d) plots the response of the PID control to the error in the *z*-axis position of the tracked object in the selected video.

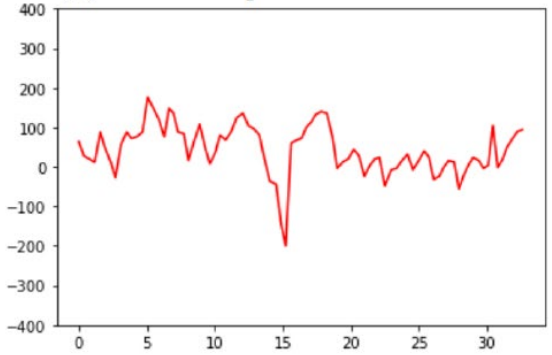
Response of the PID control to the errors in different directions for video-1.

- In video 2, the tracked subject moves upstairs and downstairs, as shown in Figure (a).
- The red arrow and blue arrow represent the directions moving up and down the stairs.
- Figure a–c show the response of the PID control to the error in the x-axis and y-axis positions and the distance in the z-axis position of the tracked object in video 2.

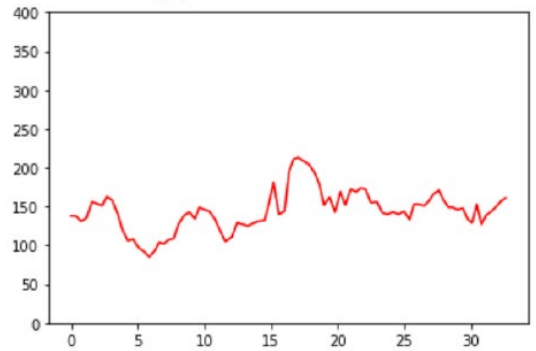
(a) Scene of experimental video 2



(b) Error in x-axis



(d) Error in y-axis



(c) Distance in z-axis

Figure demonstrates that the PID control continuously adjusts the x-axis and y-axis errors to approach zero as the subject moves forward and backward while ascending or descending the stairs.

Response of the PID control to the errors in different directions for video 2.

REFERENCES

- Yang, S. Y., Cheng, H. Y., & Yu, C. C. (2023). Real-Time Object Detection and Tracking for Unmanned Aerial Vehicles Based on Convolutional Neural Networks. Electronics, 12(24), 4928. https://doi.org/10.3390/electronics12244928

CONCLUSION

- Real-time human tracking and detection can be possible using the image process and drone technology.
- The technology not only improves the quality control measures but also contributes to cost-effectiveness and sustainability in industrial operations.


THANK YOU