Detection of Small and Rare Objects

submitted in partial fulfillment of the requirements

for the degree of

MASTER OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

by

VIJAY RAJ   EE22M215

Supervisor(s)

RAMA KRISHNA GORTHI

भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

DECEMBER 2023

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission to the best of my knowledge. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 8-12-2023

Signature
Vijay Raj
EE22M215

BONA FIDE CERTIFICATE

This is to certify that the report titled **Detection of Small and Rare Objects**, submitted by **Vijay Raj**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Master of Technology,** is a bona fide record of the project work done by him under our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 8-12-2023

Professor Name here
Guide
Dr. Rama Krishna Gorthi
Department of Electrical Engineering
IIT Tirupati - 517501

ACKNOWLEDGMENTS

I extend my heartfelt gratitude to my advisor, **Dr. Ramakrishna Gorthi**, for unwavering support, invaluable guidance, and continuous motivation during my MTech journey. His vast knowledge, patience, and keen observational skills have fostered my out-of-the-box thinking and optimism towards research. Dr. Gorthi's moral support, coupled with his kind and friendly nature, proved to be a constant source of strength during both professional and personal challenges.

I also want to express my appreciation to **Arun Kumar Sivapuram** PHD, IIT Tirupati .His support, guidance, collaboration, and the provision of datasets have been instrumental.

I am grateful to **Prof. K. N. Satyanarayana, Director, IIT Tirupati**, for providing the necessary facilities and overcoming obstacles to facilitate research in the newly established campus.

Lastly, I extend my thanks to my family, friends, and colleagues for their unwavering support throughout my academic journey

ABSTRACT

KEYWORDS: Data Augmentation,Object Detection,Object Tracking;

In the realm of surveillance and security, Anti-Unmanned Aerial Vehicle (Anti-UAV) detection and tracking play crucial roles in safeguarding sensitive areas, preserving data integrity, ensuring privacy in research zones, preventing disruptions in environmental studies, and enhancing safety measures in hazardous environments.

Traditional vision-based UAV detection and tracking systems face challenges in generalization due to limited data availability and the laborious annotation process. This study introduces a novel self-annotated framework, combining state-of-the-art tracking methods, sample selection approaches, and an image composition procedure to generate a multi-drone detection and tracking dataset using publicly available single-drone videos.

SiamMask, a mask-based tracker, is employed to extract drone instances, segmentation masks, and pseudo-detection labels in each frame of the single drone videos. The proposed image composition method generates a multi-drone tracking dataset using the obtained pseudo-labeled tracks. The framework yields the first Multi-Drone Tracking (MDT) dataset, comprising 20 multi-drone videos with approximately 70,000 self-annotated detection samples.

To assess the performance of detection and tracking on the introduced MDT dataset, the Yolov8 model is employed for detection, while ByteTrack and Strong-SORT algorithms are utilized for tracking. Experimental results demonstrate that the trained detection model achieves precision of 0.964 and 0.719 on seen and unseen MDT datasets, respectively. Strong-SORT exhibits tracking capabilities with a MOTA of 93.0

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Imagine your eyes are like a smart camera. It can not only see things but also figure out what they are and where they are. That's what we call object detection. It's like your eyes recognizing and locating stuff. Now, think about spotting drones, those flying gadgets. It's a bit tricky because there aren't many examples of drones around for your eyes (or the camera) to learn from. Why? Well, flying a drone usually needs permission, so you don't see them everywhere.

But, finding drones is important, especially for safety. So, even though it's a bit tough, making sure our 'smart cameras' can recognize and find drones is like giving our eyes an extra superpower to keep us safe when these flying gadgets are around.

Finding drones using Computer vision and deep learning is a bit like teaching a computer to recognize things in pictures and videos – this is what we call object detection. Object detection has two main jobs. First, it needs to figure out exactly where the thing we're interested in (like a drone) is in the picture or video. We call this finding the coordinates or location. Second, it needs to say, "Hey, that's a drone!" This part is like naming or classifying the object.

But here's the challenge: for the computer to learn this game well, it needs lots of practice, just like us humans. We call this practice data, and right now, there aren't many examples of drones for the computer to learn from. So, our first big task is to create a bunch of examples, or what we call Datasets, specifically focused on detecting multiple drones. Think of it like giving the computer a good set of pictures to learn from. This step is crucial because it's like laying the groundwork for the computer to become really good at spotting drones accurately. Once we have these Datasets, our computer will be much better at figuring out if there's a drone in the picture or video it's looking at.

1.1 Objectives and Scope

- Self-annotation of Drone Datasets using State-of-the-Art Trackers.
- Apply Object Detection on Generated Datasets.
- Addressing Imbalanced Classes in Detection Problems.

The main goal is to Create Drone Datasets efficiently by leveraging SiamMask tracker. Instead of manually annotating each image, we'll annotate one image ourselves, and then let the SiamMask tracker handle the annotation for the rest by finding and tracking the drones in the images. An image composition method is proposed to generate multi-drone tracking Dataset with the obtained pseudo-labelled tracks from the videos.

CHAPTER 2

Literature Review

2.1 Object Detection

The problem statement for Object Detection involves identifying both the locations of objects and the corresponding classes to which they belong. To achieve this objective, any object detection method, whether utilizing Deep Learning or less complex approaches, can be outlined in three key steps:

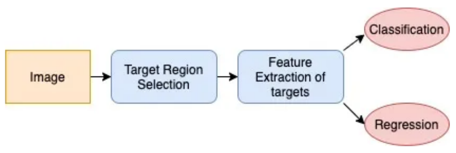

Figure 2.1: An illustrative image related to Object Detection. Source: Sik-Ho Tsang

2.1.1 Target Region Selection

In conventional methods, region selection primarily involves employing a brute-force sliding window technique on an image. The sliding window, characterized by a predetermined size and shape, traverses the image, capturing crops along the way.

It might seem like a straightforward solution – just slide a window over the image and, voila, problem solved. But, in reality, it's not that simple, especially for more general cases. Imagine dealing with various objects that come in different shapes, sizes, and positions within an image. Trying to find an ideal window for each object becomes a real headache – it's computationally intensive and ends up generating way too many unnecessary windows. This, in turn, slows down the whole process. Now, what if we decide to use a fixed set of window sizes and templates? It could save some time, but there's a catch – it won't account for the same object appearing at different scales. So, we'd essentially find ourselves stuck in a recurring problem loop.

2.1.2 Feature Extraction

Once we've got those image snippets from the previous step, we need to dive in and understand the semantic and visual essence of each object. This is where good old feature descriptors, both local and global, come into play. We're talking about techniques like SIFT, HoGKachouane *et al.* (2012), and Harr-Like features, which can be adjusted to work well for different objects. Sounds promising, right? Well, here's the catch – because objects can look different due to noise, scale, lighting, or even hiding behind something else (occlusion), manually tweaking these descriptors for each object becomes a real headache

2.1.3 Classification/Regression

In the final step of our process, we classify the identified image snippets using the feature descriptor values we obtained earlier. This involves assigning each snippet to a specific class and drawing a bounding box around the corresponding object in the image. Commonly used classification techniques include Support Vector Machines, AdaBoost, and Random Forest. However, these models demand substantial information about each class, requiring intricate adjustments to yield satisfactory results. For instance, Support Vector Machines face challenges in multi-class classification due to limited support for class probability discrimination. Moreover, these methods struggle with generalizing data, particularly in scenarios involving noise and overlapping data points.

2.1.4 Deep Learning Object Detection Model

CNNs and deep neural networks have made traditional object detection methods more reliable. With access to massive datasets and deeper structures, these networks automatically learn complex features, addressing gaps in feature extraction. Extensive training approaches eliminate the need for manual feature learning per object, ushering in a shift towards deeper architectures for a holistic, end-to-end approach to object detection.
In Deep learning there are two types of detectors
1.Single-Stage Detector: Provides both classification and bounding box information

simultaneously, as seen in YOLO.
2.Two-Stage Detector: Initially identifies the bounding box and then performs classification.
I have used YOLO for Object Detection as it is having high MAP as well as it is fast and light weight model so we can deploy it in Raspberry-Pi.

2.1.5 YOLO (YOU LOOK ONLY ONCE)

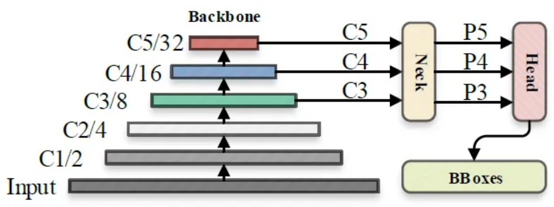

Figure 2.2: YOLO Architecture
Source: Sik-Ho Tsang

First, the image goes through an input layer and then to the backbone for feature extraction. The backbone combines features of different sizes through a feature fusion network to create three maps (P3, P4, and P5 in YOLOv5) for spotting small, medium, and large objects in the image.
These maps are then sent to the prediction head, where each pixel undergoes confidence calculation and bounding-box regression using preset anchor values. The result is a multi-dimensional array (BBoxes) containing details like object class, confidence level, box coordinates, width, and height.
To refine the information, we set specific thresholds and perform a non-maximum suppression process, filtering out irrelevant details. The final output provides us with the accurate detection information we need.

YOLO Backbone

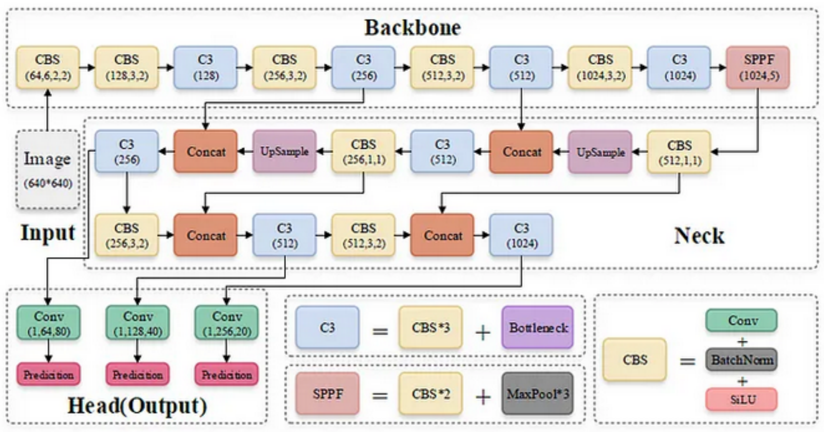

Figure 2.3: Backbone
Source: Sik-Ho Tsang

- The backbone is CSPDarknet53
- The primary structure involves stacking multiple CBS (Conv + BatchNorm + SiLU) modules and C3 modules. The sequence is capped off with the connection of a single SPPF module.
- CBS module is used to assist C3 module in feature extraction, while SPPF module enhances the feature expression ability of the backbone.
- SPPF eliminates redundancy by max-pooling previously max-pooled features, avoiding repetitive operations seen in SPPNet
- This can significantly improved the running speed of the module

YOLO Neck

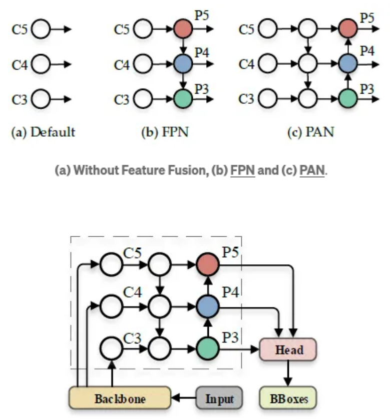

Figure 2.4: Neck
Source: Sik-Ho Tsang

- FPN's core concept involves taking the output feature maps (C3, C4, and C5) produced by various down-sampling operations in the feature extraction network and then up-sampling them.

- This process creates multiple new feature maps (P3, P4, and P5) specifically designed for detecting targets of different sizes.

YOLO Head

the "head" refers to the final part of the neural network architecture responsible for making predictions. It plays a crucial role in processing the features extracted by the backbone network and generating the final output.(Ultralytics, 2021)

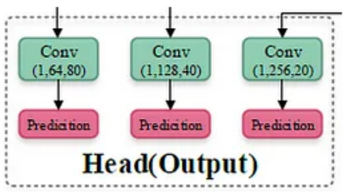

Figure 2.5: YOLO Head
Source: Sik-Ho Tsang

CHAPTER 3

Detection of small and Rare objects

3.1 Objects

Objects can be categorised as

- **Small Objects** : <32X32 Pixels
- **Medium Objects** : >32X32-96X96 Pixels
- **Large Objects** : >96X96 Pixels

Rare objects refer to items that are infrequently observed and have limited representation in datasets, making them less commonly encountered or documented.

3.2 Techniques to improve object detection

- **Dataset Generation and Training**
  Create specialized datasets with a mix of variations, encompassing different weather conditions and, notably, night vision visuals. This approach ensures the model becomes adept at object detection across a diverse set of environmental scenarios, enhancing its overall robustness.

- **Architectural Modifications**
  Modifying the architecture involves adjusting the structure, layers, or parameters of the model. This could include adding more layers, changing activation functions, or incorporating attention mechanisms to improve object detection performance

- **Dataset Generation and Training**
  Create specialized datasets with a mix of variations, encompassing different weather conditions and, notably, night vision visuals. This approach ensures the model becomes adept at object detection across a diverse set of environmental scenarios, enhancing its overall robustness.

- **Architectural Modifications**
  Modifying the architecture involves adjusting the structure, layers, or parameters of the model. This could include adding more layers, changing activation functions, or incorporating attention mechanisms to improve object detection performance

Right now, I am concentrating on creating datasets for the initial phase. This means we’re putting our efforts into making a diverse and comprehensive set of examples for the model to learn from

3.3 Challenges

- **Data Acquisition**
- **Limited Datasets- Only limited to very good weather Condition**
- **Need diversified samples in the dataset for better detection**
- **Class Imbalance**

3.4 Overcoming Datasets Limitations

Getting datasets of rare objects is tough because of privacy issues, making them not widely accessible. Yet, there's a silver lining – videos naturally show lots of different things, providing a chance to train models for real-life situations. But here's the catch – annotating the data is a real pain, it takes a lot of time and effort. To tackle this, a smart idea is using trackers that can follow any object. This way, the datasets can annotate themselves. And to make sure we get a good mix of examples, we use special methods to pick diverse samples. It's a cool way to handle the challenges when data is scarce and annotations are a headache.

3.5 Annotating Datasets with a Tracker

Drone objects in the first frame are localized in a drone video, and provided to the state-of-the-art single object trackers. These trackers were then used to determine the object's location in subsequent frames of any drone video.

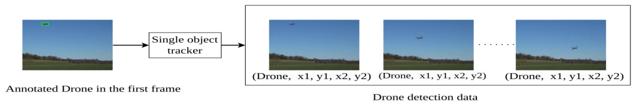

Figure 3.1: Self-labelling

3.6 Dataset Details

In this section, we provide an overview of the existing resources in the realm of vision-based drone detection and tracking. We also present details about the dataset we have generated. Fredrik's Svanstrom *et al.* (2020) dataset comprises real-time video clips captured in both day and night scenarios, utilizing high-resolution day cameras and thermal cameras. We have carefully selected 13,801 annotated training samples (ICPR), 3,451 annotated test samples (ICPR), and 106 visible drone videos (ICPR drone data) from the dataset for our experimental purposes. Our proposed framework generates 459 effective samples using Custom web data, and we identify this dataset as Custom Web Drone Video Tracks (CWDVT). Details regarding the distribution of training and test data from both existing and generated detection datasets are provided.

CWDVT and DUT test datasets are exclusively reserved for testing, categorizing them as unseen datasets in our experiments. We examine the distribution of synthetic drone samples generated from tracks in LaSOT (CLDTs) and Custom web videos (CWDVT), visualizing them in comparison with drone samples from other well-established datasets such as ICPR and DUT. The diversity observed in the generated synthetic drone samples serves as evidence of the effectiveness of our proposed approach.

Dataset Name	**No. of Training Samples**	**No. of Testing Samples**
ICPR	13,801	3,451
LaSOT	40,748	101,817
DUT	NA	2,208
CWDVT	NA	419
CLDT	1708	NA

Table 3.1: Dataset Information

3.6.1 Multi-drone tracking dataset details

The multi-drone tracking dataset was created by aggregating ICPR drone data and LaSOT videos. For annotating the single drone videos, SiamMask Wang *et al.* (2019) a mask-based tracker was employed by manually annotating the first frame. After extracting the pseudo-labeled masks from ICPR drone videos, the center of each track in the video sequence is calculated. With the obtained pseudo-labeled masks and their corresponding center coordinates, drones are composed into the LaSOT background drone video with the proposed image composition method as shown in Figure 2. Additionally, the composited drones are assigned the relevant tracker IDs for labeling. With this approach, to demonstrate the effectiveness of the proposed framework, 20 videos are generated which consist of 21818 frames, this dataset is referred to as the multi-drone tracking (MDT) dataset. Some of the parameter details of the generated MDT are given in Table 3.2. Some of the challenges.in this composed dataset encompass drones of varying sizes,including small drones as tiny as 10x10 pixels, significant drift in drone movement from frame to frame reflecting the very fast motion of the drones, and a background cluttered with various elements.

Parameter	No of videos	No of frames	Min WXH	MAX WXH	Backgrounds
Description	20	21,818	10x10	310x252	Night,Sky,Cloudy

Table 3.2: Details of the Multi-Drone Tracking (MDT) dataset.

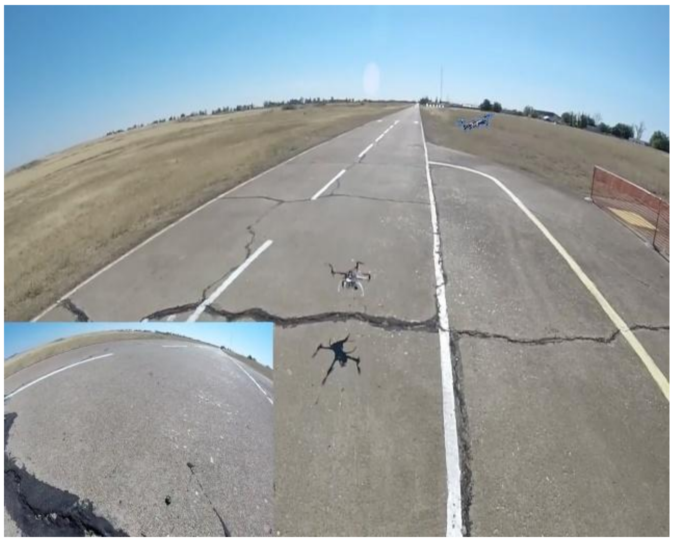

Figure 3.2: Composed Image

CHAPTER 4

Results and Discussions

To evaluate the effectiveness of the generated datasets, we employ the YoloV8 object detection model—a contemporary, high-performance method known for its speed and superior accuracy compared to other detectors. We fine-tune the Yolov8n model, pre-trained on the COCO dataset, using the Adam optimizer for 50 epochs. The fine-tuning process employs a batch size of 16, and we set other hyperparameters in alignment with the recommendations specified in the foundational paper of Yolov8n. During this fine-tuning process, we utilize various drone detection datasets as detailed in Table 3.1

For multi-object tracking, we employ two state-of-the-art methods, Byte-Track(BT)Zhang *et al.* (2022) and Strong-SORT(SS)Du *et al.* (2023), alongside the Yolov8n detection model fine-tuned using the respective training datasets outlined in Table . This comprehensive evaluation aims to leverage the strengths of the YoloV8 model in conjunction with cutting-edge multi-object tracking techniques.

Training Dataset	ICPR	DUT (Unseen)	LDT	CWDT	CMD
ICPR	0.954	0.107	0.123	0.505	0.108
LDT	0.148	0.826	0.984	0.911	0.304
ICPR plus CLDT	0.964	0.719	0.965	0.901	0.325
ICPR, CLDT, MDT	0.939	0.629	0.952	0.898	0.99

Table 4.1: Precision Results with YOLO V8 model

Detector	MOTA(BT)	MOTA(SS)	MOTP(BT)	MOTP(SS)
ICPR	0.07	-1.23	0.721	0.709
Lasot	0.697	0.756	0.683	0.658
ICPR and CLDT	0.6601	0.686	0.651	0.650
ICPR,Lasot,MDT	0.93	0.931	0.213	0.198

Table 4.2: MOT Performance with YOLO V8n Trained Model.

From the precision results of row 1 of table 4.1 , it is clearly evident that the model only trained on ICPR real data is completely failing on unseen datasets such as LaSOT Drone data and DUT-Test data. Row 3 and row 4 of table 4.1 show the efficacy of the self-annotated (LDT dataset).

The model trained with the self-annotated (LDT or ICPR Real, CLDT or ICPR Real, CLDT, MDT) datasets works on par for drone detection in seen and unseen datasets. We assessed the performance of state-of-the-art trackers, Byte-Track Zhang *et al.* (2022)and Strong-SORT, on our newly generated Multi-drone tracking dataset to benchmark their capabilities. Table V presents the performance metrics of Byte-Track [18] and Strong-SORT [19] with respect to the generated multi-drone tracking dataset. The metrics include the ratio of correctly identified detections over the average number of ground-truth and computed detections (IDF1), multi-object tracking accuracy (MOTA), and multi-object tracking precision (MOTP). MOTA and MOTP are inversely proportional to the Intersection Over Union (IoU) scores of the tracker predictions with ground truth [20]. For an ideal tracker, IDF1 is 1, MOTA is 1, and MOTP is 0.

The detector trained with a combination of existing and generated synthetic datasets outperforms the tracker compared to detectors trained with other datasets. The impact of detector performance is clearly evident in Table 4.2, demonstrating an improvement in tracker performance. Notably, MOTA and MOTP values indicate significant potential for further developments and underscore the challenging nature of the generated dataset.

CHAPTER 5

SUMMARY,CONCLUSIONS AND FUTURE WORK

5.1 Summary

In summary, our work introduces a new way to create datasets for finding and tracking drones. We use videos of drones that are already available online and make these datasets without needing much human effort. These datasets are useful for teaching and testing computer models that find and track drones.

Our method is important because it can be applied to areas like remote sensing, making it easier to find and track UAVs accurately. Our experiments show that when we use these datasets to teach computer models, they get much better at finding and tracking drones, even in situations they haven't seen before. This means our datasets are strong and work well in different scenarios.

5.2 Conclusions

The methodology employed for generating datasets in our approach can be extended to other datasets that are less readily available or present challenges in annotation.The composed datasets can serve a dual purpose by acting as data augmentation, contributing to the improvement of detection performance.

5.3 Future Work

- In the context of dealing with uncommon datasets, it's likely that we encounter imbalances. Our aim is to enhance detection performance, specifically addressing challenges posed by imbalanced datasets.

- Leveraging super-resolution techniques to enhance object detection.

- In the future, we are planning to enhance the MDT dataset using the proposed framework for other classes such as birds, airplanes, and helicopters, with multiple instances of each class as well for robust detection and tracking of flying objects

REFERENCES

1. **Y. Du**, **Z. Zhao**, **Y. Song**, **Y. Zhao**, **F. Su**, **T. Gong**, and **H. Meng** (2023). Strongsort: Make deepsort great again.

2. **M. Kachouane**, **S. Sahki**, **M. Lakrouf**, and **N. Ouadah**, Hog based fast human detection. *In 2012 24th International Conference on Microelectronics (ICM)*. IEEE, 2012. URL http://dx.doi.org/10.1109/ICM.2012.6471380.

3. **F. Svanstrom**, **C. Englund**, and **F. Alonso-Fernandez** (2020). Real-time drone detection and tracking with visible, thermal and acoustic sensors.

4. **Ultralytics** (2021). YOLOv5: A state-of-the-art real-time object detection system. https://docs.ultralytics.com. Accessed: insert date here.

5. **Q. Wang**, **L. Zhang**, **L. Bertinetto**, **W. Hu**, and **P. H. S. Torr** (2019). Fast online object tracking and segmentation: A unifying approach.

6. **Y. Zhang**, **P. Sun**, **Y. Jiang**, **D. Yu**, **F. Weng**, **Z. Yuan**, **P. Luo**, **W. Liu**, and **X. Wang** (2022). Bytetrack: Multi-object tracking by associating every detection box.