EFFICIENT MULTI-OBJECT TRACKING ON UNMANNED AERIAL VEHICLE

by

Xiao Hu

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Electrical and Computer Engineering

School of Electrical and Computer Engineering

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Yung-Hsiang Lu, Chair

School of Electrical and Computer Engineering

Dr. Qiu Qiang

School of Electrical and Computer Engineering

Dr. James Davis

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

To Swanna Yu, who has been a great support for me to overcome difficulties.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Prof. Yung-Hsiang Lu for the continuous support of my M.S. studies and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my M.S. studies. My sincere thanks also go to my thesis committee: Dr. Qiu Qiang and Dr. James Davis, for their insightful comments and encouragement, but also the hard questions which incent me to widen my research from various perspectives.

I want to thank the Drone Video team at the Purdue CAM2 Research group for the preparation work of the 2021 Low-Power Computer Vision Challenge UAV Video track. My thanks are especially devoted to the team leader Ayden Kocher and Ziteng Jiao, who have led the team to collect testing video, develop sample solutions, and maintain a problem-free online submission period. The competition could not be so successful without their help.

I thank my fellow lab mates, Ahbinav Goel, Nick Eliopoulos, and Caleb Tung, for the stimulating discussions, for the comments and advice on my papers, and for all the fun we had in the last few years.

Lastly, I thank my parents and my friends for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS & ACRONYMS

DeepSORT	Deep Simple Online Realtime Tracking
Float32	32-bit floating-point
Float16	16-bit floating-point
GPU	Graphics Processing Unit
mAP	mean Average Precision
MOT	Multi-object Tracking
IoU	Intersection over Union
POI	Premature ovarian insufficiency
R-FCN	Region-based Fully Convolutional Network
RPN	Region Proposal Network
SDP	Scalable Real-time Dynamic Graph Partitioner
SSD	Single-shot Detector
SORT	Simple Online Realtime Tracking
UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once

ABSTRACT

Multi-object tracking has been well studied in the field of computer vision. Meanwhile, with the advancement of the Unmanned Aerial Vehicles (UAV) technology, the flexibility and accessibility of UAV draws research attention to deploy multi-object tracking on UAV. The conventional solutions usually adapt using the ”tracking-by-detection” paradigm. Such a paradigm has the structure where tracking is achieved through detecting objects in consecutive frames and then associating them with re-identification. However, the dynamic background, crowded small objects, and limited computational resources make multi-object tracking on UAV more challenging. Providing energy-efficient multi-object tracking solutions on the drone-captured video is critically demanded by the research community.

To stimulate innovation in both industry and academia, we organized the 2021 Low-Power Computer Vision Challenge with a UAV Video track focusing on multi-class multi-object tracking with customized UAV video. This thesis analyzes the qualified submissions of 17 different teams and provides a detailed analysis of the best solution. Methods and future directions for energy-efficient AI and computer vision research are discussed. The solutions and insights presented in this thesis are expected to facilitate future research and applications in the field of low-power vision on UAV.

With the knowledge gathered from the submissions, an optical flow oriented multi-object tracking framework, named OF-MOT, is proposed to address the similar problem with a more realistic drone-captured video dataset. OF-MOT uses the motion information of each detected object of the previous frame to detect the current frame, then applies a customized object tracker using the motion information to associate the detected instances. OF-MOT is evaluated on a drone-captured video dataset and achieves 24 FPS with 17% accuracy on a modern GPU Titan X, showing that the optical flow can effectively improve the multi-object tracking.

Both competition results analysis and OF-MOT provide insights or experiment results regarding deploying multi-object tracking on UAV. We hope these findings will facilitate future research and applications in the field of UAV vision.

1. INTRODUCTION

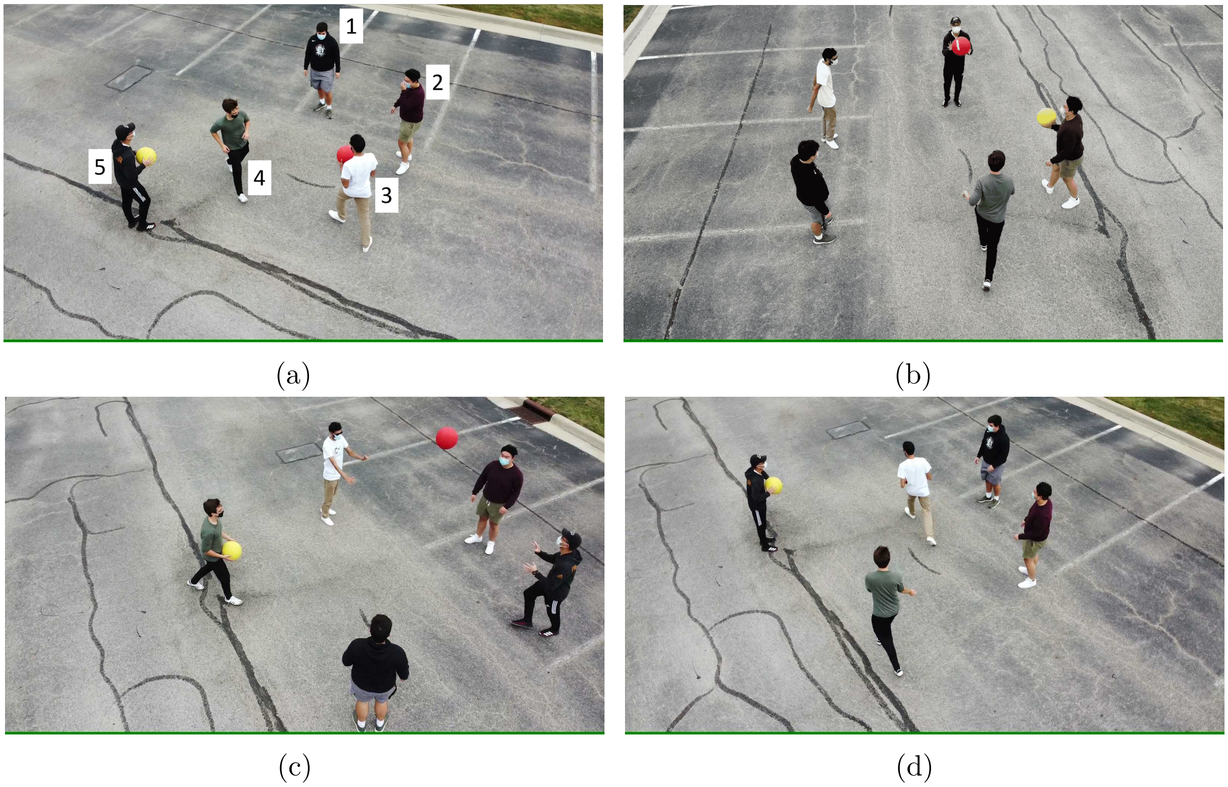
1.1 Multi-object Tracking for Unmanned Aerial Vehicle

Multi-Object Tracking (MOT), as a long-standing Computer Vision problem, aims to determine the identities and trajectories of multiple moving objects in a video. After years of development in both research and industry fields, many applications, such as video surveillance, autonomous cars, and crowd behavior analysis, benefit from high-quality tracking algorithms. At the same time, Unmanned Aerial Vehicles (UAV) attract increasing research attention in recent years due to its dynamic flexibility and convenience. Video that could not be taken from the ground can now be easily captured from the sky. These drone-captured video data provide a new perspective on the multi-object tracking tasks. Therefore, there have been a substantial number of studies devoted to tracking objects in UAV-captured videos to potentially improve aerial robot automation [1]–[7].

The standard approach used in multi-object tracking processes is instance association as a graph-based optimization problem under the "tracking-by-detection" paradigm: the detected objects in the current frame are re-identified based on the feature similarities of the detected objects in the previous frames [8], [9]. Researchers extend the prior achievement and knowledge of object detection and classification have been However, the sequences captured by UAV have attributes different from sequences captured by static cameras, resulting in a more challenging multi-object tracking problem. Although the traditional approach of combining object detection and re-identification (Re-ID) is accurate, it requires a significant amount of computational resources, especially when the objects are crowded and small in the frame. When deployed on drones, difficulties can arise from 1) the dynamic background that changes rapidly, 2) objects are small and crowded, and 3) the available energy is limited.

1.2 Contribution Summary

The contribution of this thesis can be summarized into 2 parts: 1) organizing the 2021 Low-Power Computer Vision Challenge (LPCVC) UAV Video track (Chapter 3), 2) providing



(a)

(b)

(c)

(d)

Figure 1.1. Each person (labeled 1-5) is given a number while moving and passing the balls (a, b). UAV will keep moving (c) and occlusion (the red ball) is possible (d).

an optical flow oriented multi-object tracking solution on drone-captured videos (Chapter 4).

1.2.1 2021 LPCVC UAV Video Track

To achieve a better understanding of the performance of multi-object tracking on drones, we organize a multi-object tracking problem as the task of 2021 LPCVC UAV Video track sponsored by Facebook. The competition is hosted online with a submission window opened from August 1st to September 1st in 2021.

With the sample videos provided by us, contestants are required to perform multi-class multi-object tracking on a series of videos captured by drones (Figure 1.1). The solution should decide when balls change hands by indicating the frames when the ID of the person holding the ball. The hardware for the 2021 LPCVC contest is the Raspberry Pi 3B +, and

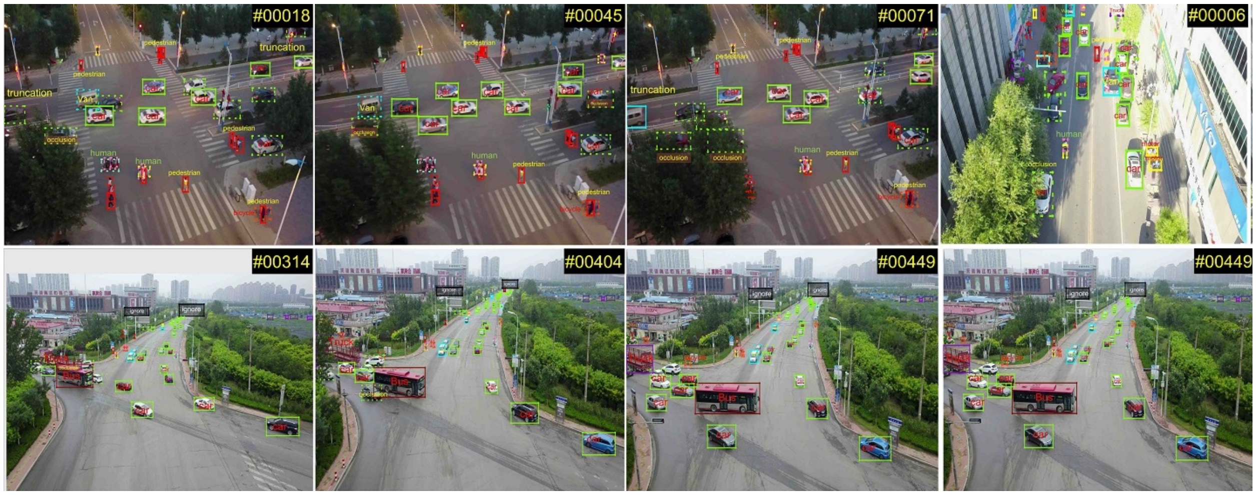


Figure 1.2. Sample annotated frames in the VisDrone-MOT2021 dataset.

the software framework is PyTorch, with technical support provided by the track sponsor Facebook PyTorch. During the month-long challenge, 53 teams submit 366 solutions with growing scores. By the end of the 2021 submission date, we receive 152 qualified solutions from 16 different teams around the world.

This thesis provides an analysis of all qualified submissions from 16 teams in the aspects of 1) compression of pretrained object detection models, 2) approaches to improve efficiency during inference, and 3) balance the trade-off between energy and accuracy. During the preparation of the competition, an innovative automated referee system and evaluation metrics are designed and implemented to provide a trouble-free submission period.

1.2.2 Optical Flow Oriented Multi-object tracking Solution on UAV

Optical flow is a conventional computer vision technique that computes the apparent multi-object tracking by individual pixels on the image plane. It provides the multi-object tracking information of each pixel in the image. This information can be combined with object detection to estimate the movement of objects in the video. Although the information provided by optical flow can be helpful for multi-object tracking, the traditional approach of calculating optical flow in video is inefficient because it requires finding correspondences between two input images at every piexl [10].

Recently, optical flow has returned to state-of-the-art object detection algorithms [11], [12] thanks to studies devoted to generating optical flow through deep learning. The deep learning approach of creating optical flow can generate a light-weighted multi-object tracking model [10], making optical flow cheaper than the traditional approach. Optical flow has also shown its advanced ability to perform object tracking proficiently [13].

To demonstrate that the inference latency can be reduced by augmenting multi-object tracking algorithms with optical flow, we propose an optical flow oriented multi-object tracking solution for drone-captured video, named OF-MOT. We create the object detector for a sequential video by aggregating the optical flow generated from a CNN-based flow generation network [10]) with a region-based detection network [14], and use a customized object tracker for instance association between keyframes to track the objects throughout the video. By analyzing the submissions of the 2021 LPCVC UAV Video track, we applied the inference acceleration techniques adopted by the participating teams to our solution and managed to reduce the model size and improve the efficiency of our model.

We present comparison experiments on the multi-class drone-captured video dataset - VisDrone2021-MOT dataset (Figure 1.2). OF-multi-object tracking runs in real-time at 34 FPS with tracing accuracy at 17% compare with the state-of-the-art multi-object tracking method [13], [15] running 19 FPS with accuracy at 22%. Our results suggest that there is instance-level multi-object tracking on modeling is of great importance for efficient online multi-object tracking solutions. Furthermore, we show that the optical flow can reduce the execution time while maintaining a reasonable accuracy, applicable in the condition where the payload for the battery is limited to UAVs.

1.3 Thesis Outline

The rest of the thesis is structured as follows: Chapter 2 presents the advantages and disadvantages of current multi-object tracking solutions and related research studies of tracking on UAV; Chapter 3 discusses the process of hosting the competition and the analysis of the submitted solutions; Chapter 4 demonstrates the proposed optical flow oriented multi-object tracking on UAV; Chapter 5 provides a summary of the findings of this thesis as well as the potential future development directions.

2. RELATED WORK

This chapter provides a background on multi-object tracking and discusses the existing solutions. This chapter is divided into three parts. Section 2.1 discusses the existing methods of multi-object tracking in different categories. Section 2.2 presents both conventional and deep learning approaches to compute optical flow, and how existing object detection solutions adopt optical flow. Section 2.3 presents the existing multiobject tracking in UAV studies, as well as competitions with similar subjects.

2.1 Multi-object Tracking

State-of-the-art multi-object tracking methods typically follow the "tracking-by-detection" paradigm: a detector first locates all objects of interest in each frame with bounding boxes. Tracking is then performed by the object association between frames through Re-ID. We classify existing works into two categories based on whether they use a single model or separate models to detect objects and extract association features.

2.1.1 Separate Detection and Tracking

Object Detection

For detection, conventional solutions operate the task of finding an arbitrary number of objects and the task of classifying every single object and estimating the bounding box into 2 stages. This is the reason why that this category is called two-stage detectors.

In two-stage detectors, the approximate object regions are proposed using deep features before these features are used for classification as well as bounding box regression for the object candidate. Girshick et al. [16] proposed Region-Based Convolutional Neural Networks (RCNN) that first select several proposed regions from an image and then label their categories and bounding boxes. The Fast-RCNN [17], Faster-RCNN [18], and Mask-RCNN [19] are extended variations of RCNN with modifications for faster processing and scene segmentation. Most benchmark datasets such as MOT17 [20] provide detection results obtained by popular methods such as Faster R-CNN and Scalable Real-time Dynamic Graph Partitioner

(SDP) [21]. This way, the work that focuses on the tracking part can be fairly compared with the same object detection for its superior performance on accuracy. However, higher accuracy comes with high computational resource requirements, making these methods difficult to deploy on edge devices, such as UAVs.

With the introduction of You Only Look Once (YOLO) [22], one-stage detectors start to emerge in the field of object detection. One-stage detectors predict bounding boxes over the images without the region proposal step. This process consumes less time and can therefore be used in real-time applications. The improved versions of YOLO, YOLOv3 [23], YOLOv4 [15], YOLOv5 [24], all prioritize the inference speed and are super fast but not as accurate at recognizing irregularly shaped objects or a group of small objects. SSD [25] detects objects in images using a single deep neural network by discretizing the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location.

Object Tracking

Simple Online Real-time Tracking (SORT) [26] first uses Kalman Filter [27] to predict future locations of the tracklets, computes their overlap with the detections, and uses Hungarian algorithm to assign detections to tracklets. DeepSORT [13] augments the IoU-based association in SORT with deep appearance (Re-ID) features. IoU tracker [28] directly computes the overlap between the tracklets (of the previous frames) and the detection without using Kalman Filter to predict future locations. Both DeepSORT and IoU Tracker are widely used in practice due to their simplicity.

2.1.2 Single Model for Both Detection and Tracking

With the rapid maturity of multitask learning in deep learning [29], joint detection and tracking using a single network has begun to attract more research attention. The first class of solutions performs object detection and re-identification feature extraction in a single network to reduce the inference time. For example, premature ovarian insufficiency (POI) [30] explores both high performance detection and Re-ID features. Track-RCNN [31] adds a re-identification head on top of Mask-RCNN[19] and regresses a bounding box and a re-identification feature for each proposal. Despite competitive accuracy, these trackers are difficult to run efficiently due to the separation of detection and Re-ID modules. To achieve high accuracy and higher running speed simultaneously, one-shot multi-object tracking methods are presented. FairMOT [9] incorporate the Re-ID module into a single-shot detector, so that the whole network can output detection and Re-ID features simultaneously.

2.2 Optical Flow in Object Detection

Optical flow is the motion of objects between consecutive frames of a sequence, caused by the relative movement between the object and the camera. There are two types of optical flow: sparse and dense. Sparse optical flow gives the flow vectors of some interesting features, such as a few pixels that depict the edges or corners of an object within the frame. The dense optical flow attempts to compute the optical flow vector for every pixel of each frame. The traditional way of estimating optical flow involves pixel-wise image recognition and calculation. For sparse optical flow, Lucas and Kanade [32] proposed a method that, under the assumption that the flow is essentially constant in the local neighborhood of the pixel under consideration, the basic optical flow equations for all pixels in that neighborhood can be solved with the least squares criterion. Farnback [33] proposed a dense optical flow estimation algorithm that requires processing all pixels in a given image and approximates the windows of image frames by quadratic polynomials through a polynomial expansion transform.

Recently, the usage of convolutional neural networks (CNNs) has become the method of choice in computer vision, including optical flow estimation. Fischer et al. [10] proposed an end-to-end CNN architecture called FlowNet, which estimates optical flow by producing the representations of the two images separately, and then combining them together in the "correlation layer", and learning the higher representation together. Ranjan et al. [34] proposed a pyramid architecture to train one deep network per layer to compute the flow update, where each level is defined based on the number of pixels related to the movement. Ilg et al. [35] extended the work of FlowNet and proposed a stacked architecture named

FlowNet2 that includes warping of the second image with intermediate optical flow and elaborates on small displacements by introducing a subnetwork specializing in small motions. FlowNet2 shows higher efficiency and precision than the original FlowNet.

With the rise of an efficient flow computation network such as FlowNet [10], optical flow and motion models have been utilized in object tracking and multi-object tracking in video. The temporal information in video provides the feasibility for improving video object detection [11], [36]. To utilize the temporal information, feature-level networks propagate the information in nearby frames. SiamMOT [12] proposes an online multi-object tracking method that combines a region-based detection network with two motion models.

OF-MOT is evaluated on a drone-captured video dataset and achieves 24 FPS with 17% accuracy on a modern GPU Titan X, showing that the optical flow can effectively improve the multi-object tracking.

2.3 MOT on UAV & Workshops

With the rapid growth in UAV technology, computer vision for UAV becomes increasingly important. The viewing angle and movement of the UAV make the multiobject tracking problem different from multi-object tracking problems captured by surveillance cameras. TNT [8] creates tracklets of each detected object using IoU as nodes throughout the entire video, and then combines the tracklets based on Re-ID.

Since 2017, the VisDrone [3]–[6] challenge incorporates multi-object tracking along with other computer vision tasks (e.g., object detection, crowd counting) into drone-captured video sequences. The 2020 LPCVC [7] UAV track is a competition in which participants used the given videos captured indoors to recognize the characters on the wall.

3. 2021 LPCVC UAV VIDEO TRACK

3.1 Introduction

Since 2015, the IEEE Annual International Low-Power Computer Vision Challenge (LPCVC) has been held to identify energy-efficient AI and computer vision solutions. These solutions have a wide range of applications in mobile phones, drones, autonomous robots, or any intelligent systems equipped with digital cameras carrying limited energy [37]–[39]. The 2021 LPCVC UAV Video track sponsored by Facebook is hosted online with a submission window opened from August 1st to September 1st in 2021. With the sample videos provided by us as organizers, the contestants are required to perform Multi-Class Multi-Object Tracking on a series of videos captured by drones (Figure 1.1). The solution should decide when balls change hands by indicating the frames when the ID of the person holding the ball. The hardware for the UAV2021 LPCVC contest is Raspberry Pi 3B+, and the software framework is PyTorch, with technical support provided by the track sponsor Facebook PyTorch. During the month-long challenge, 53 teams submitted 366 solutions with growing scores (Figure 3.1).

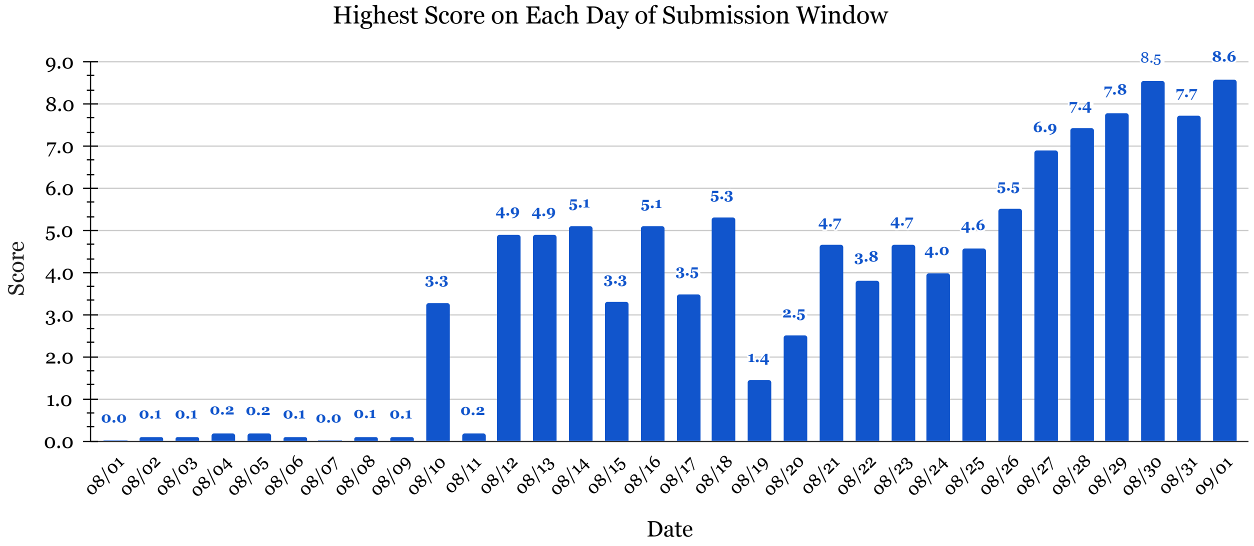
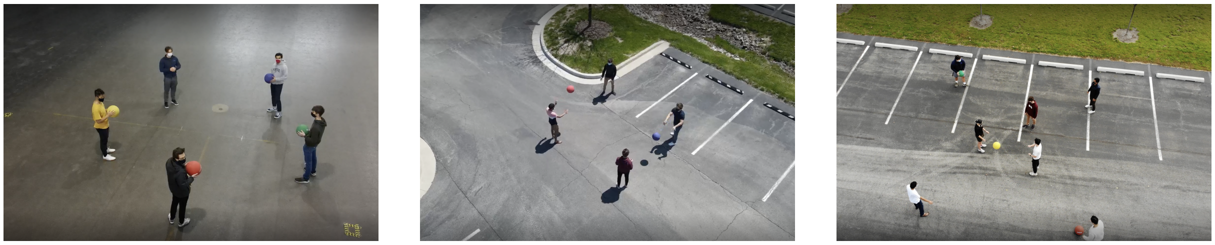


Figure 3.1. The highest score of the submissions on each day during the submission window. The score caluclation is explained in Section 3.2.3.



(a) 5p4b (b) 4p2b (c) 7p3b

Figure 3.2. Example testing video clips with different difficulty level. “p” stands for person and “b” stands for ball. “5p4b” means there are 5 different persons and 4 different balls in this video.

3.2 Challenge Setup

In this section, we discuss the preparation process for this track, including: creating testing videos, providing a sample solution, designing the evaluation metric, and setting up the automated referee system.

3.2.1 Testing Videos

Since there is no drone-captured video dataset fulfills the task we design for this track, we create the sample and tested the videos ourselves. To increase the diversity of the video content, the testing videos are divided into different difficulty levels based on the number of persons and balls in the video (Figure 3.2). The drone keeps circulating around the object to create a dynamic motion in the frames and occlusion of the object, which are the two key features of drone-captured video. All testing videos are released in the original format with a resolution at 3840 ×2160 at 30 frames per second. The drones: DJI Mavic Mini, DJI Mavic Air, and DJI Mavic Air 2.

3.2.2 Sample Solution

The purpose of a sample solution is to help participants achieve a better understanding of the submission format while not limiting the creativity of their solutions. A sample solution dose not need to be perfect, but should be thought-provoking with a clear explanation of the submission format.

With this in mind, our sample solution adopts the conventional multi-class multi-object tracking paradigm - "tracking-by-detection". Object detection is performed with YOLOv5 [24]. Then, we implement DeepSORT [13] for tracking because it contains multiple dimensions of features to track the instance across frames and has been widely used in many MOT projects. The sample solution is open-source at https://github.com/lpcvai/21LPCVC-UAV_VIdeo_Track-Sample-Solution.

3.2.3 Evaluation Metric

The LPCVC must be able to create meaningful metrics to score the submitted solutions. Therefore, we create a set of scoring equations that do not require per-frame annotation like the conventional multi-object tracking testing dataset.

The accuracy of each solution is determined by detecting the catches (Equation 3.1). Catch detection is performed to measure the two major components of an MOT solution: object detection and re-identification. A catch is defined within this competition as the instant a thrown ball touches a person's hand. The output of the solution is compared to the manually labeled ground-truth. The referee system categorizes each catch in a solution's output as True Positive (TP), False Positive (FP), or False Negative (FN). True positives demonstrate accurate object tracking and re-identification. For example, if a solution detects a catch within ±5 frames of the ground-truth, with the correct person ID ($correct_{\mathrm{i}}$), it is marked as a true positive. False positives assess re-identification performance due to object tracking correctly calculating a catch with incorrect ID matching. A false positive is given for a detected catch within the frame threshold but with incorrect person IDs. F alse negatives demonstrate poor object tracking due to the ball not seen entering a person's hands. A false negative occurs when the MOT solution does not output a catch within ±5 frames of the groundtruth.

$$Accuracy = \frac{\sum_{\mathrm{i}=0}^{n} \frac{\mathrm{correct_i}}{\mathrm{total_i}}}{TP + \frac{1}{2}(FP + FN)} \tag{3.1}$$

After calculating the accuracy, we use the Yokogawa WT310 power meter to measure the power consumption $Power$. The power meter starts tracking the power consumption by

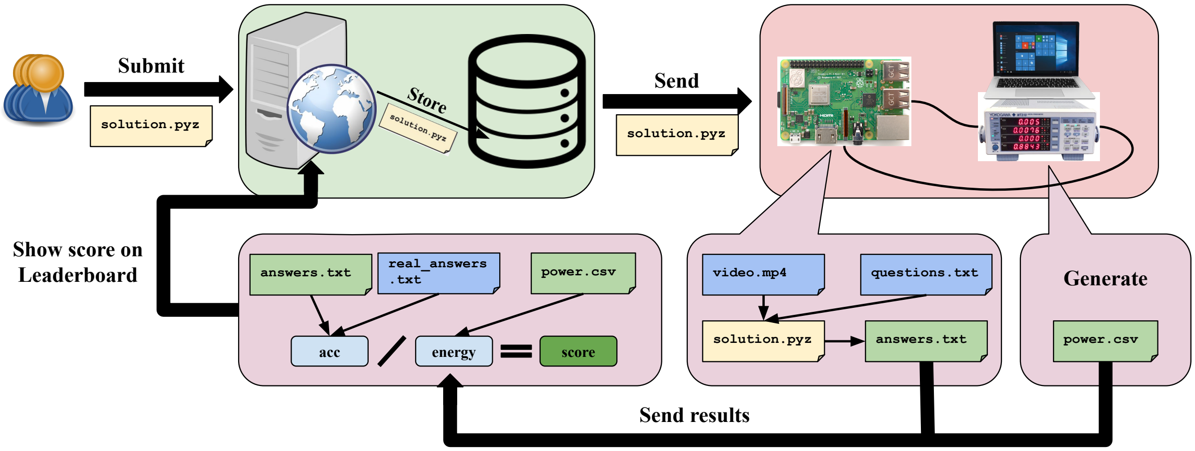


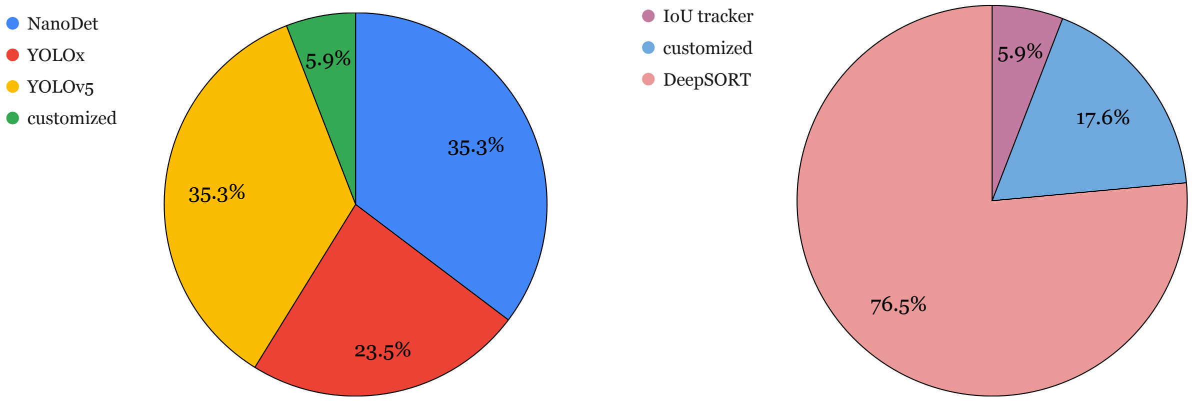
Figure 3.3. After a submission is received from the website, it is stored in a queue on the server where the website is hosted. When the execution of the previous submission is accomplished, the next queued submission is sent to the Raspberry Pi 3B+ to be executed. The power meter is triggered at the same time when the execution starts. Once finished, the power meter generates a CSV file contains power consumption information and sends it back to the server along with the accuracy score from the Raspberry Pi 3B+. Eventually, a script on the server calculates the final score and reflects to the website leaderboard.

executing the submission on the Raspberry Pi 3B+ and ends once the output of the last test video is received. Then, the accumulated power consumption during the execution period is extracted from a CSV file outputted by the power meter. Since we only have one power meter, all submissions are sent to a queue on the server to be ran by the grading system individually.

The speed of a solution is highly correlated with energy consumption: faster solutions require less runtime and less energy on the same hardware. Most high-scoring solutions focused on increasing processing throughput, leading to less energy consumption and thus much higher scores.

The final score of the tested solution is represented by the ratio of accuracy and energy for the scores, i.e.,

$$Score = \frac{Accuracy}{Energy} \tag{3.2}$$



(a) Object detection (b) Object tracking

Figure 3.4. The percentage portion of the methods used in the object detection task and object tracking task.

Notice that although *Accuracy* has the range from 0 to 1, there is no specific range for *Energy*, which makes *Score* ranges from 0 to ∞.

3.2.4 Automated Referee System

Large computer vision competitions such as ImageNet Large Scale Visual Recognition Challenge [40] usually provides an online leaderboard for participants to evaluate and improve their submissions. We design a referee system with an online leaderboard at https://lpcv.ai/scoreboard/Video21 to reflect the scores of the submitted solutions. The detailed workflow of the automated referee system can be found in Figure 3.3.

3.3 Submission Results

This section presents the best solutions for all qualified teams (Table 3.1). A team is considered qualified if their best solution exceeds the score of the sample solution. We also present the evaluation analysis focusing on the techniques that participants applied to detection, tracking, model compression, and inference.

Table 3.1. The table of all qualified submissions from 16 different teams and the sample solution. The rank represents the ranking of the team's best submitted solution amoung all submitted solutions.

Rank	Team	Energy	Accuracy	Score	Detection	Tracking
1	VITA	0.091	0.790	8.569	customized	SORT
2	baseSlim	0.097	0.830	8.556	NanoDet	DeepSORT
3	LPNet	0.078	0.683	8.551	YOLOv5	IoU tracker
9	spring	0.110	0.820	7.458	YOLOx	re-identification
19	yycnn	0.119	0.750	6.304	NanoDet	DeepSORT
20	mmtty	0.109	0.683	6.207	NanoDet	DeepSORT
23	asw	0.115	0.677	5.761	NanoDet	DeepSORT
31	xdmca	0.134	0.713	5.259	YOLOx	DeepSORT
42	baili	0.176	0.830	4.695	NanoDet	DeepSORT
38	sl	0.164	0.800	4.485	NanoDet	DeepSORT
83	BCJ	0.164	0.337	2.090	YOLOv5	DeepSORT
88	Shiyu	0.170	0.337	1.996	YOLOx	DeepSORT
92	zy42	0.167	0.297	1.833	YOLOv5	DeepSORT
93	SYSU	0.288	0.520	1.793	YOLOv5	customized
111	PotatoNet	0.393	0.313	0.800	YOLOx	DeepSORT
123	FoxNoPanda	1.108	0.367	0.337	YOLOv5	DeepSORT
139	Sample Solution	2.264	0.230	0.105	YOLOv5	DeepSORT

3.3.1 Detection & Tracking

As show in Table 3.1, most solutions apply the conventional "tracking-by-detection" paraidgm when designing their solutions. This structure provides flexibility in trying different combinations of detection and tracking strategies.

As shown in Figure 3.4(a), there are 35.5% solutions using NanoDet [41], and 35.5% using YOLOv5. Given the potential influence of using YOLOv5 in the sample solution, NanoDet shows its popularity in participants when selecting low-power object detector for edge devices. NanoDet is a Fully Convolutional One-Stage Object Detection (FCOS) style one-stage anchor-free object detection model that uses generalized focal loss as classification and regression loss [42]. According to its authors, Lyu et al., NanoDet can achieve up to 34.3 mean Average Precision (mAP) on COCO[43] validation dataset and still maintain real-time on CPU when running the object detection task. As a relatively new object detection solution, NanoDet has attracted attention in the computer vision field and hopes to replace the YOLO series in low-power object detection. Our evaluation results also demonstrate that a robust detector can significantly push forward the drone-based multi-object tracking. For example, all leading solutions adopt some advanced detectors that are designed to run in low-power environments or high efficiency, such as NanoDet and YOLOv5. Moreover, the association of cross-frame object instances is highly dependent on reliable and highly representative features. With the occlusion of the balls in the frame, the top-leading solutions all have not only color classification, but also motion features of the detected objects to improve the tracking. Similar strategies can be applied to other objects occluded in drone-captured videos.

For tracking, 76.5% of the teams selected DeepSORT to be implemented with their detectors (Figure 3.4(b)). DeepSORT has dominated the field of multi-object tracking as the ideal solution for different types of videos since being introduced to the research community in 2017. It solves the problem of ID switching in the traditional tracker (e.g., IoU tracker) using an AI model that compares the similarity between people, thus reducing the problem of switching people's identities. Unlike the VisDrone2021-MOT challenge [6] where most submissions used IoU tracker [28] or related techniques for tracking, only one team submitted their solution using the IoU tracker for our challenge. This is because there are many more objects that need to be detected in VisDrone videos, making assigning multi-dimensional features to each detection very inefficient. Since our videos have at most 5 persons and 4 balls, our participants have more freedom to assign more diverse features to the tracked objects. Also, unlike the VisDrone videos where the drone is flying at least 50 meters above the ground, our drone is flying at around 15 to 20 meters. Therefore, the objects in our video are much bigger as well. From the fact that most submissions adopt DeepSORT, we believe that similarity calculation-based re-identification is crucial for developing advanced multi-object tracking algorithms. This result also shows the demand for a more dynamic and content-based object tracker in the multi-object tracking field to resolve the increasingly complex and diverse video data.

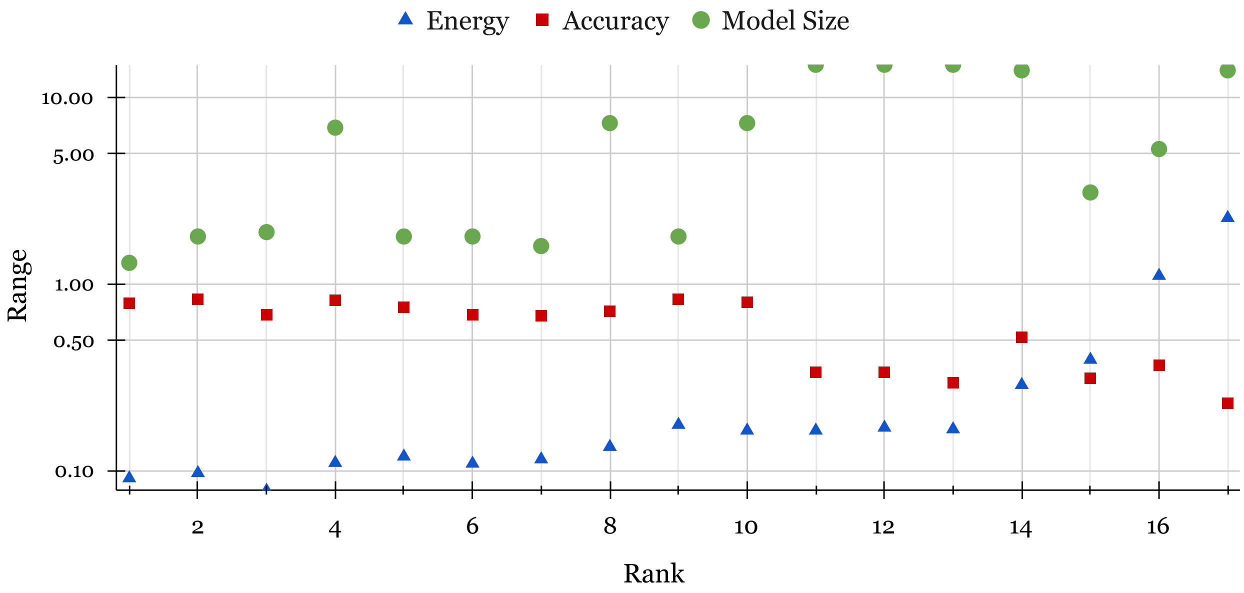


Figure 3.5. The correlation between the model size to energy and accuracy individually. The vertical axis "Range" represents the range of data for Energy and Accuracy individually. The unit of energy is kW (ranges from 0 to ∞); the unit of accuracy is percentage (ranges from 0 to 1); the unit of model size is MB. All values are in logarithm value for better visualization.

3.3.2 Model Compression

Although no compression technique is implemented in our sample solution, all qualified teams apply model compression to their trained models to increase the efficiency of their solutions. Quantization is the most utilized technique, as they are the easiest to implement without needing to modify the original neural network architecture. After anlayzing the submitted solutions, all qualified solutions quantize the data type from float32 to float16. This finding shows the importance of model compression for object detectors to run on edge devices.

The ability of shrinking your model into the As shown in Figure 3.5, all the top 10 solutions contain pre-trained weights with the total size less than 5 MB for object detection and tracking (only team LPNet has separate pre-trained weights from re-identification other than DeepSORT). For example, the winning team VITA provides a winning solution strikes a 0.790 accuracy, which is not as high as the accuracy score of the 2nd place team baseSlim at 0.830. However, VITA's solution only takes 0.091 kW to process, which is less than 0.097

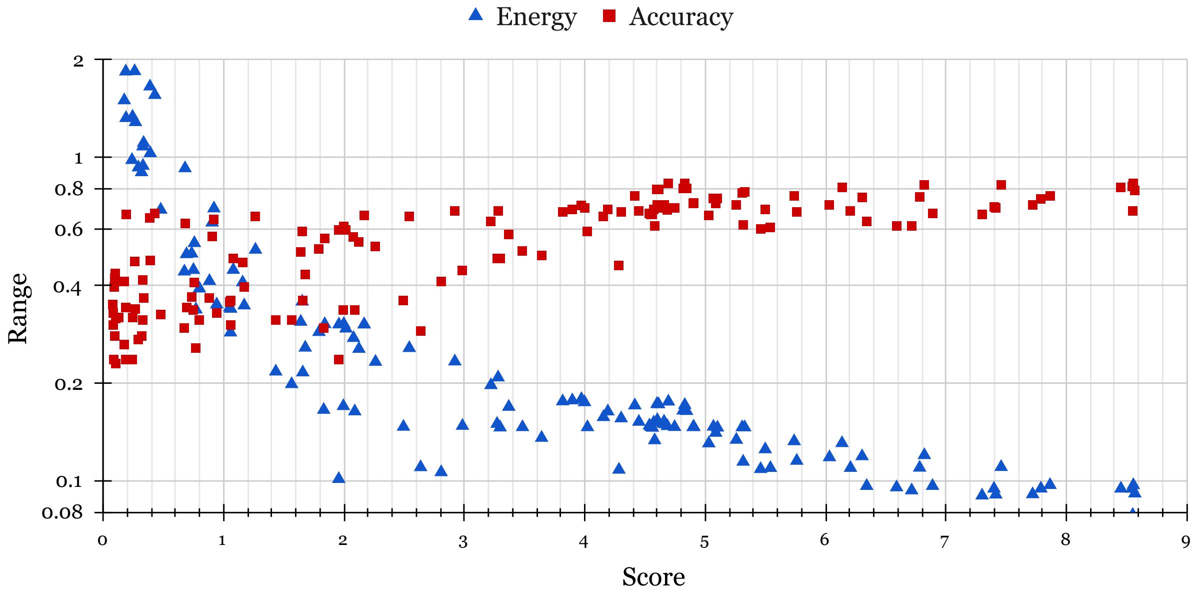


Figure 3.6. The correlation between the energy and accuracy with score of all qualified submissions. The vertical axis "Range" represents the range of data for Energy and Accuracy individually. The unit of energy is kW (ranges from 0 to ∞) and the unit of accuracy is percentage (ranges from 0 to 1). Both values are in logarithm value for better visualization.

from baseSlim. This leads to VITA's final score exceeding baseSlim's by 0.014 and eventually wins the competition. It is also worth noting that 35.3% solutions (80% top 10 solutions) have their weights stored in .jit or .torchsciprt instead of .pt or .pth to reduce the model size. This provides a guideline for compressing multi-object tracking models that will be deployed on edge devices with limited computational power (e.g., Rawsp berry Pi) should be aiming for a pre-trained weights with size less than 5 MB.

3.3.3 Inference

Apart from training, increasing efficiency during inference is also an important solution to reduce the energy of the tested solution. Most teams focus on reducing energy by speeding up the execution time of their solutions. As shown in Figure 3.6, the trend of energy is more steep than the tread of accuracy, showing that higher score solutions always consume less energy.

In general, qualified teams take the following approaches to increase inference runtime: batch processing, frame skipping, and downsampling. As introduced in Section 3.2.2, our sample solution is real-time processing, which processes the input video frame by frame and then stores and updates the processed information at each frame. However, there are 23.5% qualified teams (including the winning team VITA) that implement batch processing to achieve less memory cache during execution. Team FoxNoPanda only modifies the sample solution from real-time processing to batch processing and gained almost 200% times better score. For frame skipping, 82.4% qualified teams follow the sample solution, setting the frame skipping rate to be every 9 frames. For down-sampling, the sample solution changes the input size from 3840×2160 to 640×320. On the basis of the required shape of different object detectors, most teams have a similar or smaller input shape to achieve a balance between efficiency and accuracy.

3.3.4 Winning Solution Analysis

In this section, we analyze the winning solution of the VITA team to show the techniques applied in both training and inference.

System Design

The team's Energy-Efficient Tracking-based Action Detector (E^{2}TAD) has three core components: ball-person detection, deep association, and action detection.

Ball-Person Detection: Existing datasets are available to detect pedestrians. However, there is no ball-related dataset in the literature. The team adopts harmonization-aware image composition to generate synthetic but realistic balls on pedestrian datasets, *e.g.*, VisDrone, COCO ball-person subset PANDA (found by the baseSlim team). To address efficiency, the team designs a lightweight model based on YOLOv3 and YOLOv5. Among the three branches that detect objects in multi-scales (*e.g.*, small, medium, and large), the middle one is retained to detect medium-scale objects. The team's proposed two YOLO variants are dubbed YOLO-MobileV1 and YOLO-MobileV2. Initiated from YOLOv5, YOLO-MobileV1 modifies PANet in YOLOv5 and cuts off one upsample module to output the medium-size feature map directly. As a descendant of YOLOv3, YOLO-MobileV2 replaces the convolution with depth-wise convolution and removes upsampling modules.

The team VITA proposed models are inspired by the knowledge of [44], which validates that simple techniques can still achieve state-of-the-art performance. Following this point, the team continues to implement channel pruning on each convolution filter and decrease the number of residual modules to one in CSP.

Deep Association: This module tags the detected persons and balls with their corresponding identity labels. First, an identity-aware feature extractor (*e.g.*, ResNet-18 trained on the person re-identification task) is adopted to obtain the feature embeddings of the detected persons/balls, given the bounding boxes together with the raw image. Second, Deep Association maintains the person tracklets and ball tracklets in a gallery, where each tracklet is a queue that stores re-identification features of the latest K tracks for the corresponding person/ball. Last, using the re-identification feature of newly detected persons/balls as queries and defining the cost of matching a query with a specific tracklet in the gallery as the distance between their re-identification features, Deep Association formulates the identity tagging as a *minimum cost assignment problem* that can be solved in polynomial time by the Hungarian algorithm. To reduce association errors, any distance larger than a predefined threshold is marked as ∞ in the cost matrix to rule out the possibility of assignment.

Action Detection: Free of learnable parameters, this module only depends on bounding box trajectories to spatio-temporally localize key actions (*e.g.*, catch and throw), thus extremely energy-efficient. A ball has two states: collision or non-collision. Collision happens when the center of one ball's box falls into one person's box. For each ball, the method plots its collision history with other persons as a gated signal spanning from 0 to T, where T is the total number of frames. The start and the end of any connected part in the gated signal signify the catch action and throw action, respectively.

Efficiency

Here we discuss the efficiency strategies adopted by the team VITA in their submission.

Pruning: Pruning can be done in an unstructured or structured way. Unstructured pruning removes individual weights at the kernel level, while structured pruning removes groups of weight connections, such as channels or layers. Unstructured pruning leads to higher sparsity but requires specific hardware support for acceleration, thus not applicable to Raspberry Pi.

Quantization: There are two types of quantization: *post-training quantization* and *quantization-aware training* (QAT). Unlike *post-training quantization*, QAT supports concurrent training and quantization. Consequently, such a simulation of quantization errors during training gives minimal accuracy loss if the representation bitwidth is converted from FP32 to INT8.

Compressing re-identification Model as A Case Study: is Moreover, the input image resolution is reduced to 1/16 (from 512 × 256 to 128 × 64) and the re-identification model is finetuned on the resized images. Lastly, reducing the bitwidth from FP32 to INT8 by QAT gives faster computation and lower memory usage for re-identification: 3 times inference time speed-up and reduced 73% of the model size with minor accuracy loss.

Adaptive Inference: is Given the observation that persons and balls' locations are not changing abruptly since the videos are temporally coherent, ARC predicts the next frame's activity region ($\hat{I}_{t+1}$) based on the current frame's bounding boxes' coordinates (B_t) and crops out the non-activity region after adding some residual space (Δx and Δy), whose details are shown below:

$$\hat{I}_{t+1} = I_{t+1}[x_l - \Delta x : x_r + \Delta x, y_p - \Delta y : y_b + \Delta y], \tag{3.3}$$

where $x_l, x_r, y_p, y_b = \min B_t^x, \max B_t^x, \min B_t^y, \max B_t^y$. Since ball-person collision is a necessary condition for throwing or catching a ball, CI skips the succeeding Deep Association and Action Detection modules if no ball-person collision is inspected.

Cache-friendly Pipeline: There are three computation bottlenecks in the E^{2}TAD (ordered descendingly by profiling): video decoding, re-identification feature extraction, and

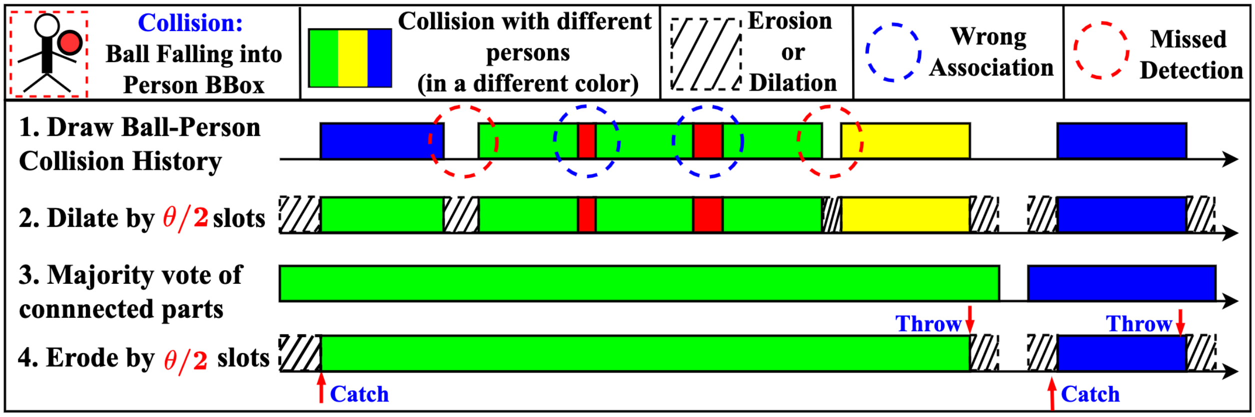

Collision:
Ball Falling into Person BBox
Collision with different persons (in a different color)
Erosion or Dilation
Wrong Association
Missed Detection
1. Draw Ball-Person Collision History
2. Dilate by $\theta/2$ slots
3. Majority vote of connnected parts
4. Erode by $\theta/2$ slots
Throw
Throw
Catch
Catch

Figure 3.7. Proposed heuristic approach for action detection. θ is the prior of minimum flying time for any ball. The solution first goes through the entire video and draw ball-person collusion history (1). Then, the parts where solutions considers no one has the ball are dilated based on θ (2). For the parts where there are collusion with different persons (connected parts), the solution will make up based on majority vote of the collision information before and after that part (3). Lastly, the beginning and ending of each period of the collision are eroded by $\theta/2$ to make the collusion more accurate (4). This figure is provided by the team VITA and improved by the author with consent.

ball-person detection. The computation pipeline of E^{2}TAD can be accelerated by addressing "temporal locality" for higher cache hit rate.

To improve the temporal locality of re-identification feature extraction, crop records and box records are used since the layer-wise computation can be fit in the L1 cache (32KB), and the weights after pruning and quantization can be fit in the L2 cache (512KB). To improve the spatial and temporal locality of video decoding, an image queue is used so that temporally consecutive frames can be decoded without interruption. (Video's temporal redundancy is utilized for high compression rates so that consecutive frames' encoded signals are closely packed together in memory.) Since the computation cost is fixed for a given video, the latency can be significantly reduced due to less memory access.

Action association: As the downstream task, action detection is vulnerable to errors propagated from the upstream tasks (*e.g.*, detection and association). In detection errors, false-negative cases may owe to occlusion, and false-positive cases can result from patches with homogeneous color. Association errors arise from non-discriminative re-identification features. Inspired by morphological operation-based denoising [45], VITA team proposed a heuristic approach in 3.7 to eliminate the preceding errors from the upstream tasks. As prior, θ is the minimum flying time for any ball. Given the collision history with different persons as a gated signal for each ball, the team first dilates it by $\theta/2$. Then, the team votes in the connected parts and selects the majority as the associated person holding the ball, so that the preceding detection and association errors are eliminated. Last, the dilated gated signal is eroded by $\theta/2$.

3.4 Conclusion

In conclusion, the 2021 LPCVC UAV Video track is a successful challenge that collected many design choices of low-power multi-object tracking solutions on drone-captured video. The analysis on the submitted solutions demonstrates that to deploy multi-object tracking on UAV: (1) a robust detector can significantly push forward the accuracy, and usually one-shot detectors (e.g., NanoDet, YOLO) are preferable for their high efficiency, (2) a good cross-frame object instance association heavily leans on reliable and highly representative features that involves both color and motion, (3) a similarity calculation-based re-identification is crucial for developing advanced multi-object tracking algorithms, (4) model compression by quantization from Float32 to Float16 can be used as the first step to reduce the model size with minor accuracy reduction, and (5) frame-skipping at around 10 and down-sampled the input frame to around 480 $\times$270 for 30 FPS input video do not lose too much information of the original video. These findings can help the relevant research community to develop multi-object tracking solutions for UAV or participate in a similar competition in the future.

4. OPTICAL FLOW ORIENTED MULTI-OBJECT TRACKING ON UNMANNED AERIAL VEHICLES

In this section, we present the details of the proposed optical flow oriented method named OF-MOT for multi-object tracking in drone-captured videos. OF-MOT adopts the conventional "tracking-by-detection" paradigm, but improves the object detection with optical flow guided feature aggregation and replaced the re-identification with motion estimation (Figure 4.1). We also apply the findings from the 2021 LPCVC UAV Video track submission analysis to training and inference, resulting in a more efficient model with smaller size. The proposed OF-MOT is trained and validated on the VisDrone-MOT2021 dataset and achieves an average accuracy of 17.7% with 24 FPS on a computer with modern GPU.

4.1 Video Processing

Multi-object tracking methods often do not require object detection for successive frames in a video, since the successive frames are very similar [11], [36]. Therefore, we treat each frame in a more efficient way during the video preprocessing. For each frame in a video, only the first frame (called keyframe) is stored, while the frames between this frame and its next keyframe (called non-keyframes) are not considered. This means that the keyframes are fixed and there is a keyframe in every k continuous frame. To clearly describe the proposed method, we define the keyframe as F_t. where $t \pm k$ represents the frame in the interval of k before or after the keyframe F_t. As summarized in 3.4, the frame-skipping number should be 10 for 30 FPS video, we set $k = 5$ since the input video has FPS at 15. The input frame is down-sized by half into 500×300 to reduce the number of pixels that need to be processed. The downsizing ratio is also set based on the analysis of 2021 LPCVC. Eventually, we learn from Section 3.4 that batch processing requires less computational power than real-time processing, so we adopt the batch processing approach to process the video.

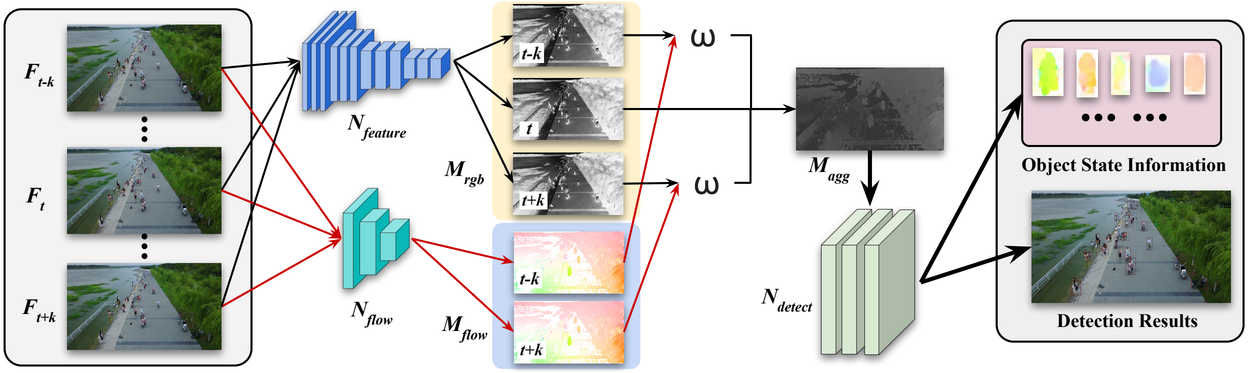


Figure 4.1. The overall network architecture of OF-MOT. For detection on frame F_t, the frame F_{t+k} and F_{t-k} are used. The RGB feature map extracted by $N_{feature}$ is stored as M_{rgb}, while the motion feature map extracted by N_{flow} is stored as M_{motion}. M_{rgb} of F_{t-k} and F_{t+k} are warped respectively through the feature warping function $\omega(\cdot)$ guided by each M_{flow}. Warped features and the M_{rgb} of F_t are aggregated into M_{agg}. Lastly, the head detector of R-FCN (N_{detect}) is applied to accomplish the video object detection task on the M_{agg}. The state information for each detected object will also be stored for instance association in the tracking.

4.2 Detection Network Architecture

Optical Flow Network: We adopt the CNN-based optical flow generation network FlowNet [10] as the backbone of optical flow feature extraction, denoted as N_{flow}. N_{motion} takes two consecutive keyframes F_t and F_{t-k} to compute the optical flow map denoted M_{motion}. To reduce the amount of computational stress during training, the optical flow output resolution is $\frac{1}{4}$ of the original image size with the original network output stride at 4. Since the feature map will be extracted by output stride at 16, to match the resolution of the feature map extracted later, we down-scale the flow field by half using bilinear interpolation.

Feature Network and Aggregation Inspired by the Flow-Guided Feature Aggregation (FGFA) [46], we adopt a similar architecture for object detection in drone-captured videos. The original FGFA uses ResNet-50 and ResNet-101 [47] as the backbone networks for their superior deep learning capabilities by utilizing the residual blocks in the neural network to achieve higher accuracy. However, what comes with the ResNet's superior deep learning ability is the enormous amount of parameters. For example, ResNet-50 outputs a model that has more than 23 million trainable parameters. To make the output model smaller, we adopt ResNet18 [48] as the backbone, since it is the smallest ResNet with 11 million trainable parameters. Since we apply the classification on the aggregated feature map later, the last 2 fully connected layers for object classification are discarded, and the first block of the conv5 layer is modified to have a stride of 1 instead of 2. This way, we are able to collect the features extracted from the previous layers of ResNet without conducting an actual object classification on the features. The last two layers of ResNet-18 are re-shaped into $256 \times 38 \times 63$ to make the output shape fit into the *R-FCN* head for object classification. The extracted RGB feature map is denoted as M_{rgb}.

After creating M_{rgb}, we aggregate the features of the current keyframe by warping the feature maps from the previous keyframe to the current keyframe guided by the optical flow. These feature maps provide more information of the objects in the current keyframe, such as varied illumination and nonrigid deformations, to make object detection more accurate. Then, we employ different weights at different spatial locations and let all feature channels share the same spatial weight to achieve aggregation. The warped feature map M_{agg} is defined in Equation 4.1 as follows.

$$M_{agg} = \sum_{\mathrm{i}=t-k}^{t+k} M_{rgb} \cdot \omega(M_{rgb}, M_{motion}) \tag{4.1}$$

where $\omega(\cdot)$ is the bilinear warping function applied on all locations for each channel in the feature map. Eventually, the current keyframe has multiple feature maps from nearby frames and its own feature map.

Detection Network: We implement our detection network following the same design of the FGFA detection network. In practice, the aggregated feature map M_{agg} is sent to the detection network to draw the bounding box around the area of the object and classify the object based on its features. Since the aggregated optical flow features augmented the region proposal of the two-shot object detectors, the state-of-the-art *R-FCN* [14] is implemented, denoted N_{detect}. On top of the 1024-dimension feature maps, the RPN sub-network and the R-FCN sub-network are applied, which connect to the first 512-dimension and the last 512-dimension features respectively. Nine anchors (3 scales and 3 aspect ratios) are utilized in RPN, and 300 proposals are produced on each image. In addition to object detection and classification, M_{flow} is also applied for extracting object state information. The bounding box of each object is applied onto M_{flow}, where the motion information for each detected object will be calculated and stored for instance association in the tracking.

4.3 Instance Association and Object Tracking

Instance association is the key component of multi-object tracking in video. To achieve good accuracy, state-of-the-art multi-object tracking methods usually store the features of the previously detected objects in the queue, then compare the newly detected objects with the queue to find the match (i.e., re-identification) [8], [9], [13]. However, it suffers from complexity issues when looking at drone-captured videos: 1) the objects are smaller, making the feature comparison hard, and 2) the crowded and the large number of objects in the frame increase the time complexity during comparison.

Since we can extract the object state information from the detector, the object association can be completed by estimating the similarity between objects from the previous frame to the current frame based on their state information. To achieve this, we propose a customized tracking algorithm with the technique of Hidden Markov Model (HMM) with particle filter [49] using the object detector and object state information from the object detector.

HMMs are one of the most popular graphical models in practice. They are characterized by variables $X_1, X_2, ..., X_n$ representing hidden states and variables $E_1, E_2, ..., E_n$ representing observations (i.e., evidence). The subscript i in X_{i} and E_{i} represents a discrete slice of time. There are three probability distributions in HMMs: a prior probability $P(X_0)$, a transition probability $P(X_k \mid X_{k-1})$ and an emission (sometimes called observation) probability $P(E_k \mid X_k)$.

Given a series of observations, we want to determine the distribution over states at some time stamp. Concretely, we want to determine $P(X_k \mid E_1, E_2, ..., E_n)$. This task is called filtering. Particle filter is intended to estimate the state of the system at time k using the linear stochastic difference equation assuming that the state of a system at time k evolved from the prior state at time $k-1$. Such a technique has been commonly used in localization scenarios and is well known for handling arbitrary distribution situations, unlike Gaussian filters or Kalman filter. Therefore, we apply the particle filter to our tracking mechanism.

In practice, we use particle filters to estimate a posterior distribution for the location of each object in the frame with the updated object state information in each frame. For the transition probability we use a uniform distribution on each object's state information to find the match. For the emission probability, we use a normal distribution about the bounding box locations. Because the emission probability would have $O\left(\frac{N_k!}{N_{k-1}!}\right)$ parameters due to the number of possible permutations of the bounding boxes to match the detected objects, we split the emission probability into Bayesian inference of the location of its bounding box given the locations of the particles and a linear assignment problem. Lastly, each object will be matched using the state information with the objects detected from the previous frame to achieve an instance association.

4.4 Evaluation and Comparison Results

4.4.1 Training & Inference

Training and testing data comes from the VisDrone-MOT2021 dataset [6]. It consists of 79 video clips with 33,366 frames in total divided into three subsets: training set (56 video clips with 24,198 frames), validation set (7 video clips with 2,846 frames) and testing set (16 video clips with 6,322 frames). All videos are taken in crowded street scenes in urban areas of China with various weather and lightning conditions. Most videos are shot with the UAV moving forward above the streets where there are different means of transport (Figure 1.2).

All training sessions are conducted on the Titan Xp GPU running on Ubuntu 20.04. For OF-MOT, we train 40 epochs with a batch size at 40 to use as a starting point and continued pruning the model until the loss curve converges. We stop the training session when the prevision reached 0.832, where the batch size is at 368. SGD optimizer is used to train 120k iterations with a weight decay of $5e^{-4}$. The learning rates are 2.5×10^{-4} and 2.5×10^{-5} for the first 80K and the last 40K iterations, respectively. After training, we apply the 16-bit floating point quantization to compress the trained model. Eventually, the model size is reduced by 45.5% with only 0.5% of accuracy drop.

4.4.2 Evaluation Metrics

We provide the evaluation results of Average Precision of tracking (AP) based on the evaluation metrics provided by the VisDrone2021-MOT challenge. Specifically, the solution outputs a list of bounding boxes with confidence scores and the corresponding class ID. A tracklet (i.e., a list of detected bounding boxes that are considered as the same object) is considered correct if the Intersection over Union (IoU) overlap with the ground truth tracklet is larger than 25%. We sort the tracklets (formed by the bounding box detections with the same identity) according to the average confidence of their bounding box detection. The score of each category represents the mean score of all ten predefined classes in all video sequences tested, where each sequence is executed independently. To demonstrate the robustness and efficiency of our method, we also show multiple object tracking precision in Multi-object Tracking Precision (MOTP, defined in Equation 4.2) [20], frame per second (FPS) and model size in megabytes (MB).

$$MOTP = \frac{\sum_{\mathrm{i},t} d_t^{\mathrm{i}}}{\sum_t c_t} \tag{4.2}$$

where d_t is the distance between the localization of objects in the ground truth and the detection output and c_t is the total matches made between ground truth and the detection output. i represents the objects on frame t.

4.4.3 Comparison

To conduct a more comprehensive evaluation, we followed the implementation details of one of the the winning solutions in the VisDrone2021-MOT challenge. This solution utilizes the conventional "tracking-by-detection" paradigm combining the state-of-the-art efficient object detection method YOLOv4 [15], OSNet [50] for re-identification, and Deep SORT [13] for instance association. GPU acceleration and dependency adjustment are applied accordingly.

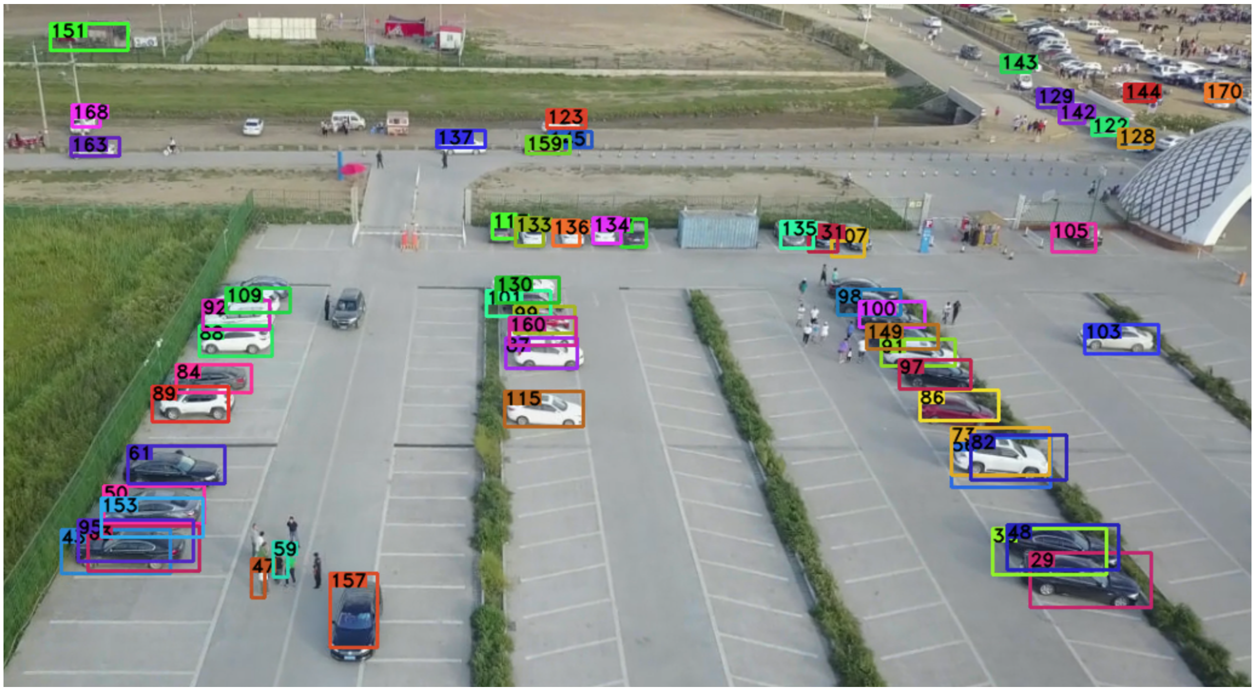


Figure 4.2. Compare to the pedestrians on the bottom left of this example frame, the ones on the top right are not detected for they are far away from the camera.

4.4.4 Results

In table 4.1, we present the *AP* score for the object class cars, buses, trucks, pedestrians and vans for a more comprehensive evaluation. The distribution of each object class in the training dataset influences the final detection and tracking results. According to the creator, there are 45.7% cars, 21.2% pedestrians, and 4.3% vans, 2.8% trucks, and 0.9% buses annotated in the training data. As shown in the table, both methods strike good accuracy on tracking the cars and pedestrians, given these are the majority of the training data. OF-MOT shows close-to-the-best score on the precision of tracking cars because the motion of cars is much more consistent in the videos captured by the drone. Due to the long distance from the drone to the objects, some objects are particularly smaller in the frame (Figure 4.2), making drone detection a more challenging task. Since we do not have re-identification features in the object tracking, the accuracy is heavily influenced by the motion of the objects in the video. As shown in table 4.1, AP_{car} is higher in OF-MOT while AP_{bus} is lower. This is because the cars in the video usually have more diverse motion information than the buses, leading to a worse association and worse tracking accuracy. In the 2019 VisDrone MOT challenge, one of the winning solutions proposed by Li et al. [51], Flow-Tracker, showed a 26.74% difference between AP_{car} and AP_{bus}, which is similar to the 31.6% difference from our solution. This further demonstrated that the objects speed in the frames can heavily impact the performance of motion-based MOT methods.

Table 4.1. The mean AP score (mAP) and individual AP score of each 5 reported object classes in the VisDrone2021-MOT testing set.

Method	mAP	AP_{car}	AP_{bus}	AP_{trk}	AP_{ped}	AP_{van}
YOLOv4 + DeepSORT	22.3	36.2	22.4	7.9	35.9	8.2
OF-MOT	17.7	42.5	10.9	7.3	20.7	15.4

In Table 4.2, OF-MOT shows a better MOTP result with the aid of optical flow. Optical flow captures pixels related to the moving object and aggregates them into the feature map before object detection. In this way, object localization is more precise compared to the other method which uses re-identification. The FPS is also significantly higher than that of the compared solution due to quantization and object association with optical flow. The results demonstrate the ability of instance association with optical flow instead of re-identification to improve efficiency of the overall multi-object tracking solution.

Table 4.2. Other evaluation scores of the tested methods.

Method	**MOTP**	**FPS**	**Size (MB)**
YOLOv4 + DeepSORT	57.6	19.7	245
OF-MOT	64.2	24	112

4.4.5 Discussion

In this thesis, we propose a novel multi-object tracking method OF-MOT, using optical flow information to assist both object detection and tracking tasks. Our OF-MOT achieves better FPS on drone-captured datasets compared to the other state-of-the-art multi-object tracking methods, while maintaining a reasonable accuracy. We also show some qualitative results on different scenarios and applications using OF-MOT, such as multi-object tracking on UAV. For future development, optical flow should be intergated not only with object detection and tracking, but also re-identification, to make OF-MOT more adaptable in more complex situations. Moreover, experimental studies should be conducted on different hardware platforms (e.g., Jetson Nano and Raspberry Pi) to evaluate the proposed MOT solution on edge devices.

5. SUMMARY & CONCLUSIONS

This chapter summarizes and concludes the findings presented in this thesis. Section 5.2 summarizes the innovation of using optical flow in both object detection and tracking to reduce the computational requirement compared to conventional multi-object tracking. Section 5.1 summarizes the findings by analyzing the 2021 Low-Power Computer Vision Challenge UAV Video track submissions. Section 5.3 discusses the potential future development regarding low-power multi-object tracking solutions for UAV video based on the findings listed in this thesis.

5.1 2021 LPCVC UAV Video Track

In this thesis, we present the process of hosting the 2021 LPCVC UAV Video track and the evaluation of the approaches adopted by qualified teams to make their submission low-power and accurate. We proposed a multi-object tracking task that does not require annotation on all frames by defining tracking as “which person is holding which ball”. In this way, the annotation only needs to document the moment when the ball changes hands. We captured videos with the drone circling around several people passing multiple balls to each other, with different difficulty levels based on the number of persons and balls in the video. A referee system involving Raspberry Pi 3B + testing hardware and a power meter is designed to automatically retrieve the submission and test the accuracy and power consumption to compute the final score. We also provide a sample solution using YOLOv5 for object detection and DeepSORT for object tracking. The sample solution also served as a baseline, where all submissions have a lower score than the sample solutions, which is considered disqualified.

Through the analysis of the submitted solutions, we found that 35.3% of the teams used NanoDet for detection and 35.3% of the teams used YOLOv5. Despite the influence of the sample solution, this result shows the advancement of NanoDet in low-power object detection on edge devices. For object tracking, 76.5% of the teams used DeepSORT, showing its superior ability to track objects when the video has dynamic motion and occluded objects. We also analyze the winning solution from the VITA team by presenting the design of an efficient object detector and the choices they made during inference to improve efficiency. The VITA team improved robustness by adversarial feature learning and morphological operation-based denoising, and improved efficiency by two adaptive inference strategies and a cache-friendly pipeline.

5.2 Optical Flow Oriented Multi-object Tracking

In this thesis, we propose a novel multi-object tracking method, OF-MOT, using optical flow information to aid both object detection and tracking tasks. The optical flow is used to guide the feature between the keyframes for region proposal to localize the object in the frame, as well as to add the motion feature to help object classification. Then the instance association is achieved through extracting the optical flow output of the detector model and applying the motion information of the detected object to the particle filter to associate it with itself in the previous frame. During implementation, the knowledge of low-power multi-object tracking on UAV that we summerized from the analysis 2021 LPCVC submissions was applied. To be more specific, we applied frame-skipping, model compression with quantization, and input frame downsampling to reduce the model size and, therefore, improve the inference efficiency. The proposed OF-MOT achieves better FPS on drone-captured datasets compared to the other state-of-the-art multi-object tracking methods, while maintaining reasonable accuracy.

5.3 Future Development

This thesis highlights the multi-object tracking solutions deployed on drone-captured videos. The practices presented in the thesis give a good direction for the future development of similar projects. The following work should improve the accuracy of tracking using the motion information from the object tracker or other sources combining with re-identification.

REFERENCES

[1] D. Du, Y. Qi, H. Yu, *et al.*, "The unmanned aerial vehicle benchmark: Object detection and tracking," *CoRR*, vol. abs/1804.00518, 2018. arXiv: 1804.00518. [Online]. Available: http://arxiv.org/abs/1804.00518.

[2] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for uav-based object detection and tracking: A survey," *CoRR*, vol. abs/2110.12638, 2021. arXiv: 2110.12638. [Online]. Available: https://arxiv.org/abs/2110.12638.

[3] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," *CoRR*, vol. abs/1804.07437, 2018. arXiv: 1804.07437. [Online]. Available: http://arxiv.org/abs/1804.07437.

[4] L. Wen, P. Zhu, D. Du, *et al.*, "Visdrone-mot2019: The vision meets drone multiple object tracking challenge results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 189–198. DOI: 10.1109/ICCVW.2019.00028.

[5] H. Fan, D. Du, L. Wen, *et al.*, "Visdrone-mot2020: The vision meets drone multiple object tracking challenge results," in Jan. 2020, pp. 713–727, ISBN: 978-3-030-66822-8. DOI: 10.1007/978-3-030-66823-5_43.

[6] G. Chen, W. Wang, Z. He, *et al.*, "Visdrone-mot2021: The vision meets drone multiple object tracking challenge results," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 2839–2846. DOI: 10.1109/ICCVW54120.2021.00318.

[7] X. Hu, M.-C. Chang, Y. Chen, *et al.*, "The 2020 low-power computer vision challenge," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4. DOI: 10.1109/AICAS51828.2021.9458522.

[8] G. Wang, Y. Wang, H. Zhang, R. Gu, and J. Hwang, "Exploit the connectivity: Multi-object tracking with trackletnet," *CoRR*, vol. abs/1811.07258, 2018. arXiv: 1811.07258. [Online]. Available: http://arxiv.org/abs/1811.07258.

[9] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "A simple baseline for multi-object tracking," *CoRR*, vol. abs/2004.01888, 2020. arXiv: 2004.01888. [Online]. Available: https://arxiv.org/abs/2004.01888.

[10] P. Fischer, A. Dosovitskiy, E. Ilg, *et al.*, "Flownet: Learning optical flow with convolutional networks," *CoRR*, vol. abs/1504.06852, 2015. arXiv: 1504.06852. [Online]. Available: http://arxiv.org/abs/1504.06852.

[11] X. Wang, Z. Huang, B. Liao, L. Huang, Y. Gong, and C. Huang, "Real-time and accurate object detection in compressed video by long short-term feature aggregation," *Computer Vision and Image Understanding*, vol. 206, p. 103 188, 2021, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2021.103188. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314221000321.

[12] B. Shuai, A. G. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," *CoRR*, vol. abs/2105.11595, 2021. arXiv: 2105.11595. [Online]. Available: https://arxiv.org/abs/2105.11595.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017. arXiv: 1703.07402. [Online]. Available: http://arxiv.org/abs/1703.07402.

[14] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," *CoRR*, vol. abs/1605.06409, 2016. arXiv: 1605.06409. [Online]. Available: http://arxiv.org/abs/1605.06409.

[15] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. arXiv: 2004.10934. [Online]. Available: https://arxiv.org/abs/2004.10934.

[16] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. arXiv: 1311.2524. [Online]. Available: http://arxiv.org/abs/1311.2524.

[17] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. arXiv: 1504.08083. [Online]. Available: http://arxiv.org/abs/1504.08083.

[18] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. arXiv: 1506.01497. [Online]. Available: http://arxiv.org/abs/1506.01497.

[19] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. arXiv: 1703.06870. [Online]. Available: http://arxiv.org/abs/1703.06870.

[20] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016. arXiv: 1603.00831. [Online]. Available: http://arxiv.org/abs/1603.00831.

[21] M. A. K. Patwary, S. Garg, S. K. Battula, and B. H. Kang, "SDP: scalable real-time dynamic graph partitioner," *CoRR*, vol. abs/2110.15669, 2021. arXiv: 2110.15669. [Online]. Available: https://arxiv.org/abs/2110.15669.

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

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. arXiv: 1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767.

[24] G. Jocher, A. Chaurasia, A. Stoken, *et al.*, *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*, version v6.1, Feb. 2022. DOI: 10.5281/zenodo.6222936. [Online]. Available: https://doi.org/10.5281/zenodo.6222936.

[25] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015. arXiv: 1512.02325. [Online]. Available: http://arxiv.org/abs/1512.02325.

[26] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *CoRR*, vol. abs/1602.00763, 2016. arXiv: 1602.00763. [Online]. Available: http://arxiv.org/abs/1602.00763.

[27] R. E. Kálmán, "A new approach to linear filtering and prediction problems" transaction of the asme journal of basic," 1960.

[28] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6. DOI: 10.1109/AVSS.2017.8078516.

[29] I. Kokkinos, "Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," *CoRR*, vol. abs/1609.02132, 2016. arXiv: 1609.02132. [Online]. Available: http://arxiv.org/abs/1609.02132.

[30] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: multiple object tracking with high performance detection and appearance feature," *CoRR*, vol. abs/1610.06136, 2016. arXiv: 1610.06136. [Online]. Available: http://arxiv.org/abs/1610.06136.

[31] B. Shuai, A. G. Berneshawi, D. Modolo, and J. Tighe, "Multi-object tracking with siamese track-rcnn," *CoRR*, vol. abs/2004.07786, 2020. arXiv: 2004.07786. [Online]. Available: https://arxiv.org/abs/2004.07786.

[32] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81, Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[33] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, ser. SCIA’03, Halmstad, Sweden: Springer-Verlag, 2003, pp. 363–370, ISBN: 3540406018.

[34] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” *CoRR*, vol. abs/1611.00850, 2016. arXiv: 1611.00850. [Online]. Available: http://arxiv.org/abs/1611.00850.

[35] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17.

[36] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, “Deep feature flow for video recognition,” *CoRR*, vol. abs/1611.07715, 2016. arXiv: 1611.07715. [Online]. Available: http://arxiv.org/abs/1611.07715.

[37] S. Alyamkin, M. Ardi, A. Brighton, *et al.*, “2018 low-power image recognition challenge,” *CoRR*, vol. abs/1810.01732, 2018. arXiv: 1810.01732. [Online]. Available: http://arxiv.org/abs/1810.01732.

[38] S. Alyamkin, M. Ardi, A. C. Berg, *et al.*, “Low-power computer vision: Status, challenges, and opportunities,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 411–421, 2019. DOI: 10.1109/JETCAS.2019.2911899.

[39] X. Hu, M.-C. Chang, Y. Chen, *et al.*, “The 2020 low-power computer vision challenge,” in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4. DOI: 10.1109/AICAS51828.2021.9458522.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[41] RangiLyu, *Nanodet-plus: Super fast and high accuracy lightweight anchor-free object detection model.* https://github.com/RangiLyu/nanodet, 2021.

[42] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: fully convolutional one-stage object detection,” *CoRR*, vol. abs/1904.01355, 2019. arXiv: 1904.01355. [Online]. Available: http://arxiv.org/abs/1904.01355.

[43] T. Lin, M. Maire, S. J. Belongie, *et al.*, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

[44] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” in *ECCV*, 2018, pp. 466–481.

[45] N. Jamil, T. M. T. Sembok, and Z. A. Bakar, “Noise removal and enhancement of binary images using morphological operations,” in *2008 International Symposium on Information Technology*, vol. 4, 2008, pp. 1–6. DOI: 10.1109/ITSIM.2008.4631954.

[46] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-guided feature aggregation for video object detection,” *CoRR*, vol. abs/1703.10025, 2017. arXiv: 1703.10025. [Online]. Available: http://arxiv.org/abs/1703.10025.

[47] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[48] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[49] M. Jaward, L. Mihaylova, N. Canagarajah, and D. Bull, “Multiple object tracking using particle filters,” in *2006 IEEE Aerospace Conference*, 2006, pp. 8–20. DOI: 10.1109/AERO.2006.1655926.

[50] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, “Omni-scale feature learning for person re-identification,” *CoRR*, vol. abs/1905.00953, 2019. arXiv: 1905.00953. [Online]. Available: http://arxiv.org/abs/1905.00953.

[51] W. Li, J. Mu, and G. Liu, “Multiple object tracking with motion and appearance cues,” *CoRR*, vol. abs/1909.00318, 2019. arXiv: 1909.00318. [Online]. Available: http://arxiv.org/abs/1909.00318.

VITA

Xiao Hu is an M.S. thesis student in Electrical and Computer Engineering at Purdue University. He received a B.S degree in Computer Science from Purdue University. He has served as a reviewer for the 2020 British Machine Vision Conference (BMVC) and International Conference on Computer Vision - Low-Power Computer Vision Workshop. He has received Purdue University ”Outstanding Undergraduate Researcher Award” and IEEE Rebooting Computing Certificate of Appreciation. Since 2019, Xiao has been acting as an organizer for the annual competition Low-Power Computer Vision Challenge (LPCVC). His current research focus is related to low-power computer vision for unmanned aerial vehicles (UAV). He also has experience in fairness in artificial intelligence and full-stack web development.

PUBLICATION(S)

A. Goel, C. Tung, **X. Hu**, H. Wang, J. C. Davis, G. K. Thiruvathukal, and Y.-H. Lu, “Low-power multi-camera object re-identification using hierarchical neural networks,” CoRR, vol. abs/2106.10588, 2021. arXiv: 2106.10588 . [Online].
Available: https://arxiv.org/abs/2106.10588.

A. Goel, C. Tung, **X. Hu**, G. K. Thiruvathukal, J. C. Davis, and Y.-H. Lu, “Efficient computer vision on edge devices with pipeline-parallel hierarchical neural networks,” CoRR, vol. abs/2109.13356, 2021. arXiv: 2109.13356 . [Online].
Available: https://arxiv.org/abs/2109.13356 .

X. Hu, M.-C. Chang, Y. Chen, R. Sridhar, Z. Hu, Y. Xue, Z. Wu, P. Pi, J. Shen, J. Tan,X. Lian, J. Liu, Z. Wang, C.-H. Liu, Y.-S. Han, Y.-Y. Sung, Y. Lee, K.-C. Wu, W.-X. Guo, R. Lee, S. Liang, Z. Wang, G. Ding, G. Zhang, T. Xi, Y. Chen, H. Cai, L. Zhu, Z. Zhang, S. Han, S. Jeong, Y. Kwon, T. Wang, and J. Pan, “The 2020 low-power computer vision challenge,” in 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2021, pp. 1–4. doi: 10.1109/AICAS51828.2021.9458522.

I. Ghodgaonkar, A. Goel, F. Bordwell, C. Tung, S. Aghajanzadeh, N. Curran, R. Chen, K. Yu, S. Mahapatra, V. Banna, G. Kao, K. Lee, **X. Hu**, N. Eliopolous, A. Chinnakotla, D. Rijhwani, A. Kim, A. Chakraborty, M. D. Ward, Y.-H. Lu, and G. K. Thiruvathukal, “Observing responses to the COVID-19 pandemic using worldwide network cameras,” CoRR, vol. abs/2005.09091, 2020. arXiv: 2005.09091 . [Online].
Available: https://arxiv.org/abs/2005.09091.

X. Hu, H. Wang, A. Vegesana, S. Dube, K. Yu, G. Kao, S.-H. Chen, Y.-H. Lu, G. K. Thiruvathukal, and M. Yin, “Crowdsourcing detection of sampling biases in image datasets,” in Proceedings of The Web Conference 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2955–2961, isbn: 9781450370233. [Online]. Available: https://-

doi.org/10.1145/3366423.3380063.

S. Kim, **X. Hu**, A. Vegesana, and K. Ramani, “First-person view hand segmentation of multi-modal hand activity video dataset,” in BMVC, 2020.

S. Kim, H.-g. Chi, **X. Hu**, Q.-X. Huang, Karthik, and Ramani, “A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks,” in ECCV, 2020.

S. Alyamkin, M. Ardi, A. C. Berg, A. Brighton, B. Chen, Y. Chen, H.-P. Cheng, Z. Fan, C. Feng, B. Fu, K. Gauen, A. Goel, A. Goncharenko, X. Guo, S. Ha, A. Howard, **X. Hu**, Y. Huang, D. Kang, J. Kim, J. G. Ko, A. Kondratyev, J. Lee, S. Lee, S. Lee, Z. Li, Z. Liang, J. Liu, X. Liu, Y. Lu, Y.-H. Lu, D. Malik, H. H. Nguyen, E. Park, D. Repin, L. Shen, T. Sheng, F. Sun, D. Svitov, G. K. Thiruvathukal, B. Zhang, J. Zhang, X. Zhang, and S. Zhuo, “Low-power computer vision: Status, challenges, and opportunities,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 2, pp. 411–421, 2019. doi: 10.1109/JETCAS.2019.2911899 .