End to End Camera only Drone Detection and Tracking Demo within a Multi-agent Framework with a CNN-LSTM Model for Range Estimation

Demonstration Track

Maxence de Rochechouart
Sorbonne University Abu Dhabi
Abu Dhabi, UAE
maxence.r@sorbonne.ae

Raed Abu Zitar
Sorbonne University Abu Dhabi
Abu Dhabi, UAE
raed.zitar@sorbonne.ae

Amal El Fallah Seghrouchni
Sorbonne University
Paris, France
Amal.Elfallah@lip6.fr

Frederic Barbaresco
Thales Group
Paris, France
Frederic.barbaresco@thalesgroup.com

ABSTRACT

We present an end-to-end camera-only drone tracking approach in a multi-agent framework. We show implementation and simulation of such a system and test the tracking components utilizing a CNN-LSTM model for range estimation tested on real data. A video of the demo is available at this link.

KEYWORDS

drone; camera tracking; MAS; CNN; LSTM

ACM Reference Format:
Maxence de Rochechouart, Raed Abu Zitar, Amal El Fallah Seghrouchni, and Frederic Barbaresco. 2024. End to End Camera only Drone Detection and Tracking Demo within a Multi-agent Framework with a CNN-LSTM Model for Range Estimation : Demonstration Track. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Drone tracking is a challenging task considering the complex motion and noise models involved [11] in simulating real-life environments. Radars that rely on Doppler signals are the most common sensors used for this purpose [12]. However, under some circumstances, high-resolution cameras with automatic tracking capabilities can also be used in drone tracking. Cameras are usually utilized in collaboration with a radar system [3] in generating high-quality tracks for the drone. Cameras, in their standard form, do not provide estimation for the drone range but can have more accurate estimates for the azimuth and elevation angles [8]. In this work, a Camera-only sensor will be used in drone tracking utilizing a multi-agent framework modeled with the Stone Soup simulator [1] and Mesa as the multi-agent platform [5]. The goal here is to present a demo work for a machine learning-based system that can estimate the range value of the drone based on training samples. The estimation is done using two methods; a (Convolutional Neural Network) CNN model is used to learn the estimation of the range in the first method and a CNN-LSTM (Long Term Short Term Memory) architecture is used for the estimation in the second method. On the other hand, a multi-agent model is used to schedule and represent the interaction between the different entities that are collaborating in implementing the different tasks of the tracker. The whole process can be summarized by; the drone scanning task, the detection task, the range estimation task, and finally the track generation task with the aid of the Kalman filter estimator. That would complete a cycle of the process and the tasks are repeated updating the track.

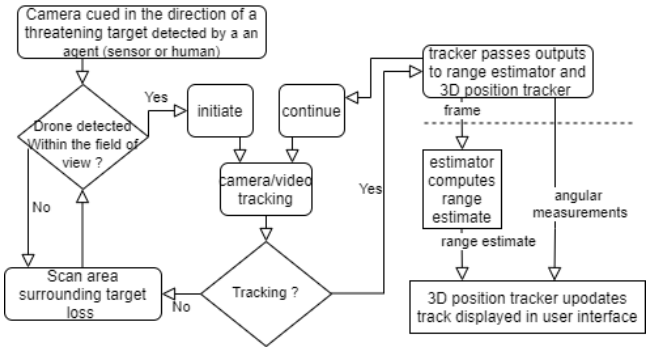

Figure 1: Interactions of system agents

2 PROBLEM STATEMENT AND PROPOSED MULTI-AGENT ARCHITECTURE

The most critical part of this system is the range estimation of the drone from a single camera output. It is important to note that if in the same context, two cameras were tracking the drone simultaneously, triangulating the drone position would be straightforward and greatly reduce the difficulty of tracking it [10]. A global view of the system is sketched in Figure 1, the individual agents are described in more details in Section 3. Although this demo only shows the use of the system in a single-target scenario, the flexibility of

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

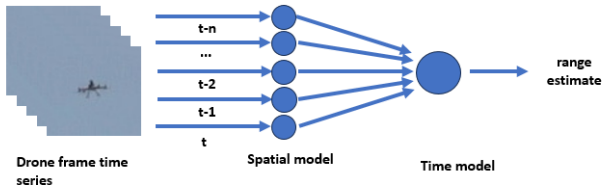

Figure 2: Range estimation from drone frames time series

the Multi-agent architecture allows to conveniently incorporate multiple targets and additional sensors.

3 THE DETECTION-TRACKING SYSTEM

The detector

The detector agent has to detect the presence of a drone within the field of view of the camera and locate it in the frame. It can be done automatically using real time object detection model such as YOLO [7] fine tuned [6] or adapted to the drone-detection problem. We did not address this issue here, and assume the role of detector to be played by a human agent.

The tracker

The tracker used here to track the drone in the video output by the camera is a Channel and Spatial Reliability Tracker (CSRT) implemented in OpenCv. At each time steps it outputs a bounding box for the tracked object.

The Range Estimator

The frame corresponding to the bounding box output by the tracker is fed in the pretrained CNN-LSTM model which outputs a rang estimate.

The 3D position tracker

The direction to which the camera is pointing along with the position of the bounding box output by the tracker are used to form accurate bearing and elevation measurements for the target position relative to the camera. The angular measurements are combined with the range estimate from the range estimator to form a complete 3D position in spherical coordinates. This position is then fed in the 3D position tracker to update the track displayed on the user interface. The tracker is made of an interacting multiple models (IMM) tracking with an extend Kalman filter (EKF) in Cartesian coordinates. Three kinematics model are used in the IMM to account for the agility and possibly heratic behavior of a drone target; nearly constant velocity (NCV), and left and right coordinated turn (CT).

4 THE RANGE ESTIMATOR

4.1 CNN Component

Initially, a deep learning model made of CNN layers followed by fully connected layers, is trained to estimate the range from single frames output by the video tracker of the camera, this is the spatial component of the final range estimator. We use a light state-of-the-art CNN model architecture (combined with batch normalization and squeeze and excite components [4]) called EfficentNetV2B0 [9] pre-trained on Imagenet dataset, with RGB input frames of size

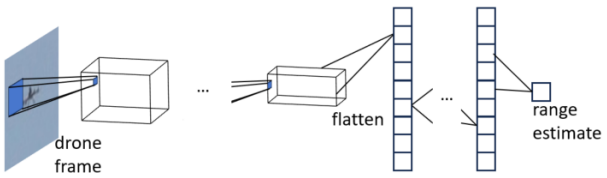

Figure 3: Sketch of CNN model

64×64. In the model with the TensorFlow library where the pre-trained EfficentNetV2B0 is available, initially the fully connected layers are trained, while the CNN pre-trained weights are frozen, and then the CNN layers are fine-tuned with a small learning rate while the fully connected layers are frozen in turn. This is a classical approach to transfer learning.

4.2 Long Term Short Term Memory (LSTM) Network

The LSTM Network is a Recurrent Neural Network (RNN) -trained model using Backpropagation through time to tackle the vanishing gradient problem [2]. It uses memory blocks connected through layers instead of neurons. A block contains gates that manage the state and output and operate on an input sequence. Each gate is triggered using the sigmoid activation units which make the flow of information conditional. The 3 types of gates within a unit are the Forget gate, which conditionally throws parts of the information away from the block, the Input gate which conditionally decides that the input value is updated, and the Output gate which decides the output based upon the input and memory of the block. The gates of the units have weights that are learned during training. For the input in our simple model, we will use one lookback input which mimics using the information from only one day before. LSTM is sensitive to scaling so before training the model, values are scaled and then unscaled after forecasting.

5 RESULTS

A simulation of the system was implemented with the MESA framework and is demonstrated in the video linked in the abstract. In the demo the system is tested with real data collected from a flight performed at 1.4km from the long range camera. The range estimation CNN-LSTM model was trained on the first half of the flight and the system was tested on the whole flight.

6 CONCLUSION

This demo paper presents simulations for a multiagent frame-based system that uses temporal deep learning in the detection and tracking of drones using only a high-resolution camera. The proposed machine learning model $CNN/LSTM$ learned to estimate the range that the camera can not typically provide alone. Several samples of drone video frames were used in the training. The track trajectory was generated with the aid of the Kalman filter using the information provided by the camera and the $CNN/LSTM$. The agents comprising the proposed system continuously and iteratively collaborated to fulfill this task. The generated tracks have comparative quality to the ground truths of the different samples.

REFERENCES

[1] Jordi Barr, Oliver Harrald, Steven Hiscocks, Nicola Perree, Henry Pritchett, Sebastien Vidal, James Wright, Peter Carniglia, Emily Hunter, David Kirkland, Divy Raval, Siyuan Zheng, Anne Young, Bhashyam Balaji, Simon Maskell, Marcel Hernandez, and Lyudmil Vladimirov. 2022. Stone Soup open source framework for tracking and state estimation: enhancements and applications. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXXI*, Ivan Kadar, Erik P. Blasch, and Lynne L. Grewe (Eds.), Vol. 12122. International Society for Optics and Photonics, SPIE, 1212205. https://doi.org/10.1117/12.2618495

[2] Jason Brownlee. 2022. Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. Available at https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/.

[3] Maxence de Rochechouart, Bashar I. Ahmad, Amal El Fallah Seghrouchni, Frederic Barbaresco, Stephen Harman, and Raed Abu Zitar. 2023. Drone Tracking Based on the Fusion of Staring Radar and Camera Data: An Experimental Study. In *2023 IEEE Radar Conference (RadarConf23)*. 01–06. https://doi.org/10.1109/RadarConf2351548.2023.10149552

[4] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. 2019. Squeeze-and-Excitation Networks. arXiv:1709.01507 [cs.CV]

[5] Jackie Kazil, David Masad, and Andrew Crooks. 2020. Utilizing python for agent-based modeling: The mesa framework. In *Social, Cultural, and Behavioral Modeling: 13th International Conference, SBP-BRiMS 2020, Washington, DC, USA, October 18–21, 2020, Proceedings 13*. Springer, 308–317.

[6] Francis Jesmar P Montalbo. 2020. A Computer-Aided Diagnosis of Brain Tumors Using a Fine-Tuned YOLO-based Model with Transfer Learning. *KSII Transactions on Internet & Information Systems* 14, 12 (2020).

[7] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. 2023. Real-Time Flying Object Detection with YOLOv8. arXiv:2305.09972 [cs.CV]

[8] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. 2015. Flying objects detection from a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4128–4136.

[9] Mingxing Tan and Quoc V. Le. 2021. EfficientNetV2: Smaller Models and Faster Training. arXiv:2104.00298 [cs.CV]

[10] RONG Yang and YAAKOV Bar-Shalom. 2022. Full State Information Transfer Across Adjacent Cameras in a Network Using Gauss Helmert Filters. *J. Advan. Inform. Fus* 17 (2022), 14–28.

[11] Alper Yilmaz, Omar Javed, and Mubarak Shah. 2006. Object Tracking: A Survey. *ACM Comput. Surv.* 38, 4 (Dec. 2006), 13–es. https://doi.org/10.1145/1177352.1177355

[12] Raed Abu Zitar, Amani Mohsen, Amal ElFallah Seghrouchni, Frederic Barbaresco, and Nidal A Al-Dmour. 2023. Intensive Review of Drones Detection and Tracking: Linear Kalman Filter Versus Nonlinear Regression, an Analysis Case. *Archives of Computational Methods in Engineering* (2023), 1–20.