Enhancing Airside Monitoring: A Multi-Camera View Approach for Aircraft Position Estimation for Digital Control Towers

Hasnain Ali, Duc-Thinh Pham, Sameer Alam
Air Traffic Management Research Institute
School of Mechanical and Aerospace Engineering
Nanyang Technological University, Singapore
{hasnain.ali, dtpham, sameeralam}@ntu.edu.sg

***Abstract*—A digital tower offers a cost-efficient substitute for traditional air traffic control towers and is anticipated to deliver video-based surveillance, which is especially beneficial for smaller airports. To fully unlock the potential of digital tower, sophisticated computer vision algorithms are pivotal for efficient surveillance. While current research predominantly concentrates on tracking aircraft movements on the airport surface, an equally crucial aspect lies in real-time monitoring of aircraft as they are are on finals. This capability plays a central role in enhancing both airport and runway operations. In this context, this study introduces a deep learning approach for precise estimation of the position of incoming aircraft, covering distances of up to 10 nautical miles. This approach surpasses the constraints of monoscopic techniques by leveraging multi-view video feeds obtained from digital towers. It combines Yolov7, an advanced real-time object detection model, with auxiliary regression and auto-calibration, allowing real-time tracking and feature extraction from different camera viewpoints. Furthermore, we propose an ensemble approach utilizing an Long Short-Term Memory model to combine input vectors, resulting in precise location estimation. Importantly, this method is designed to seamlessly adapt to different camera setups within digital towers. Its performance is evaluated using simulated video data from Singapore Changi Airport, showcasing stability in various scenarios with minimal predictive errors (Mean Absolute Percentage Error = 0.2%) over a 10 nautical mile range in clear weather conditions. These capabilities, when implemented in a digital tower setting, have the potential to significantly improve the controller's capacity to coordinate runway sequencing and final approach spacing, ultimately enhancing airport efficiency and safety remarkably.**

***Keywords*—Video-based surveillance; Computer vision; Multi-view video feeds; Location prediction.**

I. Introduction

Digital towers have emerged as a promising solution for replacing physical towers in small and medium-sized airports, and they are also being incorporated into the development of larger airports as digital twins in conjunction with their physical counterparts. Internationally, airports like London Heathrow and Budapest Ferenc Liszt Airport have adopted digital and remote tower systems, while Changi Airport is currently conducting trials for its Smart Digital Tower, exploring the benefits for enhanced safety and operational support as the airport expands [1]–[4]. These digital towers rely on video data captured by an array of cameras, which are expected to provide surveillance capabilities for airports without expensive radar systems and enhance their performance in terms of safety and efficiency. To fully provide these capabilities, a suite of computer vision techniques must be developed to leverage video streams for deriving decision-making information.

Recent years have seen a growing body of academic research focused on the application of computer vision algorithms in airport environments. These studies have explored a range of aspects, including aircraft tracking, airport surface surveillance [5]–[8], monitoring apron activities and aircraft turnaround processes [9], [10], and enhancing airport safety through debris and drone detection [11]–[13]. Notably, despite significant investigations in these areas, there remains a research gap in extracting aircraft location, particularly while tracking approaching aircraft, which can appear as small moving objects against featureless blue skies. The integration of such estimation capabilities within a Digital Tower environment has the potential to greatly improve runway controllers' sequencing and final approach spacing abilities.

Machine learning is increasingly used to track moving objects, employing two main methods in computer vision: monoscopic and stereoscopic. Monoscopic methods use a single camera for location estimation, employing object detection to identify objects using bounding boxes. This approach, supported by advanced algorithms like You Only Look Once (YOLO), has achieved a vision range of up to 1000m with different monocular cameras [14]. Another framework, DepthNet [15] estimated depth and detect objects from a single image, but faced challenges while detecting small objects (like approaching aircraft) and identifying crucial reference markers for long-distance estimation. Stereoscopic methods use multiple cameras to capture video, estimating location by comparing object or pixel disparities between these cameras. For example, [14] combined YOLO with stereoscopy for distance estimation. These methods work well with nearby stationary objects but face challenges in calibrating and aligning cameras, resulting in errors for distant objects [14].

Research studies on multi-camera approaches for vehicle tracking and speed estimation [16] have revealed the benefits of using multiple cameras for location estimation. Recently,

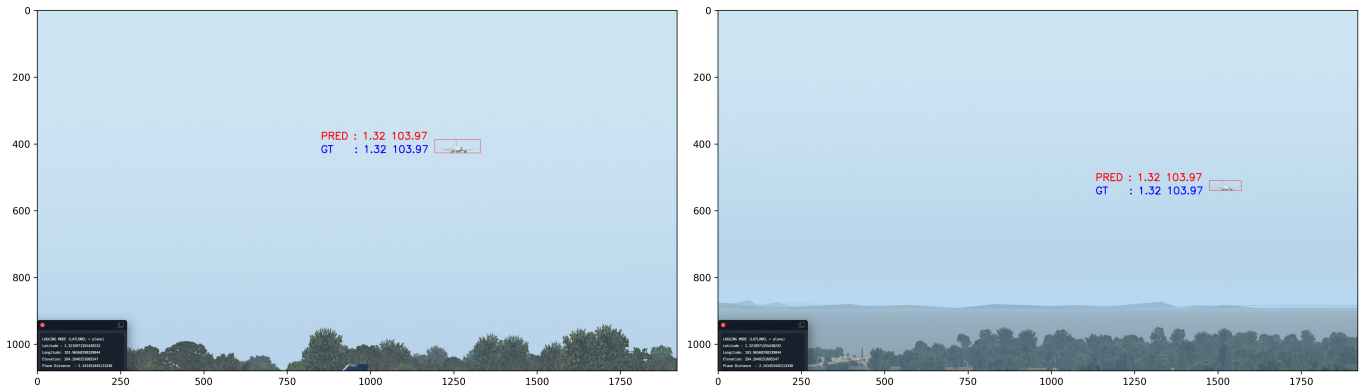


Figure 1. Aircraft location estimation: Predicted (PRED) and Ground Truth (GT) coordinates (Latitude, Longitude) of an approaching aircraft in two camera feeds within a digital tower context.

the study presented in [17] introduced a multi-view vision-based deep learning approach for estimating the Distance-to-touchdown of an approaching aircraft within a range of up to 10 nautical miles. Building upon the advancements outlined in [17], the current work extends this approach to provide real-time estimations of the WGS 84 (Earth-centered) coordinates (latitude and longitude) of approaching aircraft. This paper aims to contribute by proposing an approach that leverages multi-camera video feeds to ensure precise location prediction. Our approach introduces a model architecture that incorporates sequential layers and calibration to ensure stable performance and robustness against factors like noisy input and errors in object detection algorithms, effectively handling the stochastic nature of input video feeds. The model's efficiency is further boosted by leveraging a pre-trained object detection model and auto-segmentation techniques, reducing data requirements and training time while maintaining high accuracy in estimating the aircraft location up to 10 nautical miles. In its evaluation phase using simulated video data from Changi Airport (refer Figure 1), the model showcases exceptional location prediction accuracy.

II. The Proposed Approach for Aircraft Location Estimation

The proposed approach is visually outlined in Figure 2. This model comprises two key components: first, the extraction of the final feature vector from each camera view, and second, the ensemble component used for location estimation. This design is specifically engineered to address operational challenges, including situations where different airports have varying numbers of camera views for runway operations, or when there are potential alterations in the camera's configuration, such as changes in angle and zoom. In such scenarios, many end-to-end computer vision models typically require retraining or fine-tuning with new data to ensure consistent performance.

The model begins with the utilization of video feeds from two camera views as its inputs. These video sequences are processed through an auto-segmentation module to pinpoint the potential aircraft's position by employing an aircraft detection model. This step involves cropping out redundant areas in the video frames, eliminating unnecessary visual data and focusing solely on the small, distant approaching aircraft, which is the target of interest. Subsequently, the bounding boxes around the detected aircraft are fed into fully-connected layers, referred to as calibration networks, responsible for extracting the final feature vectors. All calibration networks are also linked to an auxiliary regression head to facilitate parameter training. This calibration step is crucial for harmonizing inputs from different camera views without necessitating manual system calibration. The resultant feature vectors are then merged using an Long Short-Term Memory (LSTM) model [18] in conjunction with fully-connected layers to predict the aircraft location. This sequential model effectively combines inputs from multiple camera views to ensure system stability, particularly in scenarios where there might be aircraft detection errors in one or more video feeds. Subsequent sections will delve into the model's architecture, implementation, and training in more detail.

III. Data Collection

TABLE I. The selected values of simulation parameters for data generation using the X-Plane 11 flight simulator.

Simulation Parameter	Selected Values
Airport	Singapore Changi Airport
Runway	02L
Aircraft Model	B737
Time of the day (5)	6:00, 8:00, 12:00, 17:00, 18:00
Weather condition (1)	Clear
Initial positions	Randomized with distance = 10NM
Number of Camera Views	2
Camera Resolution	1920 x 1280
Frame rate	30 FPS

In order to train the network effectively, it is essential to collect a substantial dataset that includes various perspectives of aircraft during their final approach. This dataset should have a high resolution to enable the identification of aircraft at long distances. Moreover, it should contain aircraft coordinates, which will serve as data labels for both training and evaluation.

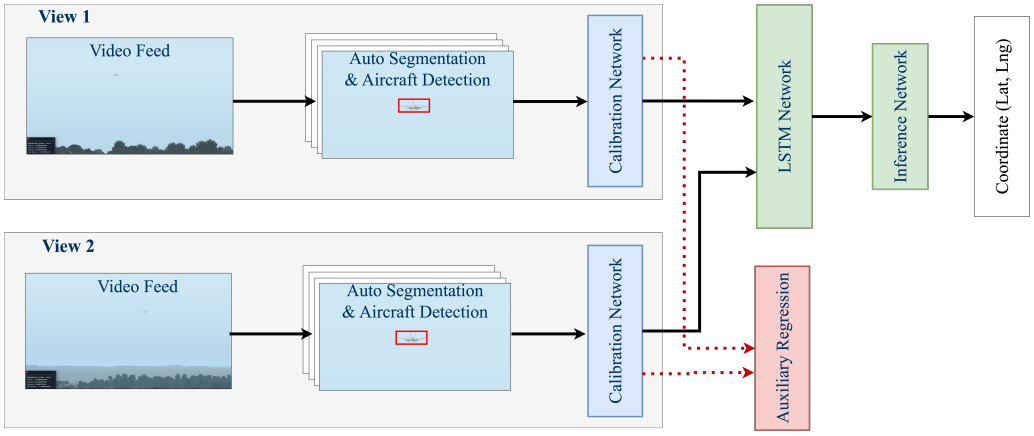


Figure 2. Proposed aircraft coordinate estimation approach involves two key components: extracting final feature vectors from each camera view and an ensemble for location estimation. It begins with processing video feeds for aircraft detection through an auto-segmentation module, followed by calibration networks extracting feature vectors. Merging these vectors using an LSTM model and fully-connected layers ensures accurate location prediction, enhancing system stability in scenarios with potential detection errors.

To meet these criteria, we have chosen to utilize X-Plane 11 by Laminar Research for data generation. To facilitate this data collection process, we have developed a Python-based tool that relies on the XPPython3 plugin [19]. This tool allows us to capture video feeds from specified camera positions and simultaneously record the aircraft's precise locations. Refer to Table I for a comprehensive overview of the controlled parameters involved in our data collection procedure.

The dataset comprises videos pertaining to 25 different scenarios, each depicting the corresponding 4D aircraft trajectories. These scenarios are created using the B737 aircraft model and the 3D model of Changi Airport, as illustrated in Figure 3. To ensure consistency and because of its widespread usage, we exclusively used the B737 aircraft model for data generation in this research. The scenarios are set in clear weather conditions with varying times of the day, and the aircraft's initial location is randomized to introduce diversity in the aircraft's landing positions while following the ILS (Instrument Landing System) guidance. It's worth noting that each scenario's trajectory spans approximately 10 nautical miles (NM), while the corresponding videos have a duration of around 4.0 minutes. These videos are recorded from two distinct camera views positioned on both sides of runway 20C, near its Instrument Landing System (ILS). It's important to highlight that, in the collected dataset used for training and testing, visibility conditions are generally favorable, with most scenarios having visibility exceeding 10 nautical miles.

IV. Experimental Setting

In this study, the proposed model is trained and tested using two camera views. In the training phase, 80% of the simulated video data (equivalent to 20 scenarios) is utilized, while the remaining 20% (5 scenarios) is reserved for testing. The data is collected over five different times of day (refer Figure 4). Data samples containing at least one detected aircraft are employed for both training and evaluation purposes, resulting in a total of approximately 124,000 data samples. Building on the

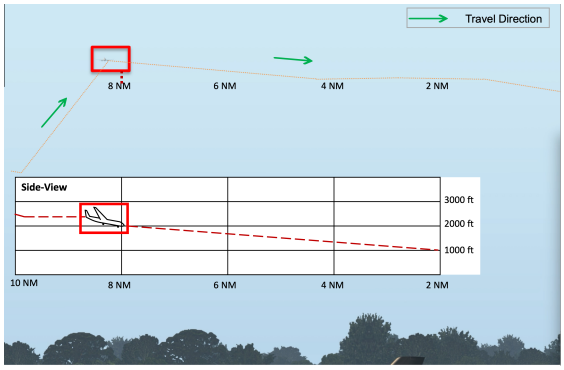


Figure 3. Illustration of an aircraft with its projected trajectory (side view) and the corresponding vertical profile (top). At approximately 8 nautical miles, the aircraft makes altitude adjustments before reaching the final approach fix, resulting in increased errors in position estimation.

approach outlined in [17], the learning algorithm incorporates YOLOv7 [20] as the aircraft detection model. Additionally, a stacked LSTM model is developed to combine the extracted information from all camera views. To optimize the model's inference speed, the TensortRT engine [21] is implemented for video processing and aircraft detection steps, yielding a significant processing speed increase of up to 300% compared to the model without the TensortRT engine. Furthermore, a series of experiments were conducted to evaluate different network architectures, resulting in the selection of Linear([256,2]) for the calibration network and [LSTM([256, 256],2), Linear([256, 2])] for the predictive model.

We evaluate the model's performance using five metrics. To understand the overall model performance during training, we use Mean Absolute Percentage Error (MAPE). This metric gives us an average measure of how well the model predicts aircraft locations with training iterations. To evaluate the trained model, we compute Latitude and Longitude Errors (degrees), along track error (meters), cross track error (meters) and euclidean distance (meters) between actual and predicted

Figure 4. Different time (local Singapore time) of the day scenarios from simulated videos with different lighting conditions.

aircraft location. These metrics help to evaluate how accurate the trained model is across different aircraft distances from touchdown. This project is implemented using PyTorch 1.13 with Python 3.10, and all the training is conducted on a single RTX 3090 GPU. The model takes approximately 10 hours to converge during the training process.

A. *Aircraft Detection*

The aircraft detection aims to localize the approaching aircraft in the video frame and determine its corresponding bounding box. This research selects a pre-trained YOLOv7 object detection algorithm [20] to reduce the necessary training data.

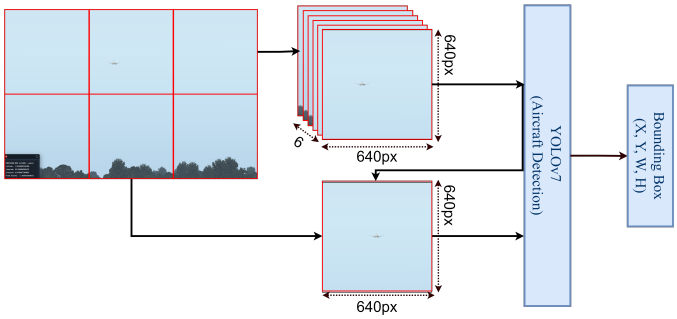


Figure 5. The Auto Segmentation process's concept diagram to localize airplanes in the video frame and extract corresponding bounding boxes.

The YOLOv7 model is a recent iteration of the YOLO family, designed for object detection in computer vision tasks. It enhances both accuracy and speed while being compatible with a small architecture suitable for single-GPU training. To adapt the model for our video frames (1920x1280x3), which differ in size from the YOLOv7 input (640x640x3), we employ a segmentation algorithm. This algorithm divides the original video frames into six non-overlapping image tiles (each 640x640x3) to feed into the pre-trained model (see Figure 5). When an aircraft is detected in any tile, we roughly estimate its location within the video frame, allowing us to extract the final image tile centered on the detected aircraft. This tile then undergoes another pass through the pre-trained YOLOv7 model to precisely identify the aircraft. This second pass resolves the issue when an aircraft crosses tile boundaries and isn't detected correctly (see Figure 6). We calculate the bounding box details, such as center location (X and Y) and size (W and H), corresponding to the original frame coordinates. The main purpose of this splitting or segmentation process is to ensure that even distant aircraft (up to 10 NM) remain sufficiently large in the image. If there are multiple aircraft in the frame, the same number of final images can be generated, and each goes through this process independently.

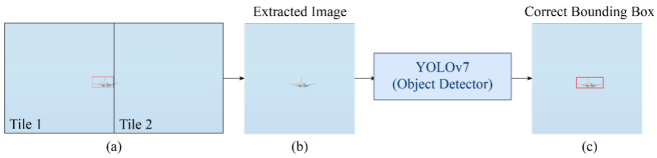


Figure 6. An example case where an airplane is situated between two different tiles. (a) the output from the first stage of the algorithm, (b) the extracted image from the second stage, and (c) the result with the accurate bounding boxes.

In this study, our goal is to predict aircraft locations in real-time, which requires efficient computing. We've fine-tuned the image-splitting method and data flow to achieve an average inference time of just 28 milliseconds. Instead of creating a custom aircraft detection model from scratch, we use advanced object detectors. This means we can easily swap out the object detector with other high-performance models for aircraft detection, and our approach should still perform well with minimal adjustments.

B. *Calibration Network*

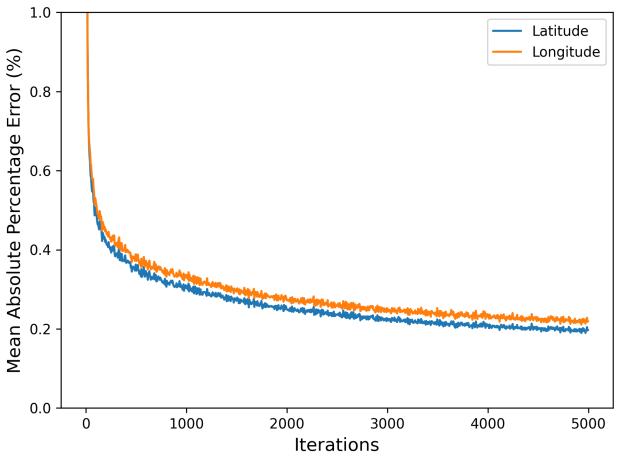


Figure 7. The convergence curve of the proposed adaptive algorithm for training the calibration layers of two camera views.

For each camera view, we train a calibration network to create feature vectors based on the detected bounding boxes. This accounts for the variations in camera perspectives. We also attach an auxiliary regression head with a reversed network structure to these calibration networks, aiming to

predict aircraft coordinates as the target. To ensure that the features generated by all calibration networks are not tied to any specific perspective, we include a regularization term in the loss function of each calibration network during the training of the auxiliary head (see Equation 1).

$$Loss_i = Loss(d_{Pred_i}, d_{Actual}) + Loss(d_{Pred_i}, d_{Pred_{1-i}}) \quad (1)$$

Here i refers to the i^{th} calibration network corresponding to either of the camera views in this study. This approach enhances the accuracy of the predicted location and lowers the computational expenses related to retraining or fine-tuning the entire system. So, when a new or adjusted camera input is introduced, we only need to train its calibration network while keeping the rest of the system as is.

C. Coordinate Estimator: LSTM and Inference Network

The Coordinate Estimator predicts the aircraft's location coordinates by combining the feature vectors from the Calibration Networks for all camera views, making the predicted coordinates more stable and accurate. This estimator consists of two parts: an LSTM and a fully-connected Inference network. The LSTM network in the Coordinate Estimator combines feature vectors from all cameras into a single representation. The proposed LSTM network has two hidden layers of 256 units each. LSTM networks are a type of recurrent neural network (RNN) known for their ability to capture long-range dependencies in sequential data, making them suitable for tasks involving temporal sequences. The LSTM's sequential design enables it to handle different numbers of cameras and maintain accuracy even if some cameras miss to detect objects. Subsequently, a fully-connected network predicts coordinates with layers of 256, 128, and 2 units in the first, second, and output layers, respectively. It's similar to the Auxiliary Regression Head but works on the unified feature vectors produced by the LSTM. During training, we randomly exclude feature vectors from one camera in each mini-batch. This ensures the model can still make sensible predictions even if some cameras miss to detect objects. In essence, this approach lets us merge information from all available camera views, boosting the overall accuracy and stability of location predictions.

V. Results and Discussion

The MAPE curves while training the proposed algorithm follow a similar pattern for both latitude and longitude prediction. They stabilize at around 0.2% for both latitude and longitude. We have found that the aircraft's bounding box position and size are closely linked to its location and is highly effective in accurately predicting the location for approaching aircraft. In the following text, we delve into the experimental results to evaluate the benefits of our approach.

The evaluation shows that our approach accurately predicts aircraft location (see Figure 8). The primary source of error in location estimation is along the track, while errors in the cross track are minimal (see Figures 9a and 9b). Up to a distance of

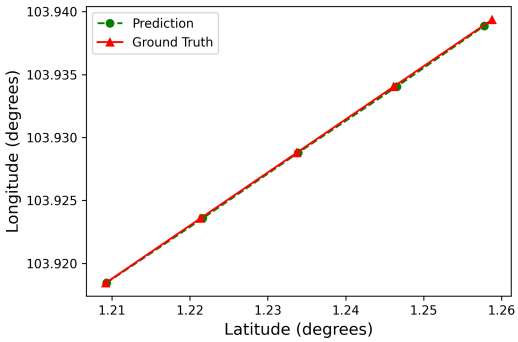
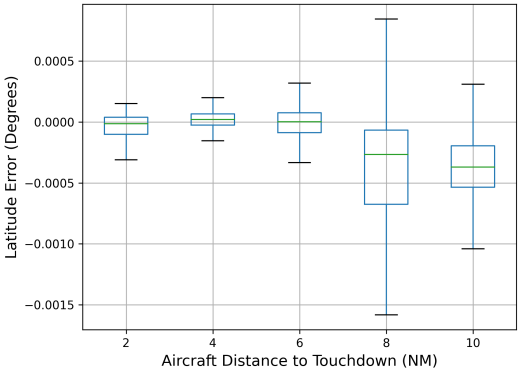
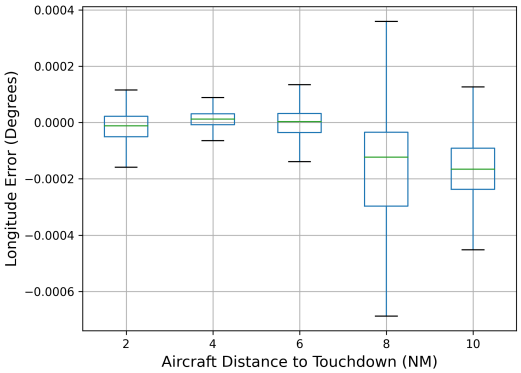


Figure 8. Comparing Predicted and Actual (Ground Truth) aircraft trajectories, our approach demonstrates high accuracy in forecasting the location of approaching aircraft.

6 nautical miles (NM), our approach performs exceptionally well, with median latitude/longitude errors close to zero (see Figure 10 and Table II).



(a) Actual vs Predicted Latitude



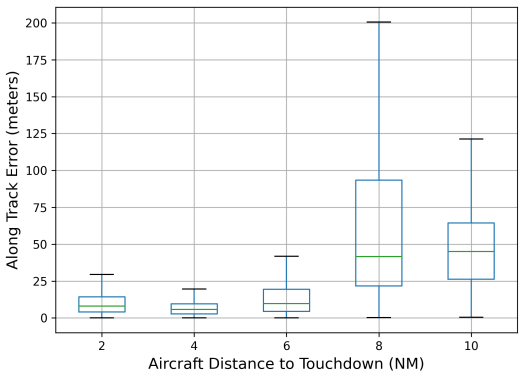
(b) Actual vs Predicted Longitude

Figure 10. The evaluation of the proposed method's performance using simulated Changi Airport data with respect to errors in latitude and longitude.

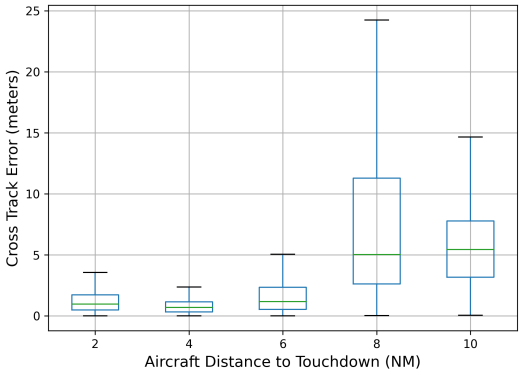
However, around 8 NM, the error increases briefly before decreasing again at around 10 NM. This corresponds to a Distance Measuring Equipment (DME) point at approximately 7.6 NM, as defined in Changi Airport's Instrument Approach Chart (AIC). Aircraft adjust their altitude at this point be-

TABLE II. EXPERIMENT RESULTS FOR COMPARISON BETWEEN THE ACTUAL AND PREDICTED AIRCRAFT LOCATION. METRIC VALUES ARE REPORTED AS MEAN ± STANDARD DEVIATION.

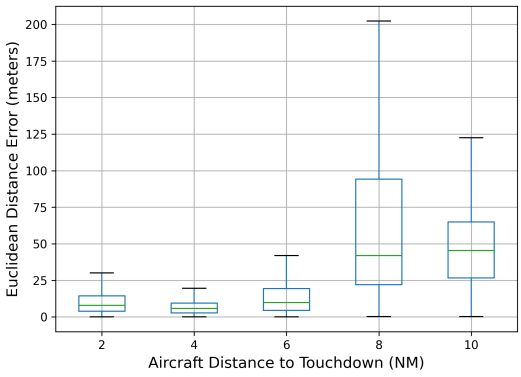
Distance to Touchdown (NM)	Latitude error (degrees in 10^{-5})	Longitude error (degrees in 10^{-5})	Along Track Error (meters)	Cross Track Error (meters)	Euclidean Distance Error (meters)
2	-6.1 ± 25.7	-1.5 ± 13.3	16.5 ± 28	2 ± 3.4	16.7 ± 28.2
4	2.1 ± 6.4	1.2± 2.8	6.6 ± 4.7	0.8 ± 0.5	6.6 ± 4.8
6	1.1 ± 18.5	0.6 ± 8.2	14.6 ± 16.8	1.8 ± 2	14.7 ± 16.9
8	-33.6 ± 60.1	-15.2 ± 27.9	62.8 ± 54.8	7.6 ± 6.6	63.4 ± 55.2
10	-35.8 ± 27	-16.2 ± 11.8	46.8 ± 26.8	5.6 ± 3.2	47.3 ± 27



(a) Along Track Error



(b) Cross Track Error



(c) Euclidean Distance Error

Figure 9. The effectiveness of the proposed method evaluated on simulated Changi Airport data. Along-track errors play a more significant role in contributing to Euclidean distance errors than cross-track errors.

fore crossing the final approach fix (see vertical profile of aircraft in Figure 3). During this period, the positions and sizes of bounding boxes in the video feeds are quite similar. Consequently, estimating location based on these detected bounding boxes leads to higher along-track errors, resulting in greater Euclidean distance errors between actual and predicted coordinates (up to 200m, see Figure 9c) than at other times. Overall, these results indicate that our approach is highly effective in accurately predicting the location of approaching aircraft.

VI. CONCLUSION

In this study, we introduce a multi-Camera view based deep learning method for estimating aircraft locations accurately up to a range of 10 nautical miles (10NM). Our approach is built to maintain stability and consistent performance, even when dealing with varying numbers of video feeds and noisy inputs or occasional miss-detections. To address potential changes in the camera system's setup, we introduce the calibration network and auto-segmentation. Using simulated data from Changi Airport, our approach achieves remarkable and consistent performance (MAPE = 0.2%) for aircraft approaching from distances of up to 10NM. In our approach, the key to location estimation lies in the positions of aircraft along their flight paths. Hence, the landing trajectory patterns captured in the videos play a crucial role in ensuring the model's accuracy. We plan to extend the model for different weather conditions and apply transfer learning, where the model can adapt to a real-time video feed by training on a combination of simulated and real data.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore.

REFERENCES

[1] D. Hughes, "Digital tower technology goes big with nats at heathrow," *Journal of Air Traffic Control*, 2019.
[2] J. Csengeri, "Remote towers ii." *Honvédségi Szemle–Hungarian Defence Review*, vol. 146, no. 1, pp. 159–175, 2018.
[3] H. Ali, P. D. Thinh, and S. Alam, "Deep reinforcement learning based airport departure metering," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 366–371.

[4] H. Ali, D.-T. Pham, S. Alam, and M. Schultz, "A deep reinforcement learning approach for airport departure metering under spatial–temporal airside interactions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 933–23 950, 2022.

[5] X. Zhang, H. Wu, M. Wu, and C. Wu, "Extended motion diffusion-based change detection for airport ground surveillance," *IEEE Transactions on Image Processing*, vol. 29, pp. 5677–5686, 2020.

[6] T. Van Phat, S. Alam, N. Lilith, P. N. Tran, and N. T. Binh, "Deep4air: A novel deep learning framework for airport airside surveillance," in *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2021, pp. 1–6.

[7] W. Li, J. Liu, and H. Mei, "Lightweight convolutional neural network for aircraft small target real-time detection in airport videos in complex scenes," *Scientific reports*, vol. 12, no. 1, p. 14474, 2022.

[8] X. Zhang, S. Wang, H. Wu, Z. Liu, and C. Wu, "ADS-B-based spatiotemporal alignment network for airport video object segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 887–17 898, 2022.

[9] Z. Lyu, D. Zhang, and J. Luo, "A GPU-free real-time object detection method for apron surveillance video based on quantized MobileNet-SSD," *IET Image Processing*, vol. 16, no. 8, pp. 2196–2209, 2022.

[10] P. Thai, S. Alam, N. Lilith, and B. T. Nguyen, "A computer vision framework using convolutional neural networks for airport-airside surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 137, p. 103590, 2022.

[11] X. Qunyu, N. Huansheng, and C. Weishi, "Video-based foreign object debris detection," in *2009 IEEE International Workshop on Imaging Systems and Techniques*. IEEE, 2009, pp. 119–122.

[12] M. Noroozi and A. Shah, "Towards optimal foreign object debris detection in an airport environment," *Expert Systems with Applications*, vol. 213, p. 118829, 2023.

[13] V.-P. Thai, W. Zhong, T. Pham, S. Alam, and V. Duong, "Detection, tracking and classification of aircraft and drones in digital towers using machine learning on motion patterns," in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2019, pp. 1–8.

[14] B. Strbac, M. Gostovic, Z. Lukac, and D. Samardzija, "Yolo multi-camera object detection and distance estimation," in *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*. IEEE, 2020, pp. 26–30.

[15] A. Masoumian, D. Marei, S. Abdulwahab, J. Cristiano, D. Puig, and H. A. Rashwan, "Absolute distance prediction based on deep learning object detection and monocular depth estimation models," in *Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, Artificial Intelligence Research and Development*, 2021, pp. 325–334.

[16] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J.-N. Hwang, "Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 108–115.

[17] D.-T. Pham, G. J. Goenawan, S. Alam, and R. Koelle, "Cleared to land a multi-view vision-based deep learning approach for distance-to-touchdown prediction," *15th USA/Europe Air Traffic Management Research and Development Seminar*, 2023.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] P. Buckner, "XPPython3 v3.1.5," https://xppython3.readthedocs.io/en/3.1.5/usage/installation_plugin.html, 2017, [Online; accessed 31-May-2023].

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[21] H. Vanholder, "Efficient inference with tensorRT," in *GPU Technology Conference*, vol. 1, 2016, p. 2.