Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)

Cleared to Land
A Multi-view Vision-based Deep Learning Approach for Distance-to-TouchDown Prediction

Duc-Thinh Pham*, Gabriel James Goenawan†, Sameer Alam*
Air Traffic Management Research Institute
School of Mechanical and Aerospace Engineering
Nanyang Technological University, Singapore
*{dtpham,sameeralam}@ntu.edu.sg, †gabr0022@e.ntu.edu.sg

Rainer Koelle
Operational ANS Performance
Performance Review Unit
EUROCONTROL
Email: rainer.koelle@eurocontrol.int

***Abstract*—With the broader adoption of digital air traffic control towers, real-time video data is expected to complement the current surveillance system (if available) and improve airport performance in terms of safety and efficiency. However, to fully utilize such data, a suite of computer vision algorithms needs to be developed for extracting useful information from real-time video feeds. Currently, most of the studies in the literature have focused only on the detection and tracking of aircraft on the airport surface, while approaching aircraft also play an essential role in airport and runway operations. The distance-to-touchdown of approaching aircraft is a critical parameter in final approach spacing and departure sequencing. Therefore, this research proposes a deep learning approach for estimating the distance of approaching aircraft to touchdown using multi-view video feeds. The proposed approach adopts a state-of-the-art computer vision model with an auto-calibration technique for detecting the approaching aircraft and extracting feature vectors from multiple camera views under various lighting and weather conditions. Then, an ensemble approach is introduced for combining the input vectors for distance estimation. The approach is evaluated with both Changi Airport simulated and real video data. Firstly, the proposed approach is designed to be easily updated and adapted for different camera system configurations. Secondly, the proposed approach has successfully combined the strength of both monoscopic and stereoscopic approaches to provide accurate distance-to-touchdown prediction in various scenarios. The experimental results demonstrate the advantages of the proposed approach with stable performance and low predicted errors (Mean Absolute Percentage Error = 0.18%) in estimating the distance-to-touchdown up to 10 NM. Such capability in a Digital Tower environment can augment the runway controller's sequencing and final approach spacing capabilities.**

***Keywords*—Digital Tower, distance estimation, multi-view cameras, distance-to-touchdown estimation, runway operation.**

I. Introduction

Digital towers rely on video data from an array of cameras which are also expected to complement the available surveillance system (if available) and improve the airport performance in terms of safety and efficiency. Digital towers are considered a promising solution to replace physical towers for small and medium airports. The efficient utilization of video data holds the potential to provide surveillance capabilities for small and medium airports without the need for complex surveillance systems like Advanced-Surface Movement Guidance and Control System (A-SMGCS). They are also part of the new development of big airports as a digital twin besides the physical ones. Several studies have demonstrated the advantage of multi-sensor data in managing airport operations by providing better situational awareness of the air traffic movements on the ground and in the final approach phase [1]. Moreover, an exciting investigation [2] has figured out that augmenting airport situation/information directly on the screens is expected to reduce the workload of controllers as well as improve their performances. Therefore, by extracting useful information from video streams, it becomes possible to enhance the current surveillance system, particularly by augmenting tracking information on screens of digital towers. To efficiently perform that task, a suite of computer vision algorithms should be investigated and developed using video streams from digital towers. Over the last few years, several studies can be found in the literature for applying computer vision in the airport environment, such as aircraft tracking and airport surface surveillance [3]–[6]; airport apron and aircraft turnaround process monitoring [7], [8]; or airport safety,e.g., debris and drone detection [9]–[11].

This study proposes a novel computer vision approach on video feeds of aircraft on the final approach for estimating the distance-to-touchdown (DTD) in real-time. Distance-to-touchdown is a critical parameter in final approach spacing and departure sequencing. Such capability in a Digital Tower environment can augment the runway controller's sequencing and final approach spacing capabilities.

The real-time distance estimation of the moving object is an active research topic and gains more and more attention, especially with the advance in deep learning technologies. Studies in literature for computer vision-based distance estimation can be broadly classified into two main categories: stereoscopic and monoscopic view. With careful calibration,

National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore.

the stereoscopic approaches use two cameras to capture the video data. The distance is estimated by calculating the disparity of the objects (or pixels) between two camera screens. However, the biggest challenge of stereoscopy is to calibrate and match the cameras appropriately [12]. Any errors in the calibration and pixel matching can lead to significant errors in the result. Additionally, the estimation error will become much more significant if the object is far away. Several studies can be found using this research direction in the literature, especially on the street view dataset. For instance, the study in [13] proposed a fast and accurate algorithm using stereo data to recover dense depth from stereo video under the assumption that the scene is static. With the introduction of a high-performance object detection algorithm, e.g., You Only Look Once (YOLO [14]), several following studies started to utilize object detection as an intermediate step for distance or depth estimation. An example can be found in [12], in which the authors presented a distance estimation solution based on the YOLO deep neural network and principles of stereoscopy. These approaches demonstrate good results but only for static and near objects.

In contrast, the monoscopic approaches use a single camera for performing the task. Since the depth information cannot be recovered directly, studies in this direction are usually based on object detection techniques for identifying the object with the corresponding bounding box. The detected bounding box with the object's size and referenced markers are used for estimating the object's distance from the camera. For instance, an straightforward approach using YOLO is presented in [15]. The approach is evaluated in different environments and provides good results using different monocular cameras. The vision range can go up to 1000m. Another work in this direction, called DepthNet [16], presented a more complicated deep learning framework that consists of two deep networks for depth estimation and object detection using a single image. However, the training for DepthNet is quite challenging to achieve a considerable good accuracy, especially facing long-distance estimation. The main challenge is to accurately detect the bounding box of small objects and identify referenced markers on the scene, which is challenging in the problem of approaching aircraft. Recently, an interesting study for inter-camera (multi-views) has been presented in [17], focusing on vehicle tracking and speed estimations. This work demonstrates the potential of combining multiple cameras to improve the distance estimation performance.

In summary, even though advanced deep learning algorithms have demonstrated several successes with street view and in-door datasets, there isn't a universal approach that can solve all the problems, especially when it comes to a featuring-less blue sky. Moreover, as this problem requires high accuracy for estimating the distance (up to 10NM) of small moving objects, novel approaches are needed to be investigated and proposed to achieve the required performance. Inspired by the state-of-the-art studies, in this work, we propose a multi-view vision-based deep learning approach for estimating the Distance-to-touchdown of approaching aircraft using multi-camera video feeds to combine the strength of those approaches. The main contributions of this paper are as follows.

- The model's architecture, e.g., calibration layers and sequential layers, is designed to provide stable operation and performance with the stochastic numbers of input video feeds due to noisy input or errors in object detection algorithms.
- The model is evaluated using both Changi Airport simulated and real video data. It can achieve high performance in challenging scenarios such as low visibility, stormy or low light.
- Using the pre-trained model for object detection with an auto-segmentation helps to reduce the amount of data and time during model training, as well as to achieve high accuracy in estimating the DTD up to 10 NM.
- The calibration network, trained with an auxiliary regression head, is proposed for tackling the potential changes in the camera system and its configuration.

In this paper, the motivation for a multi-view vision-based approach is discussed in Section I. In Section II, we will describe our proposed approach with an illustrated concept diagram. The experimental setting and data collection are mentioned in Section IV, and Section V is for results and discussion. Moreover, before the Conclusion session VII, a preliminary result for Changi Airport data is also discussed in Section VI.

II. Overview of The Proposed Approach

The concept diagram of the proposed approach is presented in Figure 1. The model has two main parts: the final feature vector extraction from each camera view and the ensemble method for estimating the DTD. Noting that it is designed this way to tackle two of the technological and operational challenges:

- Digital towers at different airports have their own suitable camera configurations and a different number of cameras for runway operation.
- Many factors can cause changes in a camera's configuration in terms of rotated angle, tilted angle, and zoom level. In those cases, most of the end-to-end computer vision models are needed to be retrained or fine-tuned with the newly collected data to maintain their performance.

First of all, video feeds from N ($\geq$1) camera views are utilized as the inputs of the model. To obtain the final feature vector for each video feed, the sequence of images is input into the auto-segmentation module for localizing the potential aircraft position using an aircraft detection model and cropping the redundant video frames' areas. This step helps remove unnecessary information in the image and keep the sufficient size of the approaching aircraft, which, by design, is far away and very small. Then the bounding boxes of the detected aircraft are input into the fully-connected layers, called calibration network, for extracting the final feature vectors. All the calibration networks are connected to an auxiliary regression head for training their parameters. This step is necessary for adjusting inputs from different camera views without requiring careful system calibration.

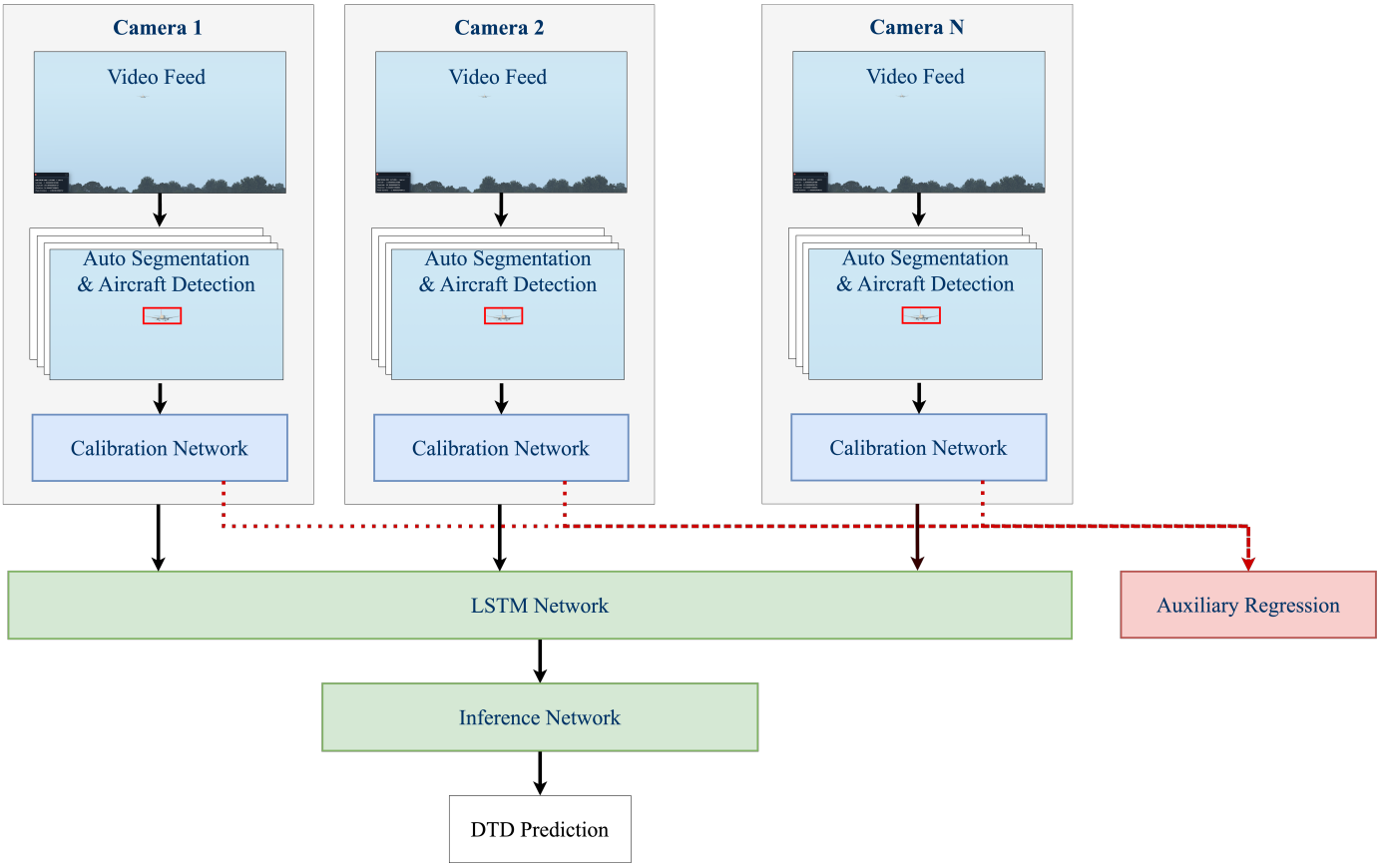

Figure 1: The concept diagram of the proposed approach for multi-view vision-based DTD Prediction.

The feature vectors are combined using a Long Short-Term Memory (LSTM) [18] model and fully-connected layers for distance prediction. The sequential model is utilized for combining the multi-view camera inputs to provide the system's stability in case of aircraft detection errors in each video feed. The model's architecture, implementation, and training will be further discussed in the following sections.

III. Data Collection

The simulated dataset is generated from X-Plane 11 flight simulator [19] due to four main reasons. First of all, it has very accurate aircraft models. Secondly, it supports setting the camera position and angle for data collection. Thirdly, it can adjust lighting and weather conditions to diversify the data. Finally, it also allows exchanging data with an external system.

TABLE I presents the values of the controlled parameters in our data collection process. The dataset contains videos for 80 scenarios with the corresponding 4D aircraft trajectories, using aircraft model B737 and Changi Airport 3D model (refer to Figure 2). The lighting (time of the day) and weather conditions are adjusted to cover scenarios with different visibility, while the initial randomized location is utilized to create the variation in aircraft position during landing. Noting that, for each scenario, videos from two different camera views are collected for training and testing the proposed model. Finally, the visibility in the collected dataset is mostly more than 10NM for training and testing, except in the case of "foggy", where it is designed to reduce the visibility down to 5NM for model evaluation.

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 (4)	Clear, Cloudy, Stormy, Foggy
Initial positions	Randomized with DTD = 10NM
Number of Camera Views	2
Camera Resolution	1920 x 1280
Frame rate	30 FPS

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

IV. Experimental Setting

In this work, the proposed model is trained and tested with two camera views. Using the simulated video data, 70% of the data (56 scenarios) is used for training, and the remaining data (24 scenarios) is used for testing. To facilitate the training, data samples that can successfully detect at least one aircraft will be used. For testing or real-time running, a filter function is added before the DTD Predictive Network to check and drop all cases with no aircraft detection. In total, the training data includes 36k data points.

In terms of the model's architecture, the YOLOv7 [20] is adopted as the aircraft detector over each camera view. And

Figure 2: The figure presents examples of four scenarios from simulated videos with different lighting and weather conditions. Each scenario is demonstrated by two images from different camera views: (1) clear weather at 6:00, (2) clear weather at 18:00, (3) cloudy weather at 6:00, and (4) cloudy weather at 18:00.

to combine the extracted information from all the cameras, a stacked LSTM model is developed. More importantly, to reduce the inference time of the proposed end-to-end model, we adopt the TensortRT engine [21] for video processing and aircraft detection steps. As experimented, the processing speed increased up to 300% compared to the model without the TensortRT engine.

The two metrics are utilized for analyzing the model performance, Percentage Error (PE) and Mean Absolute Percentage Error (MAPE). The PE reports the difference between the actual distance (A_i) and the predicted distance (P_i) for each predicted distance instance (i). It is useful for observing the changes in accuracy over variations of the aircraft distance. On the other hand, the average model performance is assessed using MAPE which is generally suitable for model comparison given the dataset with size n.

$$PE_i(\%) = \frac{A_i - P_i}{A_i} * 100 \tag{1}$$

$$MAPE(\%) = \frac{1}{n} * \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right| * 100 \tag{2}$$

Finally, this project is implemented using PyTorch 1.13 with Python 3.10, and all the training is done on a single RTX 3080. The total training time for the model convergence is 1.5 hours.

A. Learning Algorithms

1) Aircraft Detection: The aircraft detection aims to localize the approaching aircraft in the video frame and determine its corresponding bounding box using a segmentation algorithm, complementing the pre-trained YOLOv7 model (refer to Figure 3). As the pre-trained model is trained on MS COCO Dataset [22], containing images of complex everyday scenes of common objects in their natural context, with 640x640x3 image size, the high-resolution video frame (1920x1280x3) is split into six non-overlapping images (640x640x3) for aircraft detection. Once an aircraft is detected in any image, a final image (640x640x3) with that aircraft at the center is extracted and used for estimating the bounding box of the detected

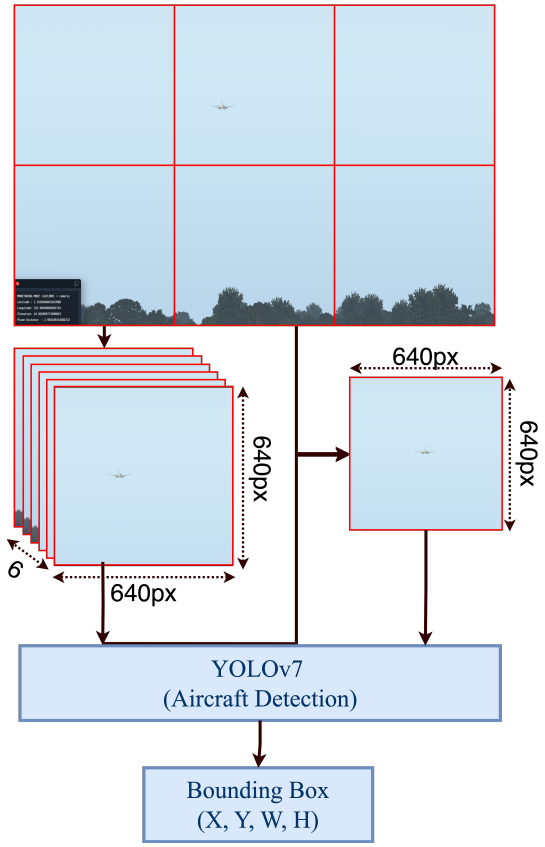

Figure 3: The illustration of the segmentation step for aircraft detection and extracting the corresponding bounding box.

aircraft. The bounding box information, e.g., location (X and Y) and size (W and H), is calculated corresponding to the coordinates in the original frame. The main purpose of the splitting or segmentation is to keep the sufficient size of the far-away aircraft (up to 10 NM) in the image. With more than one aircraft in the frame, the same number of final images can be generated and follow the same process independently.

Because of the fast pace of technological progress in object detection, this approach is designed to make use of state-of-

the-art pre-trained models rather than emphasizing the need for training or tuning a specific model for aircraft detection. As the state-of-the-art model can be easily replaced by a better model in the future, the proposed approach is expected to maintain high and stable performance.

2) An Adaptive Algorithm for Training Calibration Networks: One of the computer vision model's typical limitations is the camera configuration sensitivity, especially when working with multiple cameras. Since the central idea of our approach is the ensemble of multi-view videos for stabilizing the model performance, it must be able to handle the change in the number or the configuration of cameras without the need to retrain the whole model. Therefore, a calibration network is proposed for each camera. It is fully-connected layers designed to construct the feature vectors from detected bounding boxes considering the differences in camera configuration.

For that purpose, the auxiliary regression with reversed network structure is connected to calibration networks for training using DTD values as the target and MAPE as performance metrics. As the values and qualities of inputs from each camera can have a different impact during training, the converged models can have a significant gap between each camera in terms of errors for estimating the DTD. Thus, to further increase the quality of the calibration networks, we propose an adaptive algorithm (Algorithm 1) during the training. The convergence curve of the training process can be observed in Figure 4. As the network of Camera 2 converges much faster, and the gap is significant, it is frozen to focus the training on the other network until the performance gap is less than a defined threshold. This process is repeated until both networks are converged with the desired performance gap. The outputs of the calibration networks are the feature vectors used as the input for the distance estimation model.

With this approach, when a new or adjusted camera input is added to the system, only its calibration network is needed to be trained with the frozen auxiliary regression head while the whole system is kept unchanged.

3) DTD Predictive Network: The DTD Predictive Network includes a stacked LSTM network for combining input vectors from individual calibration networks and a fully-connected network for inference. As discussed, a sequential model (e.g., LSTM) is necessary to handle the variation in the number of input cameras in the system. Moreover, even with the same number of cameras, it is also helpful to maintain the model performance by skipping miss-detection in any camera input.

Algorithm 1: The Proposed Adaptive Algorithm.

```
current_loss_list = empty array of N_camera length
loss_list_tracker = empty array
all_training_tracker = 0
while epoch < total_epochs do
    if training_mode == 'all' then
        for n = 0 to N_camera do
            Predict_DTD_from_Camera (n);
            current_loss_list[n] = Compute_Loss(n);
            Optimize(n, aux_reg);
        end
        all_training_tracker = all_training_tracker + 1;
    else
        n = training_mode;
        Predict_DTD_from_Camera (n);
        current_loss_list[n] = Compute_Loss(n);
        Optimize(n);
    end
    if (epoch % 500) == 0 then
        /* Only switch to targeted
           training when the training
           loss is stable .          */
        if (training_mode == 'all') &
         (std(loss_list_tracker[-3:]) < 0.01) then
            training_mode =
             Argmax(current_loss_list);
        else
            /* Control the frequency of
               mode switching by
               all_training_tracker     */
            if (Max_Gap(current_lost_list) < 0.01) &
             (all_training_tracker > 1500) then
                training_mode = 'all';
                all_training_tracker = 0;
            end
        end
        loss_list_tracker.append(current_loss_list)
    end
end
```

B. Model Selection

Different network architectures have been explored and experimented with to obtain the final model for further experiments and analysis. The configurations of the nine tested models are reported in TABLE II. There are three sizes for the Calibration Network, and corresponding to each option, three networks with different sizes are selected for DTD Predictive Network. Two types of networks have been considered, which are fully-connected layers or linear networks and stacked LSTM layers. The linear network is represented by a list with an input size, a list of hidden layers' sizes, and an output size. On the other hand, an LSTM network (e.g., [LSTM, [256, 256], 2]) has information about the input size, hidden layer size, and the number of layers.

Noting that the more parameters the model has, the more computational time it requires. The results of those models in terms of MAPE(%) and Inference Time(ms) are presented in Figure 5. The medium and large Calibration Networks provide lower MAPE values than the small network while only increasing by 0.1 to 0.2 ms of inference time. Based on the experiment results, the medium network size is chosen in the final model for both Calibration Network and DTD Predictive Network.

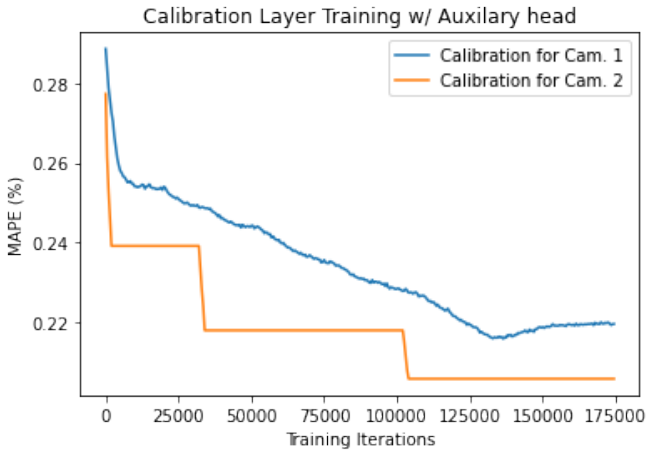

Figure 4: The convergence curve of the proposed adaptive algorithm for training the calibration layers of two camera views.

Calibration Network	Predictive Network	Total Parameters (2 Cameras)
Small Linear, [4,128,128]	Small [LSTM, [128, 64], 1] [Linear, [64, 32, 1]]	94,402
	Medium, [LSTM, [128, 128], 2] [Linear, [128, 64, 1]]	315,138
	Large [LSTM, [128, 256], 3] [Linear, [256, 128, 1]]	1,523,586
Medium Linear, [4,128,128,256]	Small, [LSTM, [256, 128], 1] [Linear, [128, 64, 1]]	339,330
	Medium, [LSTM, [256, 256], 2] [Linear, [256, 128, 1]]	1,219,074
	Large, [LSTM, [256, 512], 3] [Linear, [512, 256, 1]]	6,044,418
Large Linear, [4,256,256,512]	Small, [LSTM, [512, 256], 1] [Linear, [256, 128, 1]]	1,350,402
	Medium, [LSTM, [512, 512], 2] [Linear, [512, 256, 1]]	4,862,978
	Large, [LSTM, [512, 1024], 3] [Linear, [1024, 512, 1]]	24,147,458

TABLE II. The experimented nine model architectures for model selection and hyper-parameter tuning.

C. Monoscopic Model for Comparison

In this work, a monoscopic model, similar to [15], for DTD prediction is also developed for benchmarking. Since the videos are collected from two different camera views, merging those two sets of videos directly for training the monoscopic model has a negative impact on the model performance. Two options have been considered, which are (1) developing two independent models to achieve their average performance or (2) including a calibration network for developing one unique model on the combined dataset. The experimental results demonstrate insignificantly different performance between those two approaches. Therefore, for the monoscopic model,

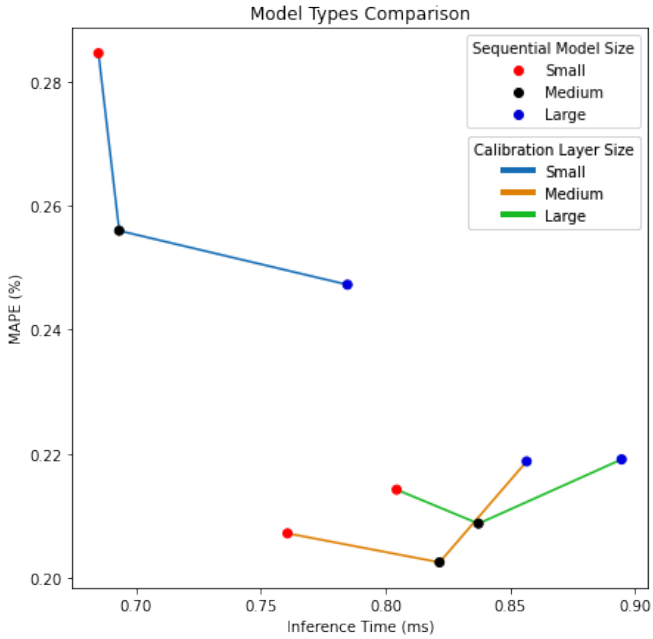

Figure 5: The results for hyper-Parameter tuning step with a variation in model sizes.

a combined video dataset is used. The model's architecture includes the feature extraction component (e.g., Segmentation, YOLOv7, and Calibration Network) and a fully-connected Inference Network.

V. Results and Discussion

First of all, we observed a significant number of miss-detection for either camera views (up to 40%), even with the clear weather condition, which comes from the limitation of the aircraft detector for detecting small objects under various lighting and weather conditions. Secondly, the aircraft positions (bounding boxes' locations) along their approaching trajectories (refer to Figure 6) are essential in estimating the DTD since their changes are significant compared to the changes in bounding box sizes. Moreover, to better assess the proposed approach's advantages, the experimental results are reported and discussed in the rest of this session.

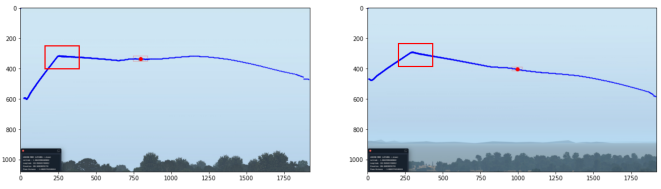

Figure 6: An example of an aircraft trajectory from two camera views in the dataset. The red dot illustrates the current position of the aircraft along the trajectory. The red rectangle highlights the trajectory segment corresponding to the DTD range around 5NM to 7NM.

The proposed approach's and monoscopic model performance is presented in Figure 7. Up to 5NM, the proposed approach achieves high performance with median errors close

to zero and a small standard deviation. On average, the performance of both models is comparable from 8NM to 10NM, which is expected due to the limitation of the pre-train detection model for small flying aircraft. Suddenly, within 5NM and 8NM, the errors become significantly higher for both models. As defined in Changi Airport's Instrument Approach Chart (AIC), there are two Distance Measuring Equipment (DME) points at 4.4NM (also the Final Approach Fix (FAF)) and 7.6NM, the aircraft would start to descend and adjust their altitude between those points and due to the camera angle, the observed trajectories are as illustrated one in Figure 6. It can be observed from the video feeds that, during that period, the positions and sizes of bounding boxes were indifferent. Thus, estimating distance based on those detected bounding boxes leads to significant big errors (up to 2% or 250m) compared to the other periods. However, our model still obtains more stable performance compared to the high errors and variations of the monoscopic model. In general, the proposed approach achieves smaller errors and more stable results. The MAPE (0.18%) reduces by 35% compared to the MAPE = 0.28% of the monoscopic model.

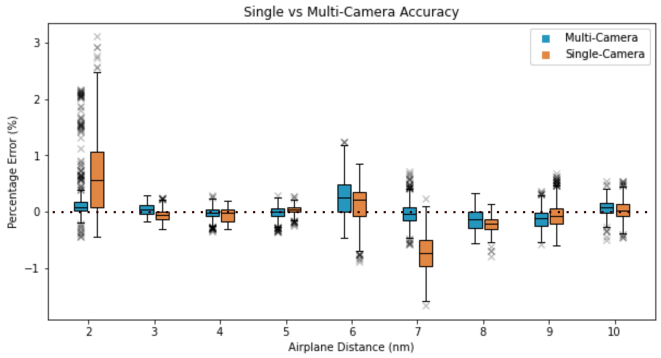

Figure 7: An experiment results for comparison between the proposed model (Multi-Camera) and the monoscopic model (Single-Camera).

TABLE III shows the impact of solely lighting conditions on prediction accuracy. As mentioned, the scenarios at 6:00 and 18:00 are considered in low light conditions, while from 8:00 to 17:00, the light is much more intense. It is interesting to observe a better performance of the model in low light conditions. Our further inspections and analyses of those cases suggest that the reflected glare and shadow have affected the accuracy of the bounding box estimation.

Time of the day	6:00	8:00	12:00	17:00	18:00
MAPE(%)	0.162	0.213	0.205	0.242	0.174
std(%)	0.250	0.287	0.263	0.273	0.207
Visibility(NM)	10	10	10	10	10

TABLE III. The model performance in different lighting and clear weather conditions.

The weather also has a strong impact on the model's performance. An example of a foggy scenario when the aircraft's DTD is 3.09NM is presented in Figure 8. It is very challenging for any model to detect an aircraft significantly

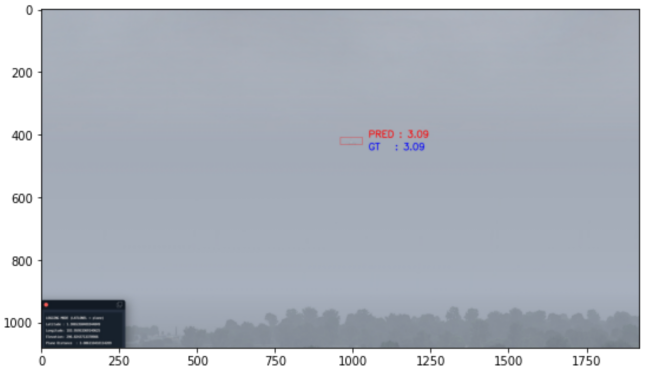

Figure 8: A snapshot in a foggy scenario (low visibility) and aircraft is 3.09NM away from the runway threshold (in blue color). The predicted DTD (in red color) can be augmented in the image in real-time for demonstration.

small and extremely blurred due to the fog. Therefore, the following analysis focuses on the performance of the proposed model under four weather conditions (refer to Figure 9). The figure shows that the model achieves high performance within 5NM (the median PE near 0 and small standard deviation). However, with DTD between 6NM and 8NM, the cloudy condition significantly impacts the model performance, especially at 6NM and 7NM. TABLE IV summarizes the experimental results, in which the MAPE for the cloudy condition (0.346%) is much higher than the others. And the smallest MAPE for foggy scenarios reflects the high performance of the proposed model for DTD within 5NM. The main reason for that is the inconsistency of the sky due to clouds, which reduce the accuracy of aircraft detection and the bounding box estimation. Noting that the heavy rain is simulated for stormy conditions, the sky is dark but much more consistent than in cloudy conditions.

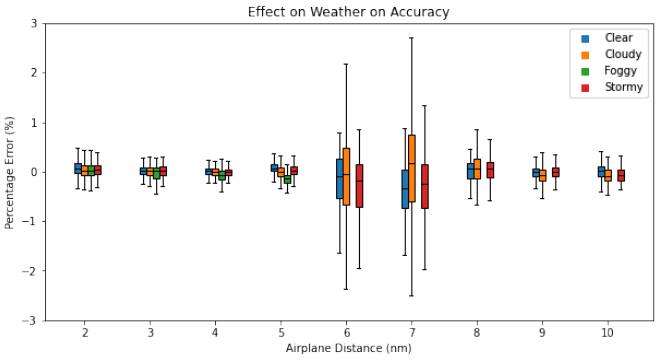

Figure 9: The model errors correspond to the aircraft distance in four weather conditions (Clear, Cloudy, Foggy, and Stormy). Noting that the maximum visibility of foggy is 5NM, thus, the results for foggy scenarios are only reported up to 5NM.

Weather	Clear	Cloudy	Foggy	Stormy
MAPE(%)	0.213	0.346	0.162	0.228
std(%)	0.287	0.624	0.235	0.306
Visibility(NM)	10	10	05	10

TABLE IV. The model performance in different weather conditions.

VI. Case Study of Changi Airport

A. Video Data from Changi Airport

A dataset with 100 landing trajectories for Runway 02L of Changi Airport is used in this work. The sky is cloudy during the data collection; thus, all of the collected videos are under cloudy conditions. Two camera views for each landing trajectory are recorded. View 1, including two cameras, is the view from the control tower, and View 2, including three cameras, is from the instrument landing system localizer (LLZ) hut. Besides, the visibility in this dataset is less than 7NM due to the weather condition. The raw videos from different cameras for each view are processed and merged to create the final dataset. The example of videos from those views in the final dataset can be observed in Figure 10. Moreover, the aircraft in the video data are also needed to be matched with their respective recorded trajectories from the surveillance system for obtaining the DTD information.

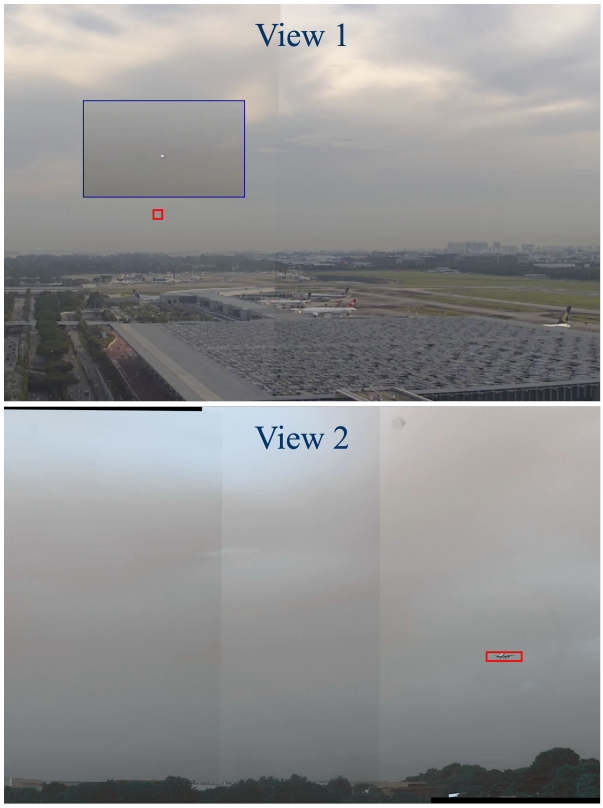

Figure 10: An example of video data from Changi Airport. View 1 is from the camera mounted on the top of the control tower, while View 2 is from the camera at the LLZ hut. The red rectangles highlight the position of the aircraft in their respective frames. The image segment with a blue border in View 1 is the zoom-in for better observing the small aircraft.

B. Results

The proposed model is trained and evaluated with the above dataset (70/30). The results are consistent with the finding and model performance of the model trained by simulated data.

First of all, View 2 from LLZ hut can clearly capture the landing trajectory (refer to Figure 11), which is necessary for estimating DTD. However, the aircraft is too small in View 1, and all the videos are also under cloudy conditions; thus, the miss-detection rate for videos from View 1 is very high (>90%). Thus, in most of the data points for training and testing, only data from View 2 is available.

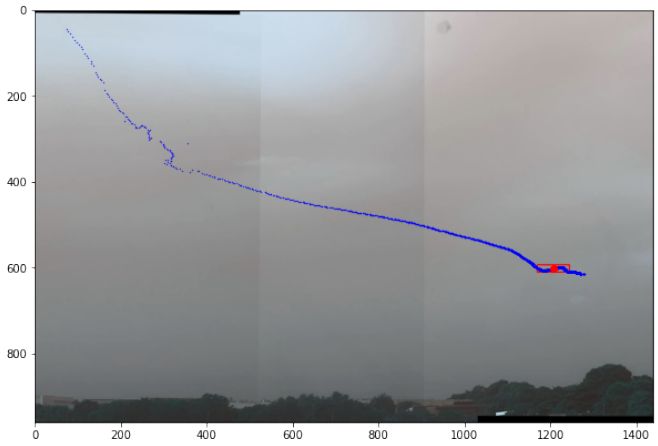

Figure 11: An example of an aircraft trajectory from camera view 2 (from LLZ hut) in the Changi Airport data. The red dot illustrates the current position of the aircraft along the trajectory.

Secondly, our model achieves good performance under cloudy conditions with MAPE: 0.33% (or $< 45m$) and std: 0.42%. Besides, model accuracy over the DTD can be observed in Figure 12. The predicted errors from 4NM to 6NM are observably higher than errors at the other distances, which is similar to our observation in Session V.

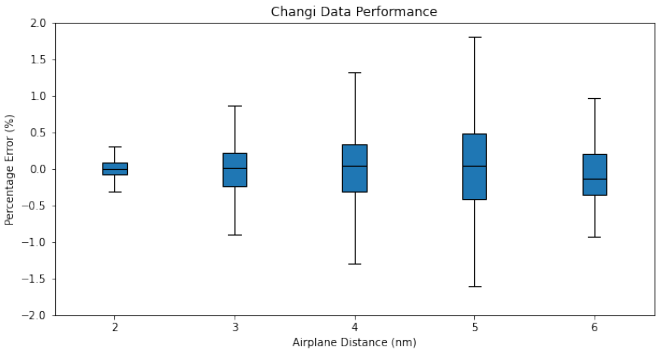

Figure 12: The performance of the proposed approach on real Changi Airport data.

VII. Conclusion and Future Work

In this work, we propose a multi-view vision-based deep learning approach for Distance-to-touchdown (DTD) estima-

tion up to 10NM under various lighting and weather conditions. The approach is designed to provide stable operation and performance with the stochastic numbers of input video feeds due to noisy inputs or miss-detection. In which the calibration network and the auto-segmentation are proposed for tackling the potential differences and changes in the camera system's configuration. The proposed approach can achieve high and stable performance with Changi Airport simulated data (MAPE = 0.18%) and real data (MAPE = 0.33% under the cloudy condition) for DTD up to 10NM. It also demonstrates a more stable performance than the monoscopic model, which solely relies on the input from one camera view. In this kind of approach, the aircraft' positions along their trajectories are the key features in the DTD estimation. Therefore, the pattern of the landing trajectories captured in the videos is a factor to be considered to ensure the model's performance. Besides, the lighting and weather conditions add a lot of challenges and uncertainties to the video dataset and have strong impacts on the predictive accuracy.

Besides, a new set of Changi Airport data is being collected with adjusted View 1 for better visibility. The new dataset will also cover more lighting and weather conditions. In terms of model development, a multi-object tracking algorithm will be developed for a complete end-to-end DTD prediction model. Finally, the auto-calibration step will be updated, and more experiments will be conducted with Changi Airport data, where the model can be trained with both the simulated and real data for quickly adopting and operating in a real airport environment.

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] R. L. Sturdivant and E. K. Chong, "Systems engineering baseline concept of a multispectral drone detection solution for airports," *IEEE Access*, vol. 5, pp. 7123–7138, 2017.

[2] A. Papenfuss and M. Friedrich, "Head up only—a design concept to enable multiple remote tower operations," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, pp. 1–10.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *European Conference on Computer Vision*. Springer, 2014, pp. 756–771.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[15] H. M. Abdul, R.-D. Danijela, G. Axel, B. Milan, and S. Dušan, "Multi-disnet: machine learning-based object distance estimation from multiple cameras," in *International Conference on Computer Vision Systems*. Springer, 2019, pp. 457–469.

[16] 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.

[17] 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.

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

[19] L. Research, "X-Plane 11 Desktop Manual," http://www.x-plane.com/wp-content/uploads/2017/04/X-Plane-11.0-Desktop-Manual.pdf, 2017, [Online; accessed 2-Feb-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.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.