System for Tracking Aircraft Ground Movement Utilizing Artificial Intelligence

Abstract
Using a virtual ATC tower is one aspect of the airport digitization initiative, which aims to cut costs. Technology for tracking and detecting aircraft is required to guarantee the security of the deployment of virtual ATC towers. The invention of visual artificial intelligence that can recognize and follow aircraft ground movement in an airport is presented in this study. One feature that should be developed in this study is the capacity to issue a warning if two aircraft are in close proximity to one another. The techniques consist of a coordinate system conversion from pixels to meters for aircraft separation computation, the Deep SORT object tracking algorithm, and the YOLOv4 object recognition algorithm that has been trained using Image Dehazing Filter. Next, a fair condition recorded airport video is used to validate the model. With a mean average precision score of 95.92%, the trained YOLOv4 model was able to track every aircraft in the video, and with an error of 5.09%, the aircraft separation warning system functioned as expected. An airport’s possible use of an aircraft ground movement tracker was demonstrated by the constructed model.

Keywords: Aircraft, economic activities, RVT, ATC function, streamline

1 Introduction

Indonesia and many other airports across the world incurred losses. Approximately 75% of many airports in the world and Indonesia suffered losses. For example, Deputy Amir Airport in Pangkal Pinang has suffered loss since 2014. Losing in airport is unwanted, as it hinders the airport to fulfill its role as nodes in the transportation network, becoming a gateway for economic activities, becoming a transit center between modes of transportation, encouraging industrial activities, develop the surrounding area, and strengthen state sovereignty. It forces airport owners to cover their losses. Efforts that could be made to minimize the occurrence of losses are to streamline operating expenses. There are several components of operating cost that can be made more effective. Airport digitalization was one of the ways to make the cost more efficient. Integrating AI technologies into telecom and security systems enhances operational efficiency, minimizes losses, and ensures the seamless functioning of airports.

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).
By harnessing the power of AI, airports can optimize resource allocation, bolster security measures, and streamline communication networks. An example of airport digitalization is the implementation of the remote and virtual tower (RVT). By using RVT, airports could reduce the need for human resources so that the airport's operating cost could be reduced. In addition, the construction and maintenance costs of RVT are also cheaper than traditional ATC towers. Salen Trysil Airport in Sweden and London City Airport in the United Kingdom are among the airport that has already implemented this technology.

Currently, no airport in Indonesia has implemented this technology, as it is still in the planning stage. Furthermore, RVT is less expensive to build and maintain than conventional ATC towers. Among the airports that have already adopted this are Salen Trysil Airport in Sweden and London City Airport in the United Kingdom. Since it is still in the planning stages, no airport in Indonesia has yet used this technology. RVT must use object tracking technology to make sure the ATC function functions well even with less human resources. Artificial intelligence algorithms might recognize and monitor aircraft visible on runways, taxiways, and aprons based on video gathered from airport cameras. To further guarantee the traceability of every visible aircraft, further data might also be sent via the display panel. As a result, this technology may improve security and safety at airports. This essay made an effort to investigate the system's operation and mechanism in more detail. Artificial intelligence might help the air traffic controllers (ATC) monitor and manage aircraft movement on the airport ground.

The research's goals.

The following are the objectives of this research: to create a visual artificial intelligence model that can identify aircraft in an airport; to create a model that can track identified aircraft; and to create a model that can raise an alert if two aircraft are approaching each other.

Range and Restrictions

The construction of a visual artificial intelligence model that can identify and monitor airplanes in the airport and provide a warning if two aircraft are approaching one another will be the main goal of this article. The only available environmental conditions are throughout the day, when there is decent weather and adequate visibility. The video that serves as the model presentation was captured by a camera while it was stationary.

2 Methodology

2.1 Data Gathering

For this study, two different kinds of data must be collected: pictures and videos. The MP4 file format of the film depicts airplane movement within an airport. The entire model will be assessed using this footage as well. Three movies in all, taken from World Aviation 4K [7], were utilized to construct the visual artificial intelligence for this study. These videos are all accessible to the public. The first video displays aircraft movements at Hong Kong International Airport on runway RWY-07R, and the second video displays aircraft movements at the airport’s cargo apron. The third video captures aircraft movements at Concourse C at Harry Reid International Airport in Las Vegas, Nevada, as well as on the apron and taxiways around the airport. The model is trained using some photos, specifically for the purpose of identifying an airplane. Furthermore, a variety of picture angles are used to improve the model’s object detection performance.

A total of 1204 photos from two sources were utilized to train the model. Google Open Image Dataset (Google OID) is the initial source [8]. From this source, a total of 1000 photos are gathered. OIDv4 Toolkit, a Python-based application, is used to extract pictures.
The application concurrently downloads all of the picture data and the annotation files required for training the model. Furthermore, more photos in comparable settings to the video used in this study are needed to enhance the model's performance in identifying aircraft at the airport. As a result, 204 pictures depicting aircraft ground movement are extracted from the film.

2.2 Developing Identification Models

YOLOv4 is the framework that was utilized to create the detection model [9]. YOLOv4 functions as a one-stage detector that separates the picture into grids and makes item predictions for each grid. The YOLOv4 model has been taught to identify movements of airplanes at airports. To determine the ground truth of the picture data, the image dataset is first preprocessed and then annotated. The text format provides the object class, object width, object height, and top left x and y coordinates—the ground truth that the YOLOv4 model requires. Since Google has already supplied the information in the appropriate needed text file, images from Google OID do not require human annotation. It was only necessary to manually annotate 204 more photos showing aircraft activity. An application named labelImg, which is written in Python, is used for manual annotation. [10]

All of the collected photographs are subjected to the image dehazing filter. In order to make it easier for the built model to identify things in the image, this filter seeks to sharpen items in the image and eliminate noise. After filtering, the images will be divided into two groups: 83% of the images will be used to train the model, and the remaining 17% will be used to evaluate how well the built model performs.

To improve training speed and accuracy, the training model makes use of the pretrained YOLOv4.conv.137 model. The Google Colab coding environment is used for the training process. It takes about 3100 cycles to complete the training process. The model is subsequently trained in the Darknet environment in order to detect the item. Table 1 below shows the configuration used for the training procedure.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>64</td>
</tr>
<tr>
<td>Subdivision</td>
<td>16</td>
</tr>
<tr>
<td>Width</td>
<td>416</td>
</tr>
<tr>
<td>Height</td>
<td>416</td>
</tr>
<tr>
<td>Channel</td>
<td>3</td>
</tr>
<tr>
<td>Class</td>
<td>1</td>
</tr>
</tbody>
</table>

The loss from every iteration was computed by the trained model system. In this study, a loss of less than two is considered an acceptable loss value. In addition, the trained model system computes a number of metrics that may be used to assess the model's effectiveness. Namely, the measurements include mean average precision (mAP), recall, F1-score, true positive (TP), false positive (FP), and false negative (FN). To assess the performance of the model, a visual examination is also conducted. Every aircraft in every frame must be detectable by the trained model. In this study, the learned model is compared with the Bochkovskiy model, which was trained using the Microsoft COCO dataset. Assume that the trained model's performance is neither higher nor lower than that of the Bochkovskiy model. If so, it is necessary to carry out the entire training procedure, which includes re-gathering more data and retraining the model.

2.3 Using the Method of Monitoring

The operation of the tracking model is shown in Figure 1. The Deep SORT algorithm was...
the tracking algorithm. The tracking model that is being used is an expansion of Vittorio's concept. Google Colab is used to carry out the tracking procedure. Every video frame was processed by the application. When every object in the movie has been identified, the tracking procedure is started.

Fig. 1. Tracking Model Flowchart

To make the YOLOv4 model compatible with DeepSORT, it must be transformed from its original Darknet format to Tensorflow format. The airplane seen in every shot is recognized by the YOLOv4 model. The DeepSORT then compares the aircraft identified in a frame with the aircraft identified in earlier frames. Moreover, the identified item is defined as the same object if the system finds two airplanes from the two frames that are identical or highly similar to one another. The program further designates the identified item with the target ID and indicates that the aircraft has been tracked. Since DeepSORT needs the prior frame as a reference, tracking begins on the third frame. The aircraft’s target ID will be removed and it will be reported missing from tracking if it is not detected for three consecutive frames. The tracking procedure is continued until the frame expires. The object’s target ID, class, and other information are among the DeepSORT algorithm's outputs.

2.4 Airplane Segregation Detection Device and Coordination

The operation of the aircraft coordinate and separation warning system is shown in Figure 2 above. A pixel coordinate system is the coordinate system that was acquired from the DeepSORT algorithm. The real position of the aircraft with respect to a predetermined reference is obtained by converting the pixel coordinate system to the meter coordinate system. The equation connecting the two coordinate systems is necessary for the conversion. The equation is derived from a single frame that was extracted from the movie. The frame serves as the source of reference points.

Using OpenCV, the distance between these spots is found in pixels, and with Google Earth, the distance is found in meters. After that, the two variables are regressed to create an equation that links them. Moreover, this formula is applied to translate the aircraft’s original pixel-based coordinates into meters. Ultimately, the software output can show these coordinates.

Fig. 2. Flowchart for the Aircraft Coordinate and Separation Warning System
Additionally, a comparison is made between the coordinates of one tracked aircraft and those of the other aircraft. The two airplanes are regarded as being near together if their distance is less than 100 meters [20]. A red bounding box serves as a warning indicator for nearby planes. The boundary box is green rather than red for planes that are at a safe distance from one another. Ultimately, the program's accuracy is assessed by comparing the estimated and real distances.

3 Findings and Interpretations

3.1 Findings

The YOLOv4 detection model is then trained using a total of 1204 photos that have been processed using the Image Dehazing Filter. The procedure outlined in the preceding chapter is followed to train the YOLOv4 model. 3100 iterations of the model training procedure are carried out. Figure 3 below illustrates the model training graph, which displays the mAP and loss values in each iteration.

Fig. 3. Training Performance Model

Figure 3 shows that the loss produced by the model has converged to almost one after 3100 iterations. It is appropriate to accept the lose value. A maximum point is also reached by the mAP value during the model training procedure. Consequently, the model training procedure is finished. The assessment criteria shown in Table 2 below are thought to represent the best YOLOv4 model in this particular scenario.

To make the YOLOv4 model compatible with the Deep SORT algorithm, it is transformed from its former Darknet format to a TensorFlow format. Every frame in the movie and every airplane item in the frame are recognized by the YOLOv4 model. The Deep SORT algorithm operates the tracking system concurrently.
Table 2.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>564</td>
</tr>
<tr>
<td>FP</td>
<td>107</td>
</tr>
<tr>
<td>FN</td>
<td>35</td>
</tr>
<tr>
<td>Precision</td>
<td>0.84</td>
</tr>
<tr>
<td>Recall</td>
<td>0.94</td>
</tr>
<tr>
<td>F1-Score</td>
<td>0.98</td>
</tr>
<tr>
<td>mAP</td>
<td>95.92%</td>
</tr>
</tbody>
</table>

Figure 3 illustrates how, after 3100 iterations, the loss generated by the model converged to virtually one. It’s acceptable to acknowledge the value loss. During the model training process, the mAP value also reaches a maximum point. As a result, the model training process is complete. It is believed that the assessment criteria listed in Table 2 below best suit the YOLOv4 model in this specific situation.

The YOLOv4 model is converted from its original Darknet format to a TensorFlow format in order to make it compatible with the Deep SORT algorithm. The YOLOv4 model recognizes every frame in the movie and every piece of aircraft in the shot. The tracking system runs in parallel thanks to the Deep SORT algorithm. Changes to the coordinate system need an equation that links the pixels unit and meters. The regression method on distance data in pixels and distance data in meters yields the link between pixels and meters. The following are the two regression results for each demonstration video:

Equation 1 corresponds to the first video in the x direction,
Equation 2 to the first video in the y direction,
Equation 3 to the second video in the x direction,
Equation 4 to the second video in the y direction,
Equation 5 to the third video in the x direction,
Equation 6 to the third video in the y direction.

\[
\text{meters} = 0.272 \times \text{pixels} - 8.16
\]

\[
\text{meters} = -1.616 \times \text{pixels} + 924.57
\]

\[
\text{meters} = 0.3041 \times \text{pixels} - 9.3062
\]

\[
\text{meters} = -1.6626 \times \text{pixels} + 1114.6
\]

\[
\text{meters} = 0.16 \times \text{pixels} - 4.8
\]

\[
\text{meters} = 0.0088 \times \text{pixels}^2 - 14.69 \times \text{pixels} + 5956.1
\]

The YOLOv4 model, Deep SORT, and the pixel and meter data regression equations are used simultaneously to produce a program that can track the aircraft’s ground movements at the airport. Along with that, the developed model also provides the aircraft coordinates and a warning system if there are two adjacent aircraft. The program is run for the three demonstration videos and produces a new video as the program’s output.

Fig. 4. Outcomes for the Initial Video (a) at 3 seconds and (b) at 5 seconds.
Fig. 5. Outcomes for Video 2: (a) in the second and (b) in the fifth seconds

Fig. 6. Outcomes for Video 3: (a) in the second and (b) in the fifth seconds

3.2 Result

Two approaches are used to analyze the performance of the YOLOv4 model: visual assessment and evaluation metrics. The first approach is implemented initially. Table 2 displays the assessment metrics for the YOLOv4 model. The YOLOv4 model has a TP value that is higher than the FP and FN values, as the table demonstrates. As a result, the resulting measures were also quite valuable. It suggests that the YOLOv4 model is precise and only had a small number of incorrect predictions. Additionally, there is a correlation with a comparatively high mAP of 95.92%. Because of this, the YOLOv4 model was deemed “good” using these assessment measures.

Visual evaluation is the second technique utilized to assess the YOLOv4’s performance. Ideally, every aircraft seen in the picture should be detectable by the detection model. The YOLOv4 model properly identifies each aircraft in the demonstration movies through visual examination. The YOLOv4 model exhibited a high degree of confidence as well; Figure 7(a) shows that each aircraft’s confidence rating falls between 98 and 100%.

Fig. 7. Yolov4 Comparison between Bochkovskiy’s Model with (a) Our Model
As a comparison to our model, a YOLOv4 model trained using Bochkovskiy's technique is utilized. Since Bochkovskiy's model was trained on a distinct dataset, it is not possible to compare the evaluation metrics of the two models. One way to make comparisons was through visual assessment. The first demonstration video shows the Bochkovskiy's YOLOv4 model in action, as seen in Figure 7(b) above. The picture demonstrates that only three of the six airplanes in the picture could be identified using Bochkovskiy's YOLOv4 model. When an aircraft's longitudinal axis is parallel to the camera, Bochkovskiy's YOLOv4 model has trouble identifying it. Compared to the authors' YOLOv4 model, this is different. In the first demonstration film, the authors' YOLOv4 model recognizes the entire aircraft correctly. Consequently, it can be said that for the particular scenario, the YOLOv4 model trained in this study performs better than the YOLOv4 model trained by Bochkovskiy.

The aircraft detected by the YOLOv4 model can be located via Deep SORT. Every aircraft that is identified by Deep SORT is given a target ID. Figure 8, which displays the Deep SORT tracking performance for the first example film, illustrates this procedure. A target ID is given to each aircraft that is found, and this ID is retained throughout the film. In every frame of the initial demonstration video, Deep SORT was able to identify, label, and forecast the direction of the aircraft's motion with success. Without Deep SORT, YOLOv4 models would not be able to provide each observed aircraft a unique identity. In Figure 8, the Deep SORT effect is plainly visible. Additionally, every aircraft shown in the second and third example movies is tracked by Deep SORT.

When it came to identity shifts and fragmentation issues, Deep SORT fared well. Identity swaps only happen once in the third demonstration video, despite the fact that they don't happen in the first or second. In Figure 4, this is illustrated. The target ID on aircraft number 4 switches to number 7 based on the two pictures. According to Deep SORT, airplane numbers 4 and 7 should be treated same even though they are distinct aircraft. The third example video also shows instances of fragmentation. On the other hand, Deep SORT does a good job of tracking and is rarely prone to errors.

The equations used in the three demonstration films differ. This is due to the fact that every video has a unique set of shooting parameters, including the object's distance from the camera, the zoom level of the camera, the angle at which the camera is positioned, etc. Every coordinate axis has unique properties as well. A linear function may be used to depict the connection between pixels and meters since the camera records pictures that are of the same size along the x-axis. On the y-axis, it stands in contrast to the pixel-meter connection. The camera and the ground are at an angle. When something is far away, it will...
There are three distinct equations in the demonstration videos. The reason behind this is that every video has a unique set of shooting parameters, including the camera’s zoom level, distance from the subject, perspective, and so on. The attributes of each coordinate axis vary as well. Since the camera records identically sized pictures along the x-axis, a linear function may be used to depict the connection between pixels and meters. It is in opposition to the y-axis pixel-meter connection. The camera is angled with respect to the ground.

Things that are far away will look smaller than identical objects that are closer to the camera. Due to non-perfectly static conditions when taking the video, the coordinates in the footage occasionally move. These variations are, nonetheless, negligible, and the coordinate system may be said to be functional. The coordinates of the two airplanes indicate their distance from one another. By comparing the estimated and real distances between the airplanes, one may ascertain the correctness of the coordinate system. The results of the computation are shown in Table 3 below. The separation warning system’s average mistake rate is 5.09%. As a result, the software is deemed accurate.

When another aircraft is approaching, the software effectively issues alerts. The airplane next to the red border box is marked by the software. Every example video demonstrates it. In the first demonstration video, aircraft numbers 1 and 4 are seen as being near. In the second demonstration video, aircraft 1, 2, and 3 are regarded as near. In the third demonstration video, airplane number 7 is originally not shown as being close to aircraft number 9. Nevertheless, the bounding box of the two airplanes becomes red as they approach each other.

<table>
<thead>
<tr>
<th>Video</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Video</td>
<td>5.18%</td>
</tr>
<tr>
<td>Second Video</td>
<td>4.24%</td>
</tr>
<tr>
<td>Third Video</td>
<td>5.86%</td>
</tr>
<tr>
<td>Average Error</td>
<td>5.09%</td>
</tr>
</tbody>
</table>

4 Conclusion

Three findings may be drawn from this study. Here are the three findings that are presented:

• Using YOLOv4, an airplane detection model at the airport has been successfully created. The Image Dehazing Filter is a prior preprocessing step performed on the pictures used to train the YOLOv4 model. Consequently, the YOLOv4 model successfully identifies all visible aircraft and has a mAP score of 95.92%.

• The YOLOv4 and Deep SORT algorithms have been successfully combined to create the airport’s aircraft monitoring model. All identified aircraft are successfully tracked by the artificial tracking model, and

• In the event that two airplanes are in close proximity to one another, the created software might further provide a warning. Regression analysis is used to build this feature based on data gathered from the video. Furthermore, the feature has the ability to provide the aircraft’s distance from one another in meters, with an error rate of less than 5.09%.

References

1 M. Cervinka 2019 Is a Regional Airport Business a Way to Make Profit? Transportation E3S Web of Conferences 477, 00009 (2024) https://doi.org/10.1051/e3sconf/202447700009