Vehicle Tracking System in Drone Imagery with YOLOv5 and Histogram

Jehwan Choi†, Seongbo Ha, Youlkyeong Lee and Kanghyun Jo

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, Korea
(Tel: +82-52-259-1664; E-mail: jhchoi@islab.ulsan.ac.kr)
(Tel: +82-52-259-1664; E-mail: sbha@islab.ulsan.ac.kr)
(Tel: +82-52-259-1664; E-mail: yklee@islab.ulsan.ac.kr)
(Tel: +82-52-259-2208; E-mail: acejo@ulsan.ac.kr)

Abstract: In this study, we propose a vehicle tracking system targeting drone footage. The proposed system utilizes the real-time object detection network, YOLOv5, to acquire vehicle location information and segment the vehicle regions based on it. The system analyzes the histogram of the segmented regions, compares them with past frames, and determines whether the objects are identical to perform tracking. To enhance the efficiency of histogram comparison, the algorithm is designed to compare objects only within a certain radius using coordinate information and past frame object data. The MOTA, a representative tracking evaluation metric, showed 90%. However, it is important to consider the limited environment of data usage and experiments. The results of this study suggest that the real-time performance of the vehicle tracking system can be utilized in various fields such as traffic control, vehicle management, and accident response.

Keywords: Vehicle tracking system, histogram-based similarity, traffic analysis with drone footage.

1. INTRODUCTION

The Vehicle Tracking System (VTS) is a technology that identifies and tracks the location of vehicles in real-time, and is used in areas such as Intelligent Transportation Systems (ITS) and autonomous vehicles. Up until now, vehicle tracking has primarily been done using the Global Positioning System (GPS). However, with the recent advancement of deep learning technologies, research on camera-based VTS technology is being actively conducted [1–4]. In addition, as interest in drones has increased due to their lesser susceptibility to weather conditions and terrain, as well as their fast movement, which makes them more effective for real-time analysis systems, many researchers have introduced computer vision technology to videos captured by drones to conduct VTS research. Examples include research on object tracking under extreme conditions [5], which uses visual information such as the appearance or shape of the target, along with motion-related information like the target's speed and direction, and studies on real-time traffic monitoring systems using OpenCV-based UAVs [6], which conduct experiments and evaluate them in comparison with high-precision GPS benchmarks. There are also studies on UAV detection and tracking benchmarking research [7] utilizing the latest deep learning networks [8–11], and papers proposing effective multi-object tracking algorithms by introducing an ID update module to address issues such as irregular camera movements and visual changes [12]. In this paper, we also conduct research on a VTS algorithm utilizing the real-time object detection network YOLOv5 [13].

According to [14], object tracking algorithms are classified into three main parts: object detection, object classification, and object tracking, which are carried out sequentially, with methods for each stage explained. The core of the tracking algorithm is described as finding an approximation of the object's path in a moving scene. In other words, by comparing how similar the path of the moving object is to the previous frame, the algorithm can recognize it as the same object and track it. In this paper, using YOLOv5, we predict the location, class, and accuracy of vehicles (bike, motorcycle, car, bus, truck) in each frame, then set the region of interest (ROI) based on these

(a) Urban traffic congestion road scene

(b) classifying vehicles with similar colors and features

Fig. 1: The challenge of object tracking in drone videos

† Jehwan Choi is the presenter of this paper.

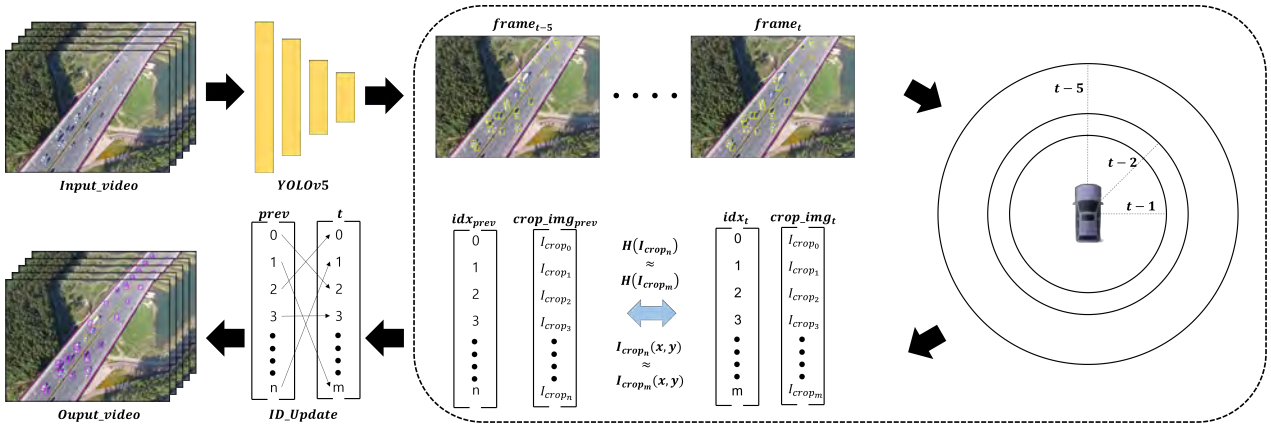

Fig. 2: Overall framework of VTS using detection results and histogram of cropped images

results. We then determine the same object through histogram calculations. Histograms are important tools for analyzing features such as color, brightness, and texture in images. They represent the frequency of these features in an image in the form of a bar graph. By quantifying and comparing image characteristics through histograms, the accuracy of identifying the same object can be improved. Calculating and tracking an object's coordinates is a commonly used method. Since most image processing techniques convert videos into several frames per second for computation, the difference in object movement between consecutive frames is insignificant. However, due to the unique characteristics of drone footage, histogram calculations are necessary within the ROI. As shown in Fig. 1, unlike footage taken from the ground, drones can capture and analyze a wide area. As a result, more objects are included, and in particular, there is a high likelihood of vehicles being densely distributed in congested urban areas. Moreover, with a large number of vehicles with similar colors present, it is crucial to properly utilize the detected vehicle's coordinates and image features.

In this paper, we propose an algorithm that utilizes the coordinates of detected objects and histogram calculations to effectively track objects detected in drone footage. We use the real-time object detection model YOLOv5 to obtain the object's coordinates and utilize the center point of the acquired coordinates (top-left, bottom-right) as the representative position of the object. All detected objects search for existing objects within a certain range in the previous frame based on their representative position and determine whether they are the same object through histogram comparison. By changing the index of the objects identified as the same object to the current object's label, a continuous VTS implementation is possible. The main contributions of this article are summarized as follows:

- We propose an efficient vehicle tracking algorithm using coordinates of objects and histograms with drone imagery.
- We apply histogram operations only to objects that exist within a certain range based on their coordinates, in order to increase the accuracy and reduce the time for identifying the same object in consecutive frames.

2. PROPOSED WORK

2.1. Overall Framework

Object tracking is mainly used in videos. In this paper, the VTS network also takes video as input. The overall framework is illustrated in Fig. 2. The input video is connected to the object detection network, YOLOv5. When a video is input, the YOLOv5 network splits it into 30 FPS for processing. The object detection network outputs a vector $\vec{V}$=(x_1, y_1, x_2, y_2, $confidence$, $category$) representing the detected object's position, accuracy, and category. If 10 objects are detected in a single frame, 10 vectors are generated. However, since the order of object prediction and the number of detected objects are not constant, the results are compared between consecutive frames. In this paper, we maintain the detection results of the most recent 5 frames. To minimize calculations, we consider the vehicle's speed and only add vehicles within a certain range in the recent 5 frames to the histogram calculation list. As the probability of a wider movement range increases for older frames, we gradually expand the search range. Vehicles captured within the range are determined to be the same object or not through HSV histogram calculations. Once the histogram and coordinate comparison process is finished, ID updates occur. ID updates are applied to all objects detected in each frame, and when the input image displays the object detection results and ID, the final result video is completed.

2.2. Object Matching

The process of verifying whether objects detected in consecutive frames are the same object can be seen in the content within the black dashed box in Fig. 2. In the object list for the most recent 5 frames, not only the detection results are stored, but also the continuously updated ID, histogram, and location information. Once the centroid calculation for all detected objects in the current frame is completed, the candidate search is performed in the most recent 5 frames for histogram calculation. The detailed process of candidate search is shown in Fig 3.

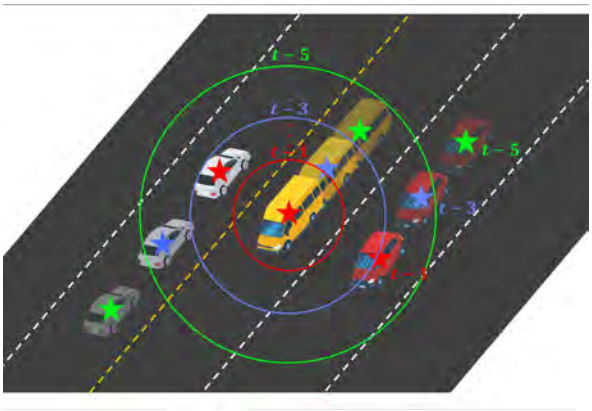

Fig. 3: Schematic Image of Fixed Radius Exploration Process for Utilizing Past Frames in Histogram Comparison Candidate Search. Red color represents $frame_{t-1}$, blue color represents $frame_{t-3}$, yellow color represents $frame_{t-5}$.

2.2.1. Histogram Comparison Target Extraction

Object tracking requires identifying the same object in consecutive frames and maintaining its assigned ID. It is essential to determine which object in the current frame was located where in the previous frame, and what value it had as an ID. As mentioned earlier, drone footage contains dozens of vehicles, and many vehicles with similar characteristics exist, making it more efficient to extract and compare candidate vehicles with a high probability of being the same object. The candidate extraction process consists of coordinate-based search followed by histogram calculation. Fig. 3 shows the extraction of histogram comparison candidates only within a certain range after coordinate-based search.

The search is conducted within a certain radius from the centroid of the current vehicle, which is the target for ID update, starting from $frame_{t-1}$ to $frame_{t-5}$. Through various experiments at different altitudes and angles, it has been observed that the single-object movement speed per frame is around 100 pixels. Therefore, as we go back in time, we increase the radius by 100 pixels for each past frame and add the vehicles within that range to the comparison target list.

2.2.2. Histogram Similarity

In this paper, we use histograms, a fundamental concept in computer vision, for calculating the similarity between objects. As previously mentioned, histograms use color-related features of an image as the main parameters and compute the frequency of these color features as the result. The color spaces commonly used for calculating image histograms are RGB(Red, Green, Blue) and HSV(Hue, Saturation, Value). In this paper, we use the HSV histogram. The HSV color space is suitable for object tracking in drone footage, where the field of view(FOV) can change drastically, as it processes color information separately from saturation and brightness. This allows the color information to remain consistent even as the object gets closer or farther away.

The histogram similarity is calculated for each vehicle in the comparison list obtained in Section 2.2.1 and the vehicle targeted for ID update. After calculating the similarity between the target vehicle's histogram and the histograms of the vehicles in the list, the past ID value with the largest similarity is replaced with the current vehicle's index. The four representative histogram similarity measurement methods are Correlation, Chi-Square, Intersection, and Bhattacharyya distance. In this paper, we use the correlation method due to its robustness to noise. It is suitable for measuring linear relationships of continuous features such as color and texture. The formula for calculating the correlation histogram is defined as:

$$\text{Hist}_{\text{corr}} = \frac{\sum((H_1(i) - \overline{H_1}) * (H_2(i) - \overline{H_2}))}{\sqrt{\sum(H_1(i) - \overline{H_1})^2 * \sum(H_2(i) - \overline{H_2})^2}} \quad (1)$$

Once the histogram calculation for all detected vehicles in the current frame is completed, the ID update is performed using the index of the object with the maximum value. The schematic image of the histogram similarity result comparison is shown in Fig. 4.

2.2.3. Final Algorithm for VTS

Algorithm 1 Histogram Comparison Target Extraction

Require: A drone video sequence
Ensure: The matched objects V_m
Comparison_list = []
for idx_m, $hist_m$, $coordinate_m$ in $frame_t$ **do**
  idx_m = -1
  Get the C_m
  for each frame in $frame\_previous$ **do**
    if $distance(C_n,C_m)$<200 at $frame_{t-1}$ **then**
      Comparison_list.append(V_m)
    else if $distance(C_n,C_m)$<400 at $frame_{t-2}$ **then**
      Comparison_list.append(V_m)
    else if $distance(C_n,C_m)$<600 at $frame_{t-3}$ **then**
      Comparison_list.append(V_m)
    else if $distance(C_n,C_m)$<800 at $frame_{t-4}$ **then**
      Comparison_list.append(V_m)
    else if $distance(C_n,C_m)$<1,000 at $frame_{t-5}$ **then**
      Comparison_list.append(V_m)
    end if
  end for
  Highest_result = 0
  for each vehicle in Comparison_list **do**
    result = Calculation_histogram()
    if result > Highest_result and result > 0.85 **then**
      index_match = index_current
    else if index_match == -1 **then**
      index_match = New_ID()
    end if
  end for
end for

In Section 2.2.1 and 2.2.2, most of the vehicles in each frame are matched after going through the processes. However, due to the characteristics of deep learning models, it is possible that an object may not be detected, or a new vehicle may appear, making it impossible to match with existing vehicles. In these cases, a new ID must be created instead of updating the existing ID. New ID creation conditions include cases where detection has occurred, but the centroid distance is far or the similarity

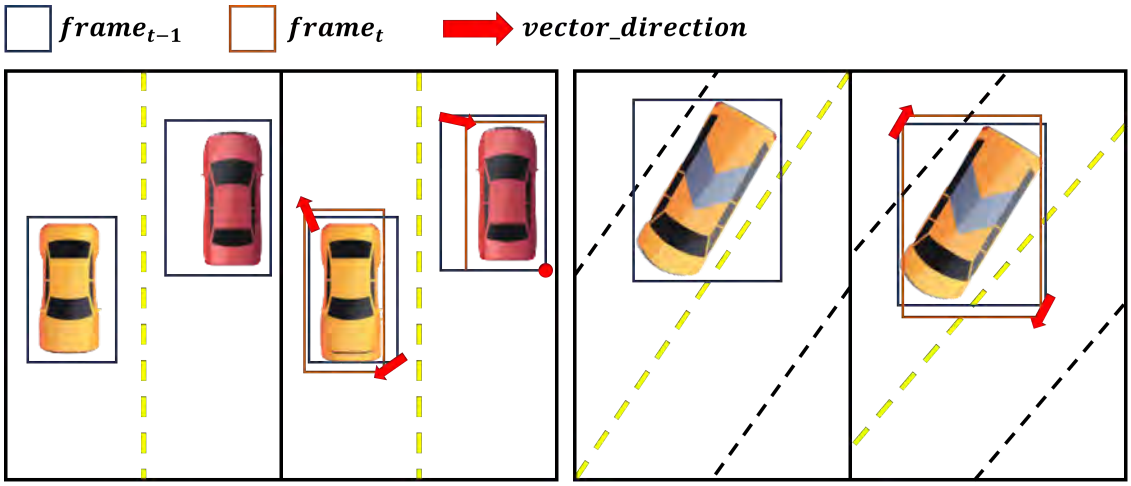

Fig. 4: Two representative examples of reduced accuracy when using Cosine similarity. Inconsistency in the creation of top-left and bottom-right coordinate vectors due to irregular bounding box generation for each frame (left), and the process where the vector direction of each vertex of the bounding box is reversed due to a rapid angle change of the drone (right).

does not exceed the threshold value. To distinguish between these cases, initialize all detection result indices in the current frame to -1. Since the ID is updated when matched with an object from a past frame, the current index is not important. If both the centroid distance calculation and histogram similarity are not satisfied, the index of the object remains -1. If this state persists until the end of the comparison calculation process, the object is considered as a new vehicle, and an ID is assigned by adding 1 to the largest existing ID value. The overall VTS algorithm can be summarized as Algorithm 1,where $frame_t$ is current frame, V_m is the matched vehicles including ID_m, $hist_m$, and $coordinate_m$, C is the center point of $coordinate$, m is the number of vehicles in $frame_t$, n is the number of vehicles in $frame_{previous}$ and $frame_{previous}$ is a set of $frame_{t-1}$ to $frame_{t-5}$.

3. EXPERIMENT

3.1. Dataset

The data used for the experiment are videos from the autonomous flight drone dataset [15] built by the University of Ulsan in 2020. A total of four videos were used for the experiment, and information about altitude and angle can be found in Table 1.

Table 1: The information of drone data.

Region	Altitude(m)	Angle(°)	Time(s)
Ulsan_Samhogyo	90	60	120
	50	50	
Ulsan_Taehwagyo	60	45	
	40	30	

3.2. Ablation Study

In this paper, we used the midpoint of the top-left and bottom-right coordinates of the bounding box(bbox) to determine the center of the vehicle for identifying the same object. The directionality of the detected object can also be a parameter for determining the same object. Therefore, we measured the similarity of the vectors of the two coordinates themselves instead of the midpoint of the coordinates. However, the accuracy using cosine similarity significantly decreased. There are two reasons for this and Fig. 4. is helpful for your understanding.

Firstly, the inconsistent size and coordinates of the bbox. Since object detection is based on a deep learning model, the size of the bbox generated in each frame is not constant, and even if the size is the same, it does not always fit the object precisely. This results in inconsistent vector direction and magnitude when converted to vectors.

Secondly, the motion characteristics of the drone. Drones are airborne vehicles, so their degrees of freedom are higher than those of ground-based objects. They move not only forward and backward but also up, down, left, and right. Therefore, it is difficult to maintain consistent directionality when calculating vectors. Consequently, it is uncertain to determine the direction of movement using the similarity of the movement vectors of the bbox coordinates when tracking objects in drone videos using deep learning models.

3.3. Evaluation Metric

To evaluate the performance of a VTS, we utilize evaluation metrics such as multiple object tracking accuracy(MOTA), false negatives(FN), false positives(FP), and ID switches(IDs). MOTA is a comprehensive metric for evaluating object tracking performance, and its formula is as follows:

$$\text{MOTA} = 1 - \frac{\text{FP} + \text{FN} + \text{IDs}}{\text{GT}} \tag{2}$$

FN refers to cases where the object actually exists but the system fails to detect it, while FP refers to cases where the object does not actually exist but is incorrectly detected as existing. IDs refer to cases where the same ID is assigned to different objects or different IDs are assigned to the same object.

Fig. 5: The result image of Ulsan_Samhogyo area'a 90m 60°

Fig. 6: The result image of Ulsan_Taehwagyo area'a 60m 45°

Fig. 7: The result image of Ulsan_Samhogyo area'a 50m 50°

4. RESULT

In this study, the proposed VTS was tested using an autonomous drone dataset that does not have ground truth(GT) for tracking. The results of applying the proposed VTS to real drone data are illustrated in Fig. 5, 6, and 7. To calculate the accuracy, we extracted 5 seconds of result images from the Ulsan_Samhogyo area's 50m 50° footage and applied the evaluation metrics. The accuracy was measured to be higher than other state-of-the-art models, however, it is essential to consider that the learning and object detection, as well as tracking, were applied in a limited situation using this dataset. The selection of the radius for extracting histogram comparison candidates based on the drone data's various altitudes and angles significantly impacts VTS accuracy. The evaluation metrics for each case are shown in Table 2.

Table 2: The results of proposed VTS

GT	MOTA	FP	FN	IDs
4,306	90%	1	401	28

5. CONCLUSION

In this paper, we conducted research on methods to detect and track vehicles in drone imagery. For vehicle detection, we used the YOLOv5, a one-stage detection algorithm. The YOLOv5 network provides the top-left and bottom-right bbox coordinates of detected vehicles. We determine the center of the vehicle by taking the midpoint of the two coordinates, and search for vehicles within a certain range from the center in the past five frames. The vehicles found are then compared with the vehicles

detected in the current frame using the HSV histogram similarity to determine whether they are the same object in consecutive frames. After confirming the identity of the same object, the vehicle tracking system is completed through an ID updating process. Although there is no ground truth for tracking in the drone dataset used in the experiments, which may result in somewhat insufficient accuracy calculations, it can be confirmed from Figures 5, 6, and 7 that tracking is working well for detected objects. The remaining challenges include establishing a standard for the range of vehicle movement between frames, as the altitude and angle differ for each drone video. In addition, since the accuracy of the deep learning model's object detection is directly related to the performance of the tracking algorithm, improving the performance of the object detection model is also future work.

6. ACKNOWLEDGEMENT

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

REFERENCES

[1] F. Li, Z. Wang, D. Nie, S. Zhang, X. Jiang, X. Zhao, and P. Hu, "Multi-camera vehicle tracking system for ai city challenge 2022," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 3264–3272.

[2] P. Lyu, M. Wei, and Y. Wu, "Multi-vehicle tracking based on monocular camera in driver view," *Applied Sciences*, vol. 12, no. 23, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/23/12244

[3] M. Wu, Y. Qian, C. Wang, and M. Yang, "A multi-camera vehicle tracking system based on city-scale vehicle re-id and spatial-temporal information," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 4072–4081.

[4] M. Shehata, R. Abo-Alez, F. Zaghlool, and M. Abou-Kreisha, "Deep learning based vehicle tracking in traffic management," *International Journal of Computer Trends Technology*, vol. 67, pp. 5–8, 03 2019.

[5] "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," vol. 31.

[6] G. Guido, V. Gallelli, D. Rogano, and A. Vitale, "Evaluating the accuracy of vehicle tracking data obtained from unmanned aerial vehicles," *International Journal of Transportation Science and Technology*, vol. 5, no. 3, pp. 136–151, 2016, unmanned Aerial Vehicles and Remote Sensing.

[7] B. K. S. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon, and H. P. H. Shum, "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE, oct 2021. [Online]. Available: https://doi.org/10.1109\%2Ficcvw54120.2021.00142

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. [Online]. Available: https://doi.org/10.1007\%2F978-3-319-46448-0_2

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[12] S. Liu, X. Li, H. Lu, and Y. He, "Multi-object tracking meets moving uav," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8866–8875.

[13] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Yifu), C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[14] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, pp. 2970–2978, 2014.

[15] K. Jo. (2020) Autonomous drone dataset. [Online]. Available: https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100