An Edge-Controlled Outdoor Autonomous UAV for Colorwise Safety Helmet Detection and Counting of Workers in Construction Sites

Susanta Sharma, Allumallu Veera Venkata Susmitha, Lan-Da Van and Yu-Chee Tseng

Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

Email: sharma.susanta@gmail.com, allususmitha@gmail.com, ldvan@cs.nctu.edu.tw, yctseng@cs.nctu.edu.tw

Abstract— **In this paper, an edge-computed and controlled outdoor autonomous UAV system is proposed to monitor the safety helmet wearing of workers in construction sites. Detection and counting of the workers with safety helmets of specified colors and those without safety helmets is the main focus of this work. Five standard safety helmet colors including blue, orange, red, white, and yellow are considered. The novelties of the work are 1) the design of a modularized software architecture running on an Android smartphone as an edge device for outdoor autonomous UAV navigation, 2) the implementation of real-time colorwise detection and counting of workers with and without safety helmets from UAV's first-person view (FPV), 3) the implementation of a simple upper-side cropping and hue, saturation, value (HSV) filtering method for color decision. The resulting average safety helmet detection accuracy for 10 different cases is 81.02%.**

Keywords— Autonomous flying, computer vision, deep learning, edge computing, UAV (Unmanned Arial Vehicle)

I. Introduction

In construction sites, there exist many potential hazards and accidents if workers do not closely follow the regulations. It can be seen from [1] that brain injury is one of the major accidents. Therefore, it is mandatory for workers to wear safety helmets (also called hardhats) [2] in construction sites. On the other hand, administrative staffs would prefer to use lightweight devices to monitor workers' status. Thus, it is our motivation to develop a deep learning-assisted model running on an edge device like a smartphone to enable UAV navigation and inspection. The challenges are as follows: 1) How to merge the software modules to perform worker detection, helmet color detection, GPS waypoint management, and UAV flight control instruction generation from SDK, together into one edge device to achieve real-time autonomous UAV navigation and inspection? 2) How to count the colorwise safety helmets of workers in a given construction site?

A lot of previous work [3-12] has addressed construction site safety. Rubaiyat *et al.* [3] employed computer vision techniques for automatic detection of safety helmets for construction safety. The proposed method includes two recognition steps: histogram of oriented gradient (HOG) and combination of color based and circle Hough transform (CHT) for feature extraction. Park *et al.* [4] proposed a detection and tracking based hybrid method via a fixed camera to continuously track workers and solve occlusion problems. Yang *et al.* [5] proposed using gait patterns of construction workers to detect falls. Fang *et al.* [6] proposed a faster R-CNN based method to detect non-hardhat users from far-field surveillance cameras. Kai *et al.* [7] proposed a method to determine safety helmet wearing by TensorFlow models to detect pedestrians and human head-to-body ratio calculation of a full human body view. The safety helmet colors are detected using a HSV filter. Kang *et al.* [8] developed a small camera along with a GPS tracker for safety helmets to collect geotagged images. Wu *et al.* [9] employed reverse progressive attention (RPA) to extract new features to detect hardhat colors of workers. Mneymneh *et al.* [10] proposed vision-based motion detection based on standard deviation matrix (SDM), vision-based human detection and vision-based hardhat detection based on cascade classifier and color classification to verify the colors of the hardhats wearing. Asadzadeh *et al.* [11] reviewed and discussed the integration of sensor-based systems and building information modeling (BIM) for safety management in construction sites. Zhang *et al.* [12] reviewed the vision-based technologies for occupational health and safety monitoring of workers in construction sites.

The other detection and counting researches [13-16] are summarized below. Lee *et al.* [13] applied NVIDIA Tegra TX1 to speed up object detection and counting. When an object is counted repeatedly, the time difference is calculated to eliminate repetitions. Saxena *et al.* [14] applied a deep learning algorithm and proposed counting algorithms to detect and count moving vehicles. Yang *et al.* [15] proposed a vision-based mobile people counting system in a given area using a smartphone, where the non-maxima suppression (NMS) algorithm is used to reduce redundant bounding boxes. Farjon *et al.* [16] proposed real-time flower detection and counting using faster R-CNN for apple trees. Vehicle counting at road intersections via time-spatial images and dangerous driving behavior detection via road-side cameras are discussed in [17, 18].

However, the above works did not focus on edge-guided autonomous flying for color counting of safety helmets of workers under different illumination levels. This motivates us to develop an edge-controlled outdoor autonomous UAV for colorwise safety helmet detection and counting of workers in construction sites. The novelties of the work are 1) the design of a modularized software architecture running on an Android smartphone as an edge for outdoor autonomous UAV navigation, 2) the implementation of real-time colorwise safety helmet and without-safety-helmet workers detection and counting from UAV's FPV, and 3) the implementation of a simple upper-side cropping and HSV filtering method for color decision.

II. System Architecture and Method

To design a real-time construction site monitoring system, we propose a system architecture as shown in Fig. 1. The off-the-shelf UAV (DJI Matrice 210 V2) is used to capture on-site videos. In this process, GPS waypoints are set

978-1-6654-1368-8/21/$31.00 ©2021 IEEE

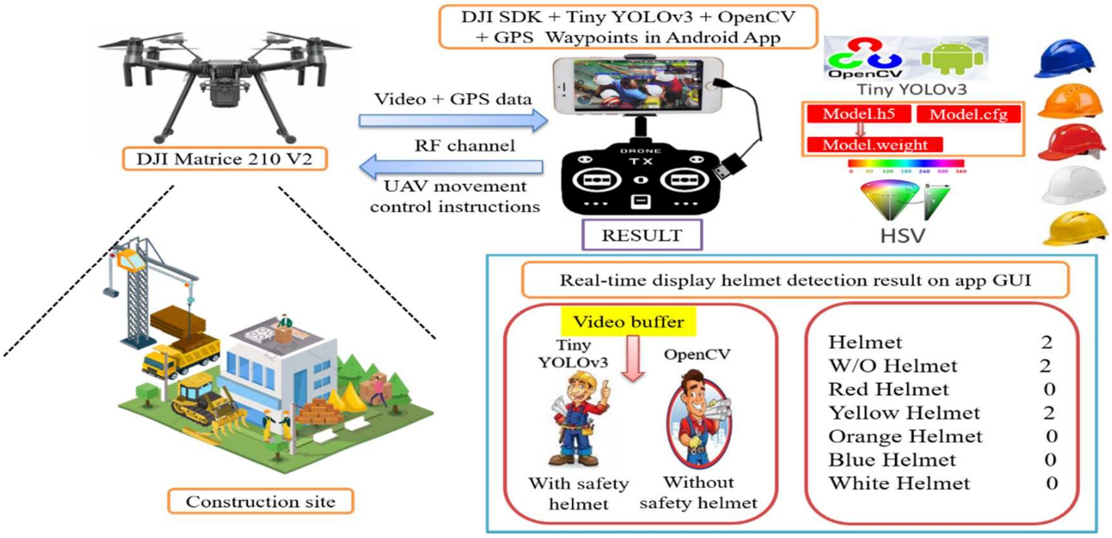

Fig 1. Proposed outdoor autonomous flying control and computation system architecture.

according to the construction site of interest. Through RF channel, the captured video is sent to the edge (smartphone)-oriented autonomous outdoor flight control system to perform real-time detection and counting. In the proposed outdoor flight system, a software system with DJI SDK is integrated in an Android smartphone for video streams processing as well as GPS waypoint data processing of flying mission, tiny YOLOv3 for safety helmet detection, and a HSV filter for safety helmet color analysis.

A. Software System Integration on Android Studio

The architecture of the software system and the workflow of our app are shown in Fig. 2 and Fig. 3, respectively. It is a practical challenge to integrate DJI SDK, GPS waypoint data, tiny YOLOv3, OpenCV into an Android smartphone to develop a fully functional app. The integration is challenging due to its parallel processing, computation management, real-time drone control as well as display the data on the APP GUI. DJI SDK is used to access the drone camera view and issue movement control instructions. Tiny YOLOv3 model is used for the object detection, and HSV filter is used to analyze the safety helmets of workers.

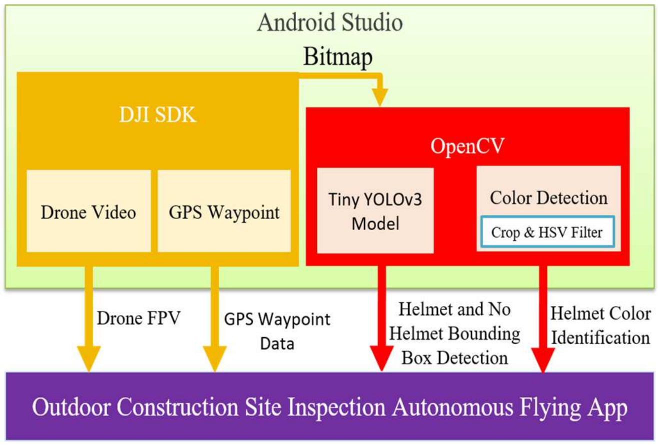

Fig 2. Software system structure.

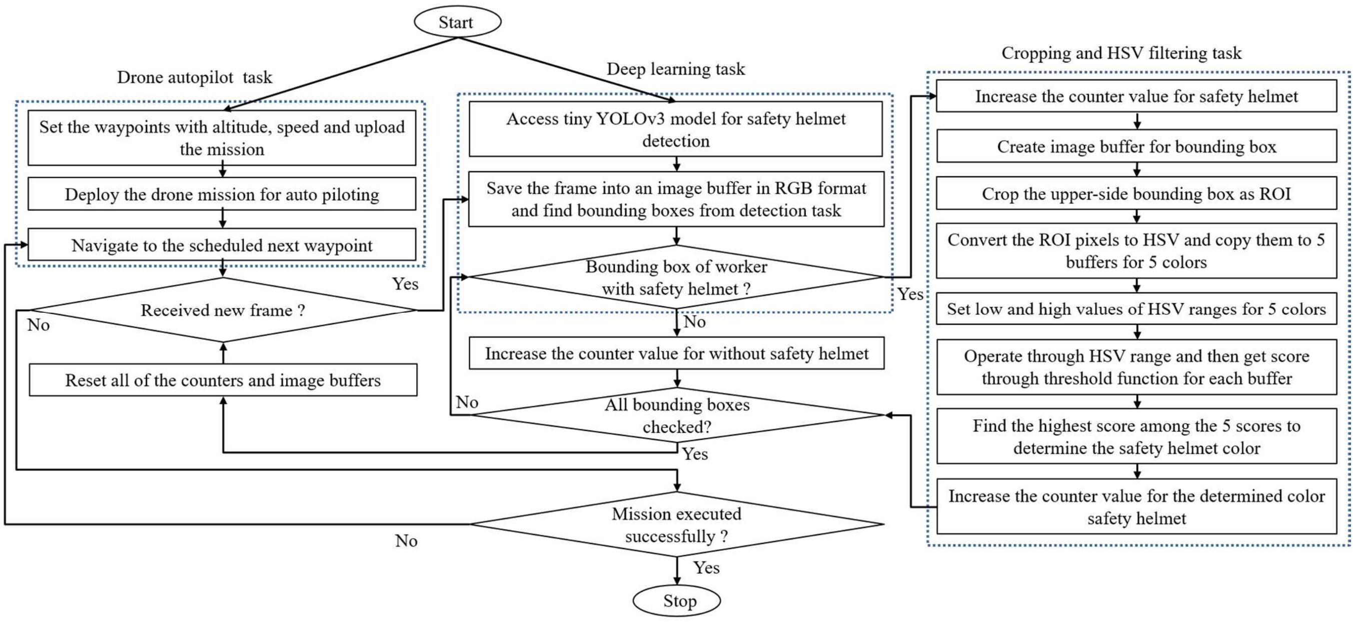

Fig 3. System workflow.

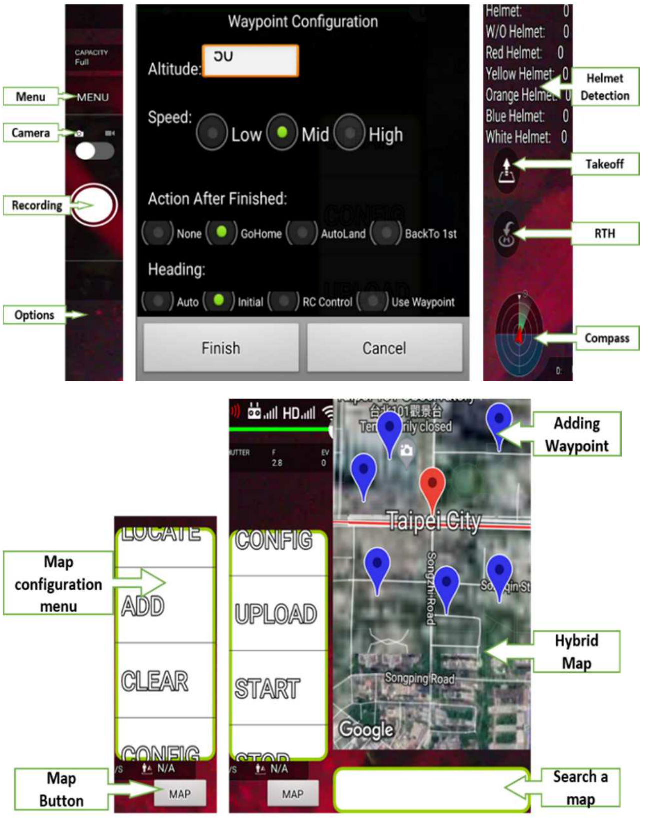

Fig 4. The UI for GPS waypoint of autonomous flying app.

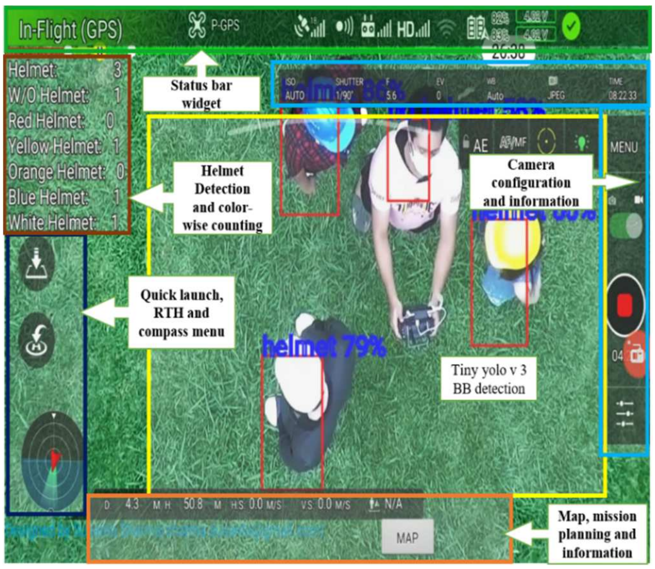

Fig. 5. The detection result of the app in an outdoor environment (UAV on GPS waypoint flight).

DJI provides the corresponding SDK, which includes some "Android widgets" and services. As shown in Fig. 4 and Fig. 5, our app UI provides multifunctional dropdown items and widgets to switch between FPV and Zenmuse camera views. We provide a "fragment" map view to customize the waypoint flight mission. The buttons in the view are used for waypoint selection, altitude setting, post-mission UAV path setting, and mission emergency abortion.

For better visualization experience, the real-time flight path, map and waypoints can be shown or hidden with a single soft button click while working in the background. The map view is a hybrid Google map view. This allows users to see the buildings and other constructions on the waypoint's flight path. Map can be accessed via the Internet by the mobile phone or the current location of the UAV's GPS. The waypoints, which consist of specific latitude and longitude, guide the UAV while flying. The UAV control option is synchronized with the UAV GPS and waypoints. Since safety is a major concern, there are one-touch widgets for return to home (RTH) and confirmation of the emergency landing option. Our app provides real-time display of safety helmet detection results along with video and image recording options.

B. Simple Upper-Side Cropping and HSV Filtering Method for Colorwise Safety Helmet Detection

There is a universal color code for safety helmets [19]. For example, white is for construction supervisors and blue is for electricians. The captured construction site videos/images for evaluation are in the RGB space. However, it is not easy to correctly determine the colors by RGB values because there are different shades, lighting and weather conditions. This motivates us to implement a simple and effective cropping and HSV filtering method [7, 10, 21] to determine safety helmets color in the HSV space, where the copping concept is the same as mentioned in [10]. In the HSV space, hue determines the desired color, saturation determines the intensity, and value determines the brightness. The detailed transformation from RGB to HSV [20] is formulated as follows.

$$H=\begin{cases} 0^\circ & , if\ \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right), & if\ C_{max}=R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta}+2\right), & if\ C_{max}=G', \\ 60^\circ \times \left(\frac{R'-G'}{\Delta}+4\right), & if\ C_{max}=B' \end{cases} \tag{1}$$

$$S=\begin{cases} 0, & if\ C_{max}=0 \\ \frac{\Delta}{C_{max}}, & if\ C_{max}\neq 0 \end{cases} \tag{2}$$

$$V=C_{max} \tag{3}$$

where C_{max} = max(R', G', B'), C_{min} = min(R', G', B') and $\Delta = C_{max} - C_{min}$, where R' = R/255, G' = G/255, and B' = B/255. In this work, R, G, and B values are normalized from [0, 255] to [0, 1]. This color space is particularly useful in predicting colors in continuous videos under different illumination and shading conditions. Tybusch *et al.* [21] also used HSV ranges to robustly determine pixel colors for a sampled image area. In this paper, our targets include five colors: blue, orange, red, white and yellow. Herein, we apply the same HSV filtering concept [7, 10, 21] to design the five-color HSV ranges to determine the color for each pixel, where users can set their own low and high HSV ranges. The simple and effective HSV filtering is shown in Fig. 3 and is described as follows. 1) Crop the upper-side of each bounding box detected by YOLOv3 and consider it as the region of interest (ROI) for each image, 2) transform the cropped ROI image to HSV color space using equations (1), (2), (3), 3) set the low and the high HSV ranges of each of the five colors, 4) use these HSV ranges to filter out the pixels that do not match the HSV ranges and then obtain its scores through a threshold function, 5) determine the safety helmet color by the highest score. Finally, we can count different helmet colors per input frame.

III. Evaluation Results

To evaluate the system performance of our detection and counting model, we use the accuracy defined below as the performance metric.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}\ . \qquad (4)$$

Similarly, to evaluate color detection, we use precision as our performance metric as defined below:

$$Precision = \frac{TP}{TP+FP}\ , \qquad (5)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. Since we would like to target safety-helmet and without-safety-helmet workers' detection and counting, we modify Eq. (4) as follows.

$$\text{Safety\_Helmet\_Detection\_Accuracy} = \frac{TH+TNH}{TH+FH+TNH+FNH}\ , \qquad (6)$$

where TH denotes true safety-helmet prediction, TNH denotes true no-safety-helmet prediction, FH denotes false safety-helmet prediction and true no-safety-helmet detection miss, and FNH denotes false no-safety-helmet prediction and true safety-helmet detection miss.

Similarly, we modify Eq. (5) as follows.

$$\text{True Color Precesion} = \frac{TC}{TC+FC}\ , \qquad (7)$$

where TC denotes true positive safety-helmet color prediction and FC denotes false positive safety-helmet color prediction. Eq. (6) and Eq. (7) are important to evaluate the real-time performance of our deep learning model and our cropping and filtering method on an edge device. We have considered 10 standard test cases. Among them, 3 cases are shown in Fig. 6. Our evaluation was performed in two different scenarios: 1) we use DJI Matrice 210 V2 for real outdoor testing at a maximum height of 50 m with a 30x zoom camera, as shown in Fig. 5 and Fig. 6(b), 2) we hold a DJI Mavic AIR in front of a PC screen with different YouTube videos and use the same software to check the detection performance, as shown in Fig. 6(a) and 6(c). Table 1 shows the detailed statistics for each case by randomly selecting two frames, denoted as SF1 and SF2. When checking the images from PC, it is worth noting that our simple cropping and HSV filtering method performs worse on poor video quality and almost washes out helmet colors, while it performs very well on real outdoor tests. We use both good and bad data to compute and assess safety helmet detection accuracy and true color precision through the developed system. In summary, the average safety helmet detection accuracy is 81.02% and the average true color precision is 80.86%. The performance evaluation also includes a comparison with other existing works as shown in Table 2. Our proposed system can achieve edge-controlled autonomous UAV navigation for outdoor construction sites with satisfactory accuracy and precision.

IV. Conclusions

In this paper, we have successfully developed 1) an edge-device-oriented outdoor autonomous flying control and computation system for off-the-shelf DJI Matrice 210 v2, 2) an outdoor modularized software system architecture framework integrating the DJI SDK, tiny YOLOv3, OpenCV and GPS waypoints, 3) a simple upper-side cropping and HSV filtering method for detecting and counting colorwise safety helmets and without-safety-helmet workers. As to future work, we will explore more complicated scenarios in construction sites.

(a)

(b)

(c)

Fig 6. Three case scenarios for safety-helmet observation (a) Case 2, (b) Case 9, and (c) Case 10.

Acknowledgement

The research is co-sponsored by ITRI, Pervasive Artificial Intelligence Research (PAIR) Labs, and Ministry of Science and Technology (MOST). This work is also financially supported by "Center for Open Intelligent Connectivity" of "Higher Education Sprout Project" of NYCU and MOE, Taiwan.

References

[1] S. Konda, H. M. Tiesman, and A. A. Reichard, "Fatal traumatic brain injuries in the construction industry, 2003-2010," American Journal of Industrial Medicine," vol. 59, no. 3, pp. 212–220, 2016.

[2] Safety helmets standards: [Online]. Available: https://www.graphicproducts.com/articles/osha-ansi-hard-hat-requirements/

[3] A. H. M. Rubaiyat, T. T. Toma, M. K. Khandani, S. A. Rahman, L. Chen, Y. Ye, C. S. Pan, "Automatic detection of helmet uses for construction safety," in Proc. IEEE/WIC/ACM Intelligence Conference of Web Intelligence Workshops, 2016, pp. 135–142.

[4] M. W. Park and I. Brilakis, "Continuous localization of construction workers via integration of detection and tracking," Automation in Construction, vol. 72, pp. 129–142, 2016.

[5] K. Yang, C. R. Ahn, M. C. Vuran, and H. Kim, "Collective sensing of workers' gait patterns to identify fall hazards in construction," Automation in Construction, vol. 82, pp. 166–178, 2017.

[6] Q. Fang, H. Li, X. Luo, L. Ding, H. Luo, T. M. Rose and W. An, "Detecting non-hardhat-use by a deep learning method from far-field

surveillance videos," Automation in Construction, vol. 85, pp. 1–9, 2018.
[7] Z. Kai and W. Xiaozhi, "Wearing safety helmet detection in substation," in Proc. IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE), 2019, pp. 206–210.
[8] S. Kang, S. Kim, B. Choi, M. W. Park, and W. Suh, "Development of smart helmet for monitoring construction resources based on image matching method," Journal of Imaging Science and Technology, vol. 63, no. 3, pp. 30403-1-30403-10, May 2019.
[9] J. Wu, N. Cai, W. Chen, H. Wang and G. Wang, "Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset," Automation in Construction, vol. 106, pp. 1–7, 2019.
[10] B. E. Mneymneh, M. Abbas, and H. Khoury, "Vision-based framework for intelligent monitoring of hardhat wearing on construction sites," Journal of Computing in Civil Engineering, vol. 33, issue 2, Mar. 2019.
[11] A. Asadzadeh, M. Arashpour, H. Li, T. Ngo, A. B. Hadiashar and A. Rashidi, "Sensor-based safety management," Automation in Construction, vol. 113, pp. 1–16, 2020.
[12] M. Zhang, R. Shi, and Z. Yang, "A critical review of vision-based occupational health and safety monitoring of construction site workers," Safety Science, vol. 126, pp. 1–18, 2020.
[13] S. H. Lee and C. S. Yang, "A real-time object recognition and counting system for smart industrial camera sensor," IEEE Sensors Journal, vol. 17, no. 8, pp. 2516–2523, Apr. 2017.
[14] G. Saxena, A. Gupta, D. K. Verma, A. Rajan, and A. Rawat, "Robust algorithms for counting and detection of moving vehicles using deep learning," in Proc. IEEE 9th International Conference on Advanced Computing (IACC), 2019, pp. 235–239.
[15] X. Yang, J. Gaspar, W. H. Lou, W. Ke, C. T. Lam, and Y. Wang, "Vision-based mobile people counting system," in Proc. ACM International Conference Proceeding Series, 2019, pp. 42–46.
[16] G. Farjon, O. Krikeb, A. B. Hillel, and V. Alchanatis, "Detection and counting of flowers on apple trees for better chemical thinning decisions," Precision Agriculture, vol. 21, pp. 503–521, 2019.
[17] Y. Y. Tseng, P. M. Hsu, J. J. Chen, and Y. C. Tseng, "Computer vision-assisted instant alerts in 5G," in Proc. International Conference on Computer Communications and Networks (ICCCN), 2020.
[18] Y. Y. Tseng, T. C. Hsu, Y. F. Wu, J. J. Chen, and Y. C. Tseng, "Efficient vehicle counting based on time-spatial images by neural networks," in Proc. IEEE International Conference on Mobile Ad-hoc and Smart Systems (MASS), 2021.
[19] The truth about safety helmet color code. Source: [Online]. Available: https://safels.com/safety-helmet-color-code/
[20] RGB to HSV color conversion. Source: [Online]. Available: https://www.rapidtables.com/convert/color/rgb-to-hsv.html
[21] D. Tybusch, G. R. D. Moraes, O. M. dos Santos, A. P. Legg, and S. R. Ceolin, "Color-based and recursive fiducial marker for augmented reality," in Proc. SIBGRAPI Conference on Graphics, Patterns and Images, 2017, pp. 254-261.

Table 1. Detection results per frame in the outdoor environment

	Case 1		Case 2		Case 3		Case 4		Case 5	
Body Posture	Full Body		Half Body		Bend		Stand		Sit	
	SF1	SF2	SF1	SF2	SF1	SF2	SF1	SF2	SF1	SF2
Total People	12	12	11	11	9	9	6	6	9	9
Real Safety Helmet	1	1	10	10	5	5	6	6	0	0
Safety_Helmet_Detection_Accuracy (%)	83.3	91.7	63.6	63.6	77.8	77.8	85.7	85.7	88.9	77.8
True_Color_Precision (%)	100	100	100	100	33.3	25	100	100	N/A	N/A
	Case 6		Case 7		Case 8		Case 9		Case 10	
Types of View/ Weather	Front view		Back view		Top view		Sunny		Cloudy	
	SF1	SF2	SF1	SF2	SF1	SF2	SF1	SF2	SF1	SF2
Total People	10	10	10	10	4	4	2	2	9	9
Real Safety Helmet	10	10	10	10	3	3	1	1	9	9
Safety_Helmet_Detection_Accuracy (%)	80	80	60	60	100	100	100	100	66.7	77.8
True_Color_Precision (%)	33.3	40	100	100	100	100	100	100	66.7	57.1

Table 2. Comparison results

Methodology	UAV Surveillance	Edge-Computing	Accuracy		
			Safety Helmet / without Safety Helmet Detection (%)	Color Detection	Worker with Safety Helmet / without Safety Helmet Counting
Faster R-CNN for non-hardhat-use [6]	No	No	98.4 (Precision)	No	No
TensorFlow+HSV filter [7]	No	No	89 (Accuracy)	(Red, Blue, Yellow, White)	No
RPA+SSD [9]	No	No	83.89 (mAP) 90.86 (Precision)	(Red, Blue, Yellow, White)	No
Cascade classifier+color classification [10]	No	No	94.65 (Precision)	(Blue, Yellow, White, Orange)	No
DJI SDK+ tiny YOLOv3+OpenCV+GPS waypoints in Android phone [Our work]	Yes	Yes	81.02 (Accuracy)	(Red, Blue, Yellow, White, Orange)	Yes