Instant Counting & Vehicle Detection during Hajj Using Drones

Abdullah M. Algamdi[1, 2,*] and Hammam M. Alghamdi[1]

[1] Department of Computer Science, University of Jeddah, Jeddah, Saudi Arabia;
Email: hmsalghamdi@uj.edu.sa (H.M.A.)
[2] King Abdulaziz University, Jeddah, Saudi Arabia
*Correspondence: amalgamdi6@uj.edu.sa (A.M.A.)

***Abstract*—During the past decade, artificial intelligence technologies, especially Computer Vision (CV) technologies, have experienced significant breakthroughs due to the development of deep learning models, particularly Convolutional Neural Networks (CNNs). These networks have been utilized in various research applications, including astronomy, marine sciences, security, medicine, and pathology. In this paper, we build a framework utilizing CV technology to support decision-makers during the Hajj season. We collect and process real-time/instant images from multiple aircraft/drones, which follow the pilgrims while they move around the holy sites during Hajj. These images, taken by multiple drones, are processed in two stages. First, we purify the images collected from multiple drones and stitch them, producing one image that captures the whole holy site. Second, the stitched image is processed using a CNN to provide two pieces of information: (1) the number of buses and ambulances; and (2) the estimated count of pilgrims. This information could help decision-makers identify needs for further support during Hajj, such as logistics services, security personnel, and/or ambulances.**

***Keywords*—unmanned aerial vehicles, deep networks, instant crowd counting, vehicle detection, image stitching, Hajj**

I. Introduction

Hajj is a ritual practice for millions of Muslims, who travel to a specific location (located in Mecca, Saudi Arabia) at a specific time during the lunar year. Every Muslim must perform Hajj once in his or her lifetime as part of their religion. Each year, Saudi Arabia welcomes around 1 to 2 million foreign pilgrims. For example, in 2018 (right before COVID-19) Saudi Arabia received 1,758,722 individuals [1]. The total number of pilgrims can reach 2.5 million, including local pilgrims.

During the Hajj season, all pilgrims move from one holy site to another at an exact time. It is a challenge for the Saudi government to facilitate this process for pilgrims. Moreover, the holy sites are small (e.g., Muzdalifah = 20 square km [2]), so they have difficulty accommodating such a large number of pilgrims. In addition, cars, buses, and ambulances occupy holy sites, increasing the difficulty

During the past years, some incidents have caused the death of innocent individuals. For example, 717 pilgrims died during a heart-breaking stampede that occurred in 2015, according to the Saudi Press Agency [3]. Some of these incidents happen due to the pilgrims' ignorance or their disobedience to the government's rules, procedures, and schedules, which are carefully designed to protect the pilgrims' safety.

The government of Saudi Arabia is taking this matter seriously. However, there are some breaches that require real-time monitoring and observation so that officials and decision-makers can intervene and act accordingly before a disaster occurs or accelerates. With the rapid evolution of technology, computer vision models have significantly contributed to monitoring and observation applications. The robust and widely used CNNs are able to provide detailed information about holy sites in real time. Such instant information supports decision-making processes by providing facts on the ground rather than reports or speculation.

Our framework has two main stages. First, it takes real-time images from multiple drones covering the holy sites. Due to the challenge of photographing an entire holy site with a single drone, we first stitch the collected images, generating one single image covering the entire holy sites. The stitching stage helps to reduce the redundancy in the information provided in the second stage. Next, we perform object detection and crowd counting using CNNs. As a result, our framework produces instant information (i.e., the number of buses and pilgrims occupying the holy sites).

This paper is organized as follows. We present selected recent work related to image stitching, object detection, and counting in Section II. In Section III, we describe the methodology of our proposed framework, followed by a presentation of our results in Section IV. In Section V, we discuss the experimental setup and the steps of the proposed approach. Section VI concludes the report and presents some future work directions.

Manuscript received October 20, 2022; revised November 16, 2022, accepted November 30, 2022.

II. Related Work

Existing solutions have focused on monitoring and managing crowds during Hajj. They represent a systematic alternative to the current manual crowd monitoring and management approach.

A. Crowd Monitoring and Management during Hajj

Baqui *et al.* [4] proposed a framework that processes real-time videos captured from surveillance cameras. The videos are processed using pedestrian velocity extraction by incorporating cross-correlation between different image frames. As a result, high-density pedestrian traffic is highlighted. Another framework was suggested by Felemban *et. al.* [5] to speed up the transmission of information from videos captured by drones during Hajj. The framework prioritizes the selection of critical image data. Moreover, Nasser *et al.* [6] proposed a framework that monitors crowds on the paths and roads leading to holy sites. Their framework analyzes the data collected from Information and Communication Technology (ICT) sensors such as e-bracelets, smartphones, and Radio-Frequency IDentification (RFID) information. Next, it suggests the optimal schedule, i.e., with a shorter arrival time accommodating larger number of pilgrims. The data are collected from multiple sensor sources, suggesting the importance of Internet of Things (IoT) technology in today's technical applications [7, 8].

To the best of our knowledge, there is no instant crowd monitoring solution for the Hajj season based on drone images processed using a deep learning model. Nevertheless, studies have made remarkable and significant contributions in terms of deep models for solving object detection and crowd-counting problems.

B. Object Detection and Counting

Many methods and techniques have been proposed for the task of car detection. For example, hand-crafted features have been proposed to solve the object counting problem. Moranduzzo and Melgani [9, 10] proposed a Scale-Invariant Feature Transform (SIFT) and Support Victor Machine (SVM) to detect and classify objects. Then, the same SIFT points that belong to the same car are merged. Shao *et al.* [11] proposed a sliding window search, with filtering operations per- formed to extract HOG features. Similar vehicles are detected by measuring the similarity between detected objects. Xu *et al.* [12] proposed using Viola-Jones and Histogram of Gradient (HOG) with SVM and a detector switching strategy to improve both speed and detection efficiency. Recently, deep learning networks have outperformed hand-crafted features in many ways. One author of [13] applied a sliding window approach and CNNs on multiple scales to detect cars in the image. Due to the computation cost of the sliding window approach, a region proposal network was proposed with segmentation to detect vehicles in images [14, 15]. Hsieh *et al.* [16] introduced an RGB dataset for car counting. Another large-scale dataset was proposed by Zhu *et al.* [17] for object counting and object detection with RGB and thermal infrared (RGBT) images.

Zhu *et al.* [18] collected the largest UAV dataset on multi-task applications.

C. Crowd Counting

The crowd counting problem can be divided into two solution groups — the traditional solution, which is handcrafted, and CNN-based solutions. Since deep learning is outperforming traditional hand-crafted feature extraction, we focus on the CNN-based algorithms. Many methods have been proposed to solve the problem of crowd counting based on counting the number of heads [19−21]. However, these methods are complicated and cannot be applied to real-life scenarios. Further, background noise is often added to the crowd count [22]. A different approach is to generate a density map of the crowd and treat it as a regression problem to get the estimated crowd count. contextual pyramid CNN (CP-CNN) [23] was proposed to solve the problems of underestimation and overestimation in crowd counting by leveraging the context at different levels. This work fits with our objective by generating a density map and accurately estimating the counts from different elevations. Our research adopts these methods and approaches to fit the Hajj context. Specifically, the aim is to cover the large area of Hajj holy sites using state-of-the-art methods and techniques, including drones, crowd density maps, and object counting.

III. Methodology

This project aims to capture high-level information during Hajj, including the distribution of pilgrims and Hajj vehicles. This project is designed in two main stages:

(1) image collection and stitching (pre-processing) and
(2) extraction of high-level knowledge.

A. Stage 1: Image Collection and Stitching

We operate multiple drones (e.g., four drones) which follow pilgrims around the holy sites during Hajj. A single drone could be used for this purpose. However, capturing detailed information, such as identifying small objects (cars & Hajj officers) requires a high-resolution image, which might be challenging to acquire with a single drone.

The collection of information from multiple drones requires some image preparation and pre-processing. In this stage, we stitch the collected images producing a single high-resolution image for the entire holy site. This stitched high-resolution image is then input to the second stage, where we extract high-level information about the holy sites and pilgrims.

To the best of our knowledge, the available image-stitching approaches in the literature are not capable of stitching more than two images. Therefore, we designed an approach that stitches multiple images from multiple drones. An overview of the proposed stitching approach is shown in Fig. 1.

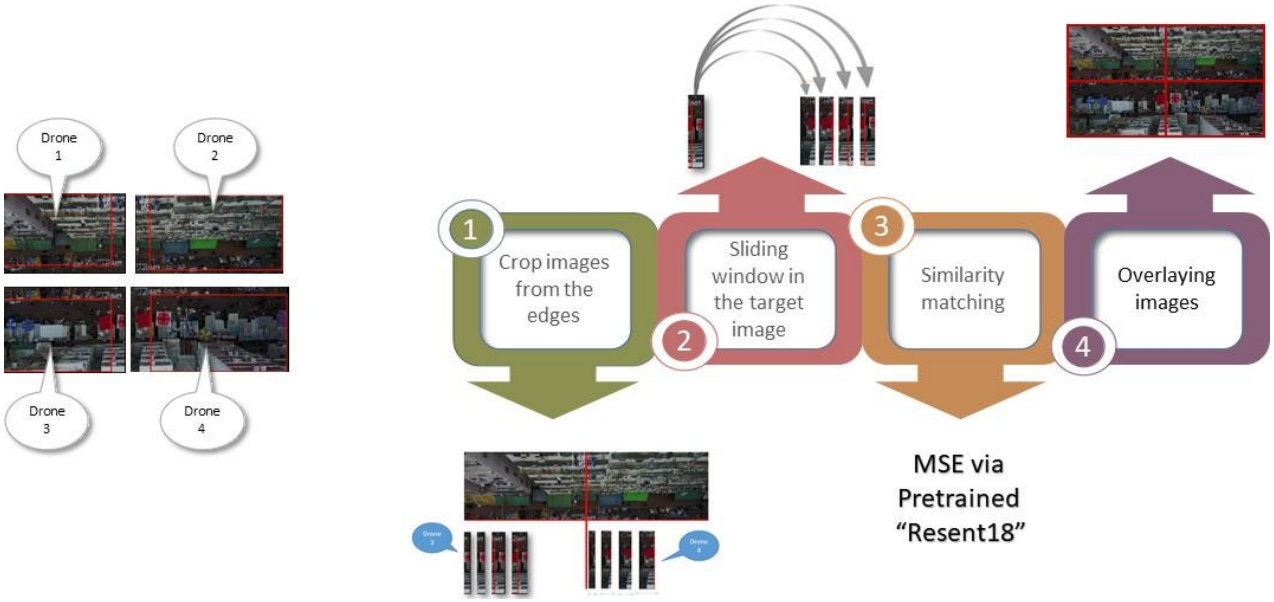

Figure 1. An overview of the proposed stitching pipeline.

Let us say that we have four drones, each of which captures an angle of the holy site (Fig. 2). We then conduct an "anti-clockwise" iteration, where, for example, the image captured by *Drone 1* is stitched with the image from *Drone 3* (as presented in Fig. 3). For each stitching iteration, we set one drone as the source image (S_{img}) and the other image is the target image (T_{img}). Table I shows details of these iterations.

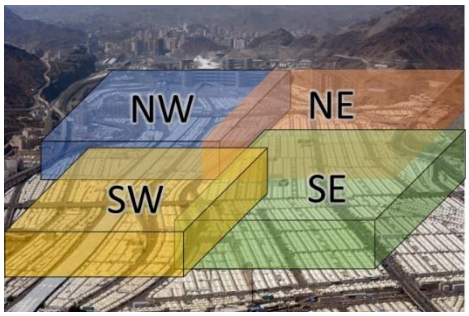

Figure 2. The distribution of the drones around the holy site.

TABLE I. IMAGE-STITCHING ITERATIONS FOR THE SOURCE IMAGE (S_{IMG}) AND THE TARGET IMAGE (T_{IMG})

Iteration	S_{img}	T_{img}
A	Drone 1	Drone 3
B	Drone 3	Drone 4
C	Drone 4	Drone 2
d	Drone 2	Drone 1

In this stage, we aim to overlay the overlapping portions of the neighboring drones. To do so, we crop multiple slices of the S_{img} from the side of the T_{img}.

For example, to stitch images from *Drone 3* and *Drone 4* (iteration *B* in Table I), we crop multiple slices from the east edge of the image from *Drone 3* in order to find the best stitching overlap with the image from the image from *Drone 4* (Fig. 4). The first slice is the left 5% of the width of the image from *Drone 3*, whereas the other slices are 10%, 15%, and 20% from the same east edge.

a) Sliding window: After cropping each slice, we apply the sliding window in the T_{img} and check the similarity between the slices from the S_{img} and T_{img}, allowing us properly stitch those two images. A total of 16 "similarity matching" comparisons are applied, if we set the number of slices = 4, and the number of sliding window steps = 4.

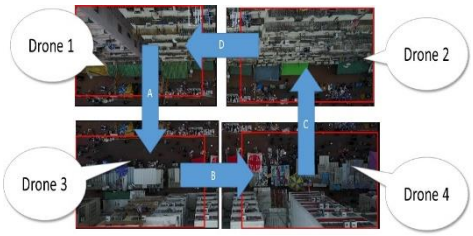

Figure 3. The "anti-clockwise" iterations for stitching the drone images.

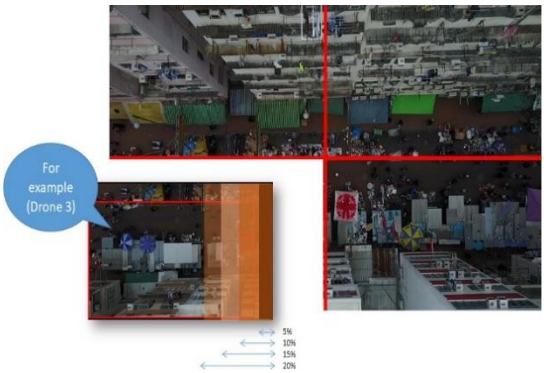

Figure 4. Cropping slices from the S_{img} image.

1) Similarity matching

To Match two slices from the S_{img} and T_{img} images, we utilize the pre-trained Resnet18 [24]. with ImageNet [25]. Each slice is fed into the model as feed-forward manner, and then we extract the vectors of both slices from the last fully connected layer (i.e., the layer

before the prediction layer). Lastly, we check the similarity between those two vectors by calculating the MSE (i.e., "mean squared error (MSE)"). The slices with the lowest MSE, among all comparisons (i.e., 16 comparisons in our situation), would be the chosen for stitching S_{img} and T_{img}.

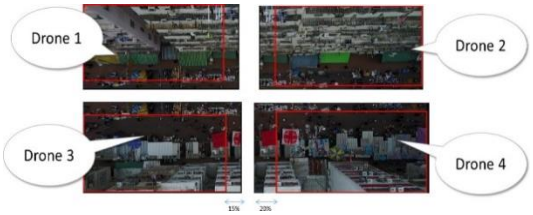

Figure 5. The newly re-generated drone dataset.

2) Dataset creation

To the best of our knowledge, there is no existing dataset for Hajj and pilgrims, and there is also no existing dataset based on images from multiple drones capturing the same site. Due to this limitation, we generated our own dataset from an existing single-drone dataset, i.e., VisDrone [26]. We cropped four different images from each corner (i.e., NW, NE, SW, and SW). Each drone image was cropped with a random extra portion from the neighboring drone corners, as shown in Fig. 5. For example, the extracted image for *Drone 4* has an extra portion of 20% from the coverage area of *Drone 3*, whereas *Drone 3* has 15% from the coverage area of *Drone 4*. Our new dataset was generated in this way from all the images of VisDrone dataset. This dataset is publicly available for any further evaluation and studies.

a) Extracting ground truth: While generating our dataset, we recorded the ground truth (GT), including the random extra portions that were included for each drone image extracted from the VisDrone dataset. A snapshot of the GT is presented in Fig. 6. The GT file is also publicly available, along with our generated image dataset.

3) Evaluation

To evaluate this approach, we compare the prediction with the extracted GT. The approach predicts two pieces of information: (1) the amount of context in the S_{img} that is overlapping with the T_{img} (which is the so-called S_{ratio}) and (2) the number of steps in the T_{img} (i.e., n_steps). The S_{ratio} is either 5%, 10%, 15%, or 20%. So, the final prediction for each stitching iteration is calculated as follows: $S_{ratio} \times n_steps$.

Folder	Img	Drone	T	B	L	R
1	1	1	-	0.05	-	0.2
1	1	2	-	0.1	0.1	-
1	1	3	0.15	-	-	0.15
1	1	4	0.05	-	0.05	-
1	2	1	-	0.05	-	0.05
1	2	2	-	0.05	0.2	-
1	2	3	0.1	-	-	0.2
1	2	4	0.05	-	0.05	-
1	3	1	-	0.15	-	0.15
1	3	2	-	0.2	0.2	-
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Figure 6. A snapshot of the GT table.

The GT of an iteration is the sum of both ratios, which is designated as $GT_{iteration}$. For instance, iteration *C* in Table I is for stitching *Drone 4* and *Drone 2*, so the GT of this iteration is sum of the "T" ratio of *Drone 4* and the "B" of *Drone 2* (see Fig. 6). Therefore, the evaluation of each iteration is computed using the following Equation:

$$Accuracy = 1 - |GT_{iteration} - prediction| \quad (1)$$

B. Stage 2 (a): Object Detection

Vehicle detection*:* Since we do not have a GT for our Hajj video from the drone, we choose the [27] YOLOv5 object detector as our baseline object detector. YOLOv5 is best known for its real-time performance and high precision. It meets our needs for the Hajj season since it Is fast and reliable to use.

YOLOv5 uses multi-scale feature extraction to detect objects at different scales. The model divides the image into grid cells, and each grid cell predicts the object, confidence score, and four bounding box coordinates. The YOLOv5 model outputs multiple sets of feature maps with different scales. Each grid cell of each scale outputs multiple prior bounding boxes. In all scales, each grid cell predicts specific number of predictions prior to the bounding boxes.

When predicting the location of the object in all scales, the output would generate multiple bounding boxes around the object. The Non-Max Suppression method in YOLOv5 is used to select the best bounding box and rejects another bounding box around the object. The method accepts the objectiveness score of the model and the Intersection Over Union (IOU) of the bounding box. Although the model performs well on object detection benchmarks, our data will be different. We aim to detect specific objects, such as vehicles. To solve the problem of detecting specific objects, we propose to fine-tuning the detector with a vehicle dataset. Fine-tuning YOLOv5 with such a dataset will improve the performance without harming the processing speed.

C. Stage 2 (b): Crowd Counting

Density map-based counting methods are popular for crowd counting due to their use of deep neural networks (DNNs). These methods predict the density maps from crowd images using DNNs. To get the estimated count of the crowd, most of the current methods sum the density maps. The work proposed by [28] achieved state-of-the-art counting performance on many large-scale datasets. The authors propose the use of a generalized loss function for the density maps for crowd counting. Further, a perspective-guided transport cost function was proposed for better estimation of the crowd count and localization.

a) Crowd counting for real Hajj scenes: There are special requirements for counting crowds during Hajj, and it poses many challenges. First, individuals wear identical White clothing, which makes detection challenging. In addition, some vision blockers are expected at hajj sites, such as umbrellas, trees, and bridges, which limit vision and hamper the crowd-counting performance. In addition, the Hajj scenes are recorded by a drone, which is another challenge for the detection task, as the image is captured

from an unfamiliar perspective. Furthermore, in the Hajj, there is no GT, so we cannot be able to compare the performance of different crowd-counting methods.

Considering the aforementioned challenges, we can assume that the closest dataset to our Hajj scenes is the UCF-QNRF [22]. A sample of this dataset is depicted in Fig. 7. One of the dataset sources is Hajj footage, which is selected carefully. According to the authors proposed in [29], the Hajj images were captured from multiple places with different viewpoints and perspectives. Further, the images were taken at different times of the day for better generalization results.

Figure 7. A sample of UCF-QNRF dataset [22].

The dataset consists of 1353 images with 1,251,642 annotations. The average number of annotations per image is 815. The maximum number of annotations per image is 12,865 and the average resolution of the images is 2013×2902 pixels. These images were collected from different parts of the world according to the geo-tags of the images.

For our Hajj videos, we assumed that the methods that achieved state-of-the-art performance on the UCF- QNRF would also be the most efficient for our chosen method. We chose [28] as our baseline model, which can guarantee an accurate count estimation of our crowd scenes during Hajj.

IV. Results and Evaluation

A. Image Stitching

The VisDrone dataset contains around 1234 images, each of which is stitched four times, following the "anticlockwis" iteration explained in Table I. We calculate the median value of the accuracies computed from Eq. (1). The overall accuracy for the entire approach is 70%, which is the median value for our generated image dataset.

B. Vehicle Detection and Crowd Counting

Since we are using two methods in our pipeline, one for vehicle detection and the other for crowd counting, we pay attention to the processing time as well as performance. The processing frame rate should not exceed the most common frame rate in the live stream so we can provide a fast estimation for the decision-makers.

For vehicle detection, we adjusted the YOLOv5 hyper-parameters so that they can detect small-to-tiny objects, as we used video captured by drones. Similarly, we adjusted the dilation rate in every layer of the architecture in [28] so that it can provide a more reasonable estimation of the crowd.

1) Performance on real Hajj video

Due to the limited number of Hajj videos captured with drones, we choose two videos to show the performance of our framework. The first video (V1) captures pilgrims walking on a bridge. The second video (V2) is recorded by a drone in the area where pilgrims are walking inside a housing compound. The vehicle in the V1 video is hardly seen, but it can be clearly seen in V2. In the qualitative results, we show only vehicles detected in V2.

2) Qualitative results

The first set of results is depicted in Fig. 8. The second set includes the estimated counting in V1, while the third set is the estimated crowd counting and vehicle detection. In the first set, we show a sample of vehicles detected in V2. The detected vehicles are captured with a drone from a challenging viewpoint. However, with fine-tuning YOLOv5, it was able to detect these vehicles and provide their count and locations.

Figure 8. Vehicle detection with YOLOv5 after fine-tuning the weights to detect tiny objects and only vehicles included in Microsoft Common Objects in Context (COCO) dataset labelling.

The second set of results depicted in Figs. 9−11 show the qualitative performance of the crowd counting module. The figures show the estimated crowd counting on one of the Hajj bridges. The V1 is limited to crowd counting since no vehicles appear in the video frames.

Figure 9. The first figure in the V1 set of results.

Figure 10. The second figure in the V1 set of results.

Figure 11. The Third figure in the V1 set of results.

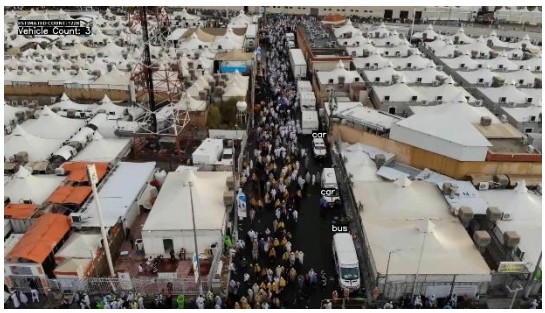

Figure 12. The first figure in the V2 set of results.

Figure 13. The second figure in the V2 set of results.

Figure 14. The third figure in the V2 set of results.

Figure 15. The fourth figure in the V2 set of results.

In the third set of results, Figs. 12−15 show both crowd counting and vehicle detection. Fig. 12 shows mixed detection between crowds and vehicles. As shown in the figure, the crowd counting and vehicle detection modules perform well when the videos are captured from drones with different viewpoints and azimuths. In Fig. 13, a limited number of people are captured by drones from an overhead position, making it challenging to detect people and count them. However, the method was able to estimate the count with acceptable precision. Fig. 14 shows the highest estimation crowd count in among the whole result sets. It is challenging to estimate the actual number, but the crowd-counting module estimates the crowd count instantly. Fig. 15 depicts the same scene with a smaller crowd, and the method behaves as expected by estimating a smaller number of people.

Technologies, such as Smart Healthcare, Smart Agriculture, and Smart Cities. For example, gathering location information from e-bracelets, which are used as smart health assistants, along with an analysis of drone images, would lead to more precise and accurate crowd estimation. Utilizing the Internet of Vehicles would also help to gather real-time data on the routes that the vehicles take during the Hajj season. Further research studies are needed to explore the impact that these emerging technologies would have on crowd estimation and vehicle detection problems during Hajj season.

V. Discussion

A. Image Stitching

Controlled locations of drones: Our approach assumes that the drones are distributed in predefined locations. The number of drones should be fixed throughout the season, and each should be controlled to cover a specific location (e.g., the southwest corner of the holy Site). Moreover, the approach is scalable, where the number of drones can be increased to 6, 9, or even more.

Predefined overlapping ratios: In our experiment, the overlapping ratios are predefined (5, 10, 15, 20) %. However, in real-time implementations, the ratios can be adjusted based on the desired ratios. The smaller intervals (e.g., 2.5%, 5%, 7.5%) should lead to more precise stitching but are more computationally expensive.

Anti-clockwise comparison: We chose to apply an anti-clockwise iteration to stitch the images from multiple drones. However, a clockwise iteration or random selection of the S_{img} and T_{img} could be applied between every two neighbouring drones.

B. Object Detection and Crowd Counting

Our qualitative results differ at the vehicle and crowd levels. The aim of our research is to provide estimations for the decision-makers to enable the smooth flow of crowds and vehicles. Our results show that the proposed method is able to estimate crowds effectively. Moreover, the results are not limited to crowds only; and the model also estimates the number and the types of vehicles detected by the drones. Knowing the vehicle count can help decision-makers to determine whether the number of vehicles is reasonable. Moreover, knowing the type of vehicle is important for decision-makers. For example, detecting a low number of ambulances might cause decision-makers to request more vehicles for the health authority. Similarly, detecting too many buses on the road, which can cause slow traffic would allow decision- makers to inform the traffic authority to reduce the number of vehicles.

C. Future Research Direction

Problems studied here, i.e., crowd estimation and vehicle detection, could benefit from other emerging technologies, such as Smart Healthcare, Smart Agriculture, and Smart Cities. For example, gathering location information from e-bracelets, which are used as smart health assistants, along with an analysis of drone images, would lead to more precise and accurate crowd estimation. Utilizing the Internet of Vehicles would also help to gather real-time data on the routes that the vehicles take during the Hajj season. Further research studies are needed to explore the impact that these emerging technologies would have on crowd estimation and vehicle detection problems during Hajj season.

VI. Conclusion

In this project, we develop an approach that extracts high-level information to support decision-makers during the Hajj season. Specifically, multiple drones follow pilgrims around the holy sites. Each drone captures images of a corner of a holy Site. Our proposed approach first stitches the images captured by multiple drones, providing a single high-resolution image of the entire holy Site. Then, we extract high-level information, for example, highlighting crowded areas and identifying the number and locations of buses and ambulances.

In the future, datasets specifically created for holy sites and pilgrims are needed to improve fine-tuning and evaluation of Hajj-related approaches. Additionally, as holy sites are wide and vast, they cannot be covered using a single drone. Therefore, approaches enabling more accurate and precise stitching of images from multiple drones would benefit Hajj-related approaches.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

AA and HA prepared the data; HA conducted the image stitching experiments; AA conducted crowd counting and object detection experiments; AA and HA wrote the paper; all authors had approved the final version.

Funding

This research work was funded by Makkah Digital Gate Initiative under Grant No. (MDP-IRI-12-2020). Therefore, the authors gratefully acknowledge technical and financial support from the Emirate of Makkah Province and King Abdulaziz University, Jeddah, Saudi Arabia.

References

[1] General Authority for Statistics. The total number of pilgrims in 1439H Hajj season reached (2.371.675) pilgrims. [Online]. Available: https://www. stats.gov.sa/en/news/280

[2] M. Yamin, "Secure and healthy hajj management: a technological overview," *American Academic & Scholarly Research Journal*, vol. 7, no. 3, 2015.

[3] Saudi Press Agency. Civil Defense: Death toll rises to 453 dead and 719 injured in Mina stampede incident. [Online]. Available: https://www.spa. gov.sa/viewstory.php?newsid=1402015

[4] M. Baqui and R. Löhner, "Towards real-time monitoring of the Hajj," *Collective Dynamics*, vol. 5, pp. 394–402, 2020.

[5] E. Felemban, A. A. Sheikh, and A. Naseer, "Improving response time for crowd management in Hajj," *Computers*, vol. 10, no. 4, 2021.

[6] N. Nasser, M. Anan, M. F. C. Awad, H. B. Abbas, and L. Karim, "An expert crowd monitoring and management framework for Hajj," in *Proc. 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2017, pp. 1–8.

[7] Q. V. Khanh, N. V. Hoai, L. D. Manh, A. N. Le, and G. Jeon, "Wireless communication technologies for IOT in 5g: Vision, applications, and challenges," *Wireless Communications and Mobile Computing*, 2022.

[8] V. K. Quy, V. H. Nam, D. M. Linh, and L. A. Ngoc, "Routing algorithms for manet-iot networks: A comprehensive survey," *Wireless Personal Communications*, pp. 1–25, 2022.

[9] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635–1647, 2013.

[10] T. Moranduzzo and F. Melgani, "A sift-SVM method for detecting cars in UAV images," in *Proc. 2012 IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 6868–6871.

[11] W. Shao, W. Yang, G. Liu, and J. Liu, "Car detection from high-resolution aerial imagery using multiple features," in *Proc. 2012 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2012, pp. 4379–4382.

[12] Y. Z. Xu, G. Z. Yu, Y. P. Wang, X. K. Wu, and Y. Ma, "A hybrid vehicle detection method based on viola-jones and hog+ SVM from UAV images," *Sensors*, vol. 16, no. 8, 2016.

[13] X. Y. Chen, S. M. Xiang, C. L. Liu, and C. H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1797–1801, 2014.

[14] X. Y. Wang, M. Yang, S. H. Zhu, and Y. Q. Lin, "Regionlets for generic object detection," in *Proc. IEEE International Conference on Computer Vision*, 2013, pp. 17–24.

[15] J. R. R. Uijlings, K. E. V. D. Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[16] M. R. Hsieh, Y. L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proc. IEEE International Conference on Computer Vision*, 2017, pp. 4145–4153.

[17] P. F. Zhu, Y. M. Sun, L. Y. Wen, Y. Feng, and Q. H. Hu, "Drone based RGBT vehicle detection and counting: A challenge," Computer Science, 2020.

[18] P. F. Zhu, L. Y. Wen, D. W. Du, X. Bian, H. Fan, Q. H. Hu, and H. B. Ling, "Detection and tracking meet drones challenge," ArXiv, 2020.

[19] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Advances in Neural Information Processing Systems*, vol. 23, 2010.
[20] C. Zhang, K. Kang, H. S. Li, X. G. Wang, R. Xie, and X. K. Yang, "Data-driven crowd understanding: A baseline for a large-scale crowd dataset," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1048–1061, 2016.
[21] D. B. Sam, N. N. Sajjan, R. V. Babu, and M. Srinivasan, "Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3618–3626.
[22] J. H. Huang, X. G. Di, J. D. Wu, and A. Y. Chen, "A novel convolutional neural network method for crowd counting," *Frontiers of Information Technology and Electronic Engineering*, vol. 21, no. 8, pp. 1150–1160, 2020.
[23] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid CNNS," in *Proc. IEEE International Conference on Computer Vision*, 2017, pp. 1861–1870.
[24] K. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[25] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
[26] P. F. Zhu, L. Y. Wen, D. Du, X. Bian, H. B. Ling, Q. H. Hu, Q. Q. Nie, H. Cheng, C. F. Liu, X. Y. Liu, *et al*., "Visdrone-det2018: The vision meets drone object detection in image challenge results," in *Proc. European Conference on Computer Vision (ECCV) Workshops*, 2018.
[27] G. Jocher, A. Stoken, *et al*., "Ultralytics/yolov5: v3.1 — Bug fixes and performance improvements," October 2020.
[28] J. Wan, Z. Liu, and A. B. Chan, "A generalized loss function for crowd counting and localization," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1974–1983.
[29] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 532–546.