Real-Time Detection, Tracking and Classification of Multiple Moving Objects in UAV Videos

Hüseyin Can Baykara*, Erdem Bıyık*, Gamze Gül*, Deniz Onural*, Ahmet Safa Öztürk*, İlkay Yıldız*
Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey
{cbaykara, ebiyik, gamzegul, donural, ahmetsafaozturk, ilkyyldz95}@alumni.bilkent.edu.tr
* Authors contributed equally.

***Abstract*—Unnamed Aerial Vehicles (UAVs) are becoming increasingly popular and widely used for surveillance and reconnaissance. There are some recent studies regarding moving object detection, tracking, and classification from UAV videos. A unifying study, which also extends the application scope of such previous works and provides real-time results, is absent from the literature. This paper aims to fill this gap by presenting a framework that can robustly detect, track and classify multiple moving objects in real-time, using commercially available UAV systems and a common laptop computer. The framework can additionally deliver practical information about the detected objects, such as their coordinates and velocities. The performance of the proposed framework, which surpasses human capabilities for moving object detection, is reported and discussed.**

Keywords – surveillance, moving object detection, object tracking, aerial image classification, deep learning, transfer learning, real time, low altitude, aerial video, automation

I. Introduction

UAVs have become an essential part of surveillance and reconnaissance in recent years. Beside their widespread usage in the entertainment and media production industries, UAVs are used for many military and civil applications. These applications include search and rescue, traffic control, border patrol and security. The use of UAVs for security applications requires an operator that should be able to process the information rather quickly. Since human operators are generally quite behind these criteria, computers with the ability to meet these specifications with consistent reliability, accuracy, and precision, at quick rates and for a low cost, are expected to surpass human operators.

An important application of aerial surveillance for security purposes is to detect and classify moving objects in the observed area. There are some studies in the literature that accomplish either detection, or classification, or both with limitations. In their study Máttyus et al. [1] accomplished object detection and tracking for vehicles and humans in real-time from low altitude aerial surveillance. In addition to the detection and tracking, Iwashita et al. [2] also performed classification with 80% accuracy for vehicles and humans but the operation was not in real-time. Another similar study is conducted by Oreifej et al. [3] where they performed detection and classification for humans but not tracking. In [3], classification accuracy was 85% and the operation was not in real-time. Although detection, tracking and classification were achieved in real-time in [4], the camera was stationary and much closer to the moving objects, which makes the entire process easier due to higher resolution, larger target objects, and fewer noise sources.

To the best of our knowledge, previous studies that perform detection, tracking and classification collectively do not work in real-time. Useful information that complement aerial videos are also provided by UAVs, such as geographical coordinates, telemetry data, and flight information; other studies have neglected to utilize such information for object localization. Hence, these systems should be improved to operate in real-time, and their scope can be extended.

Here, we propose a framework that performs moving object detection, tracking, and classification for vehicles and humans in real-time for low-altitude (near 100 meters) aerial surveillance. Our technique also uses telemetry data of the UAV to calculate the GPS coordinates of the classified objects.

II. Methods

The main purpose of this study is to propose a robust framework that performs moving object detection, tracking and classification in real-time, with computation power requirements that can be met with a common laptop computer. We start this section with an overview of the proposed framework, and then describe the components.

The main blocks of the algorithm are presented in the flowchart given in Fig. 1. Instead of designing a unidirectional system, we leverage the information from tracking and classification to improve detection performance. Relative position and size mean the position and size of objects in terms of pixels.

A. Image Undistortion

We start processing after the acquisition of a video frame as an image. While camera lenses help widening or shrinkening the field of view, they also deform the images. These distortions have an essential negative impact on image registration [5]. In order to remove the effect of the lens from the retrieved images, lens parameters, i.e. intrinsic, extrinsic, and lens distortion parameters, were extracted via camera calibration, which enables image undistortion [6].

As the camera lens is the same throughout a video (or a flight), we compute the mapping between the original image and the destination image in advance, and use it for proceeding video frames. This precomputation step significantly speeds up the undistortion process.

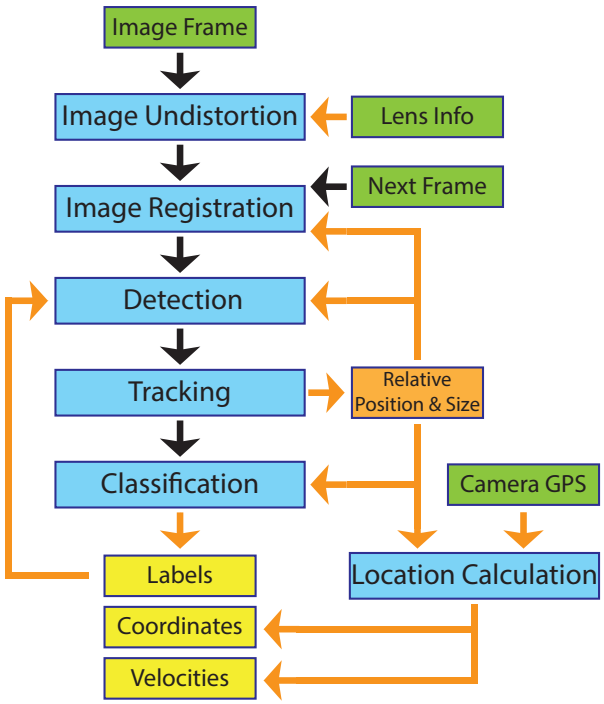

Fig. 1: General flowchart of the proposed framework. Blue blocks show the main components. Green and yellow blocks represent the input and the output, respectively. While black arrows show the flow of processed images, orange items are other information that the framework utilizes.

B. Image Registration

While working with a mobile camera, image registration is one of the key steps for surveillance. To align two images, one should determine the keypoints of the images, and then match them. The matched keypoints are used to compute a transformation between images, which performs the alignment.

Every frame that enters the registration step is converted into a grayscale image for fast computation of the remaining processes. These images are processed until the classification step where color images are used. The registration process continues with the alignment of consecutive frames.

Corner points are considered as keypoints. In literature, several keypoint detectors have been proposed. While some of them are suboptimal in terms of image registration performance, some suffer from long computation times. In [7], authors showed the FAST corner detector outperforms other techniques in most cases in terms of repeatability, which measures the ability to detect the same corner points under geometric and photometric transformations [8]. We employ the FAST algorithm because it is designed for high speed performance [7], which is critical for our framework. We extract 200 best-response keypoints per image, which is adequate for keypoint matching and does not immensely slow down the process.

To match the keypoints of two images accurately, it is important to employ a high-performance keypoint descriptor. In [9], authors showed the superiority of FREAK descriptors over its competitors, in terms of recall values and computation times. In the proposed framework, keypoints are matched based on the closest distance between their FREAK descriptors. Using these matchings, homography transform matrix between two images are computed with a RANSAC method, which has been preferred for its robustness under noisy conditions. At the end, we align the frame onto the next consecutive frame via the homography matrix as [1] suggests.

We also transform the relative positions of previously detected objects in order to keep the track of all objects on the matched frames. Without this important step, tracking would fail when the camera moves fast.

C. Moving Object Detection

Having two consecutive images aligned, frame differencing is a widely known technique for moving object detection for its favorable performance and speed [10, 11].

One problem associated with frame differencing is that it is vulnerable to the errors caused by image registration. When the alignment of the two images undergoing differencing is faulty, misdetections occur at the sharp color transition points. To suppress this problem, median filtering is generally employed. In our framework, we use mean filter for its significantly higher speed. Under the assumption that moving objects are not too small in terms of pixel size (i.e. the camera is high resolution and/or the camera is not at very high altitudes), we proportionally select the window size of the filter to video resolution. For a 1080p video, a square window size of 7 was found to be optimal considering the detection accuracy.

To binarize the image, we apply simple thresholding. If the number of pixels found to be moving is higher than 10% of the entire image, we consider the images to be misaligned and go back to the image registration step with the next frame. If such a problem does not occur, we perform connected component labeling, and then apply morphological dilation to each individual object by using a kernel identical to the binarized object itself. The purpose of this adaptive dilation step is to combine objects that are moving as parts of the same entity, e.g. hood and trunk parts of a car.

Having detected moving object candidates, we perform a "median across frames" step to eliminate detections that are actually noise. In this method, three consecutive frames are compared and detections are removed if they appear only once. Normally, this technique requires finding the median value of all individual pixels. To accelerate, we take the mean of the binarized images and apply simple thresholding. We then check which objects have not disappeared, and keep them in the original labeled image while discarding the others.

Finally, we enlarge all detection rectangles so that they contain the complete object. To do so, we apply a filter whose impulse response is a rectangle window of size propotional to video resolution. 11-by-11 window has been found to be sufficient for a 1080p video.

The impacts of tracking and classification on the detection stage are described in the corresponding subsections.

D. Tracking

Tracking enables the framework to match previous detections with the current detections, and is essential for improving performance and speed of the detection and classification processes.

The tracking step is initiated with the calculation of the center point and the size of each and every detection. These detections are then matched with existing trajectories with respect to the distances between their center points. While doing so, trajectories that cannot be matched to latest detections are removed, since they correspond to either noisy detections or objects that have stopped for a long time. After matching is complete, trajectory of each moving object is independently controlled: If an object is changing its direction too frequently, its trajectory is removed as it is noise with an adequately high probability. Since it is more important not to miss moving objects rather than completely eliminating noise, we set a threshold that is only high enough to eliminate radically variant noise sources: trajectory of an object is removed if that object's direction changes at least once in three consecutive frames on average. Upon the exclusion of abnormal detections, Kalman filter is employed to track the moving objects. Kalman filter is an especially useful tracking tool when objects disappear or stop temporarily. When an existing moving object is not detected, our framework keeps track of it for another 3 seconds while labelling it as passive; and if it does not appear again, the track is removed. For the filter, the process and measurement noise covariance matrices are initialized empirically, and are stored separately for each object. These matrices are then updated with respect to the percentage of frames in which the object has remained passive, so that the Kalman gain is low for mostly passive objects. The transition and measurement matrices are formed by kinetic calculations using the natural physical model. Further details on object tracking using Kalman filter can be found in [1].

E. Classification

The classification step is a member of the core framework processes. The detections that are carried onto this step are classified as vehicles or humans. This step consists of two substeps: *Preclassification* and deep convolutional neural network (deep CNN) classification.

During preclassification, positive and negative biases are prepared and enforced on the output of the deep CNN. It can be seen in Fig. 2 that the vehicles and humans are highly separable with respect to their sizes. To utilize this fact, an adaptive threshold is applied on detection size; the threshold level depends on the camera altitude and the video resolution. If a detection's size is larger than the vehicle threshold, then its probability of being a vehicle is increased by 20% (being a human is decreased by the same amount). Similarly, if it is smaller than the human threshold, then its probability of being a human is increased by 20%. With the same reasoning, a velocity threshold is also performed: Detections that move faster than 10 m/s are biased to be vehicles and the ones slower than 5 m/s are biased to be humans.

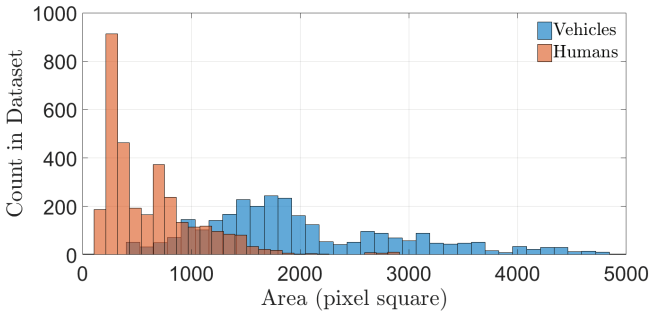

Fig. 2: The histogram for area size of human- and vehicle-class detections from a dataset collected by a UAV at an altitude of 100 meters. The video resolution is 1080p.

Deep CNN classification technique was chosen due to its favorable performance in image classification tasks [12]. As the network is deep and the number of moving objects may be large, an accelerated framework was needed. To address this issue, we employed pretrained SqueezeNet model [13], which achieves AlexNet-level [12] accuracy with much fewer parameters. SqueezeNet also enables us to benefit from transfer learning concepts, where partial training of the network eliminates the need for massive datasets.

We prepared a dataset using data from three sources: Videos taken from 100 meters altitude with the camera of Phantom 3 Advanced Drone (DJI, China) under different illumination and weather conditions, videos taken from 25 meters altitude with Hero4 camera (GoPro, CA) which are then downsampled by 4 in both directions, and videos from Stanford Drone Dataset [14]. Sample data are shown in Fig. 3. The entire set consists of 32760 training and 3640 validation images, equally divided into four classes: humans, groups of humans, vehicles and faulty detections, where the class of faulty detections is aimed at improving the detection performance. Humans and groups of humans are defined as two distinct classes in order to improve classification accuracy, although both are reported as humans at the end.

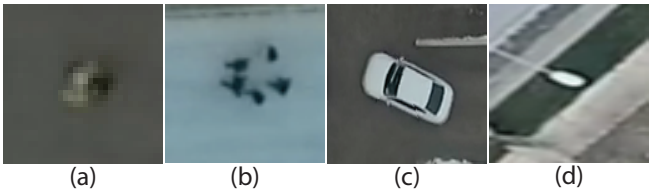

Fig. 3: One data sample from each class is shown: **(a)** human, **(b)** group of humans, **(c)** vehicle, **(d)** faulty detection. Images are resized for visual quality.

Using the dataset, two convolutional and one fully-connected layers of SqueezeNet were trained, and the resulting network was used for moving object classification. Images that are ready for classification are first resized to 227-by-227 pixels to match the network. Resized images pass through the network, and the bias values obtained from preclassification are added onto the resulting probabilities. If the image is classified as a faulty detection, the detection is removed. Otherwise, the image is labeled and the corresponding probability is used to determine when the same object will be classified

again, i.e. if the classifier gives a high probability of belonging to a class, then the object does not need to be classified for some time depending on the probability (up to 1.5 seconds).

F. Location Calculation

The last block of our proposed framework is the calculation of detection locations in universal measures. For this, the GPS of the UAV is utilized. If this information is not available, this step can be skipped, as it does not affect the other blocks.

To calculate the GPS coordinates of a moving object, its distance from the point where the camera is perpendicular to the ground is computed in North East Down (NED) coordinate frame. We use this distance along with the altitude of the camera from the ground, its azimuth and field of view in both north and east directions to calculate the relative coordinates of the object in NED frame. These coordinates are then converted into GPS frame by first calculating the Earth-Centered Earth-Fixed (ECEF) coordinates of the UAV using its GPS position and elevation from the sea level. Further details on these calculations can be found in [15].

III. Experiments

A. Implementation Details

A Phantom 3 Advanced UAV (DJI, China) was used throughout this implementation. The video quality supplied by the UAV was 1080p at 60 frames per second (fps). Videos were recorded from an altitude of 100 meters and camera exposure was manually adjusted depending on the brightness of the environment.

A number of frames proportional to the total processing time were skipped to achieve real-time results: Any two consecutive frames that are processed by the framework would correspond to the samples of the source video at every time interval where the framework returns to the image undistortion step. The framework provided results at an average of 6 fps (see Fig. 4).

The main components of the proposed framework were implemented using the C++ language, supported with OpenCV libraries [16] and the Caffe framework [17]. The classifier utilized the SqueezeNet CNN [13], partially retrained with our dataset, implemented with CUDA (NVIDIA, CA) to run on the laptop graphical processing unit (GPU). A modified version of the DJI Mobile SDK Android application, aided with a custom TCP client and server, and a Java™ application that uses Google Maps API (Google, CA) to retrieve elevation above sea level from the given GPS coordinates, were employed to access the UAV's telemetry information.

The experiments were performed on CASPER CN TKI-3210E laptop computer with a 2.5 GHz Intel Core™ i5-3210M CPU, 16 GB DDR3 RAM. And, 2 GB NVIDIA GeForce GTX 950M Graphic Card was used for the classification step.

B. Experiments

The core elements of the system that affect performance are image registration, moving object detection, Kalman filter tracking and classification. To quantify the performance of the set of algorithms used during each core element, we performed several experiments that are described below.

Image registration performance was quantified by measuring the peak-signal-to-noise ratio (PSNR) between the 800-by-600 pixels central regions of two consecutive frames that are aligned. For this, our UAV was flown over a mixed terrain area consisting of a few buildings, roads, hills, and trees. The acquired video consisted of two parts; in the first part the UAV was stationary to reveal the image registration performance under random instabilities, and in the second part the UAV was flown over the same area in a circular motion at a speed ranging between 1 m/s to 5 m/s and less than 0.5 rad/s rotation.

Moving object detection was tested with a video where the UAV was flown at an average 2 m/s for duration of 25 s, across a densely populated region with over 100 persons and vehicles in total, with an average of 10 moving objects visible at a frame. All detections were examined and labeled as true or false negatives or positives. In order to obtain a rough reference to human performance, the same videos were later shown to 4 volunteer human subjects on a 17-inch screen. The subjects were asked to identify all moving objects that they were able to spot. Calculating the precision and recall values as given in Eqs. 1 and 2, F_1 scores (see Eq. 3) have been used to quantify the performance of both the detection process and the reference human test subjects.

$$Precision = \frac{\textit{\# of all true positives}}{\textit{\# of all positives}}, \tag{1}$$

$$Recall = \frac{\textit{\# of all true positives}}{\textit{\# of all moving objects}}, \tag{2}$$

$$F_1 = \frac{2 \times \textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \tag{3}$$

The video used for detection testing was also used to quantify the Kalman filter performance. Labels of the detections were surveyed; tracking losses and joined labels were identified. These were used to quantify the number of times the tracker was successful, failed to track the target after initializing the tracks, or failed to produce persistent track.

Classification performance was tested using offline results from a collection of four videos acquired at a variety of terrain conditions and population densities. The actual type of the objects were later determined and compared to the classifier results. A total of 287 humans and 128 vehicles were present in the videos.

IV. Results

A. Computation Time

The computation time of each system block with their corresponding algorithms was measured, and reported in Fig. 4.

B. Performance

In the registration stage, the system achieves a minimum PSNR of 27 dB while UAV is stationary, and 17 dB while it is moving with an approximate speed of 5 m/s. As it is shown in Fig. 5, PSNR increases when the UAV slows down. UAV speed and rotation rate are inversely correlated with image

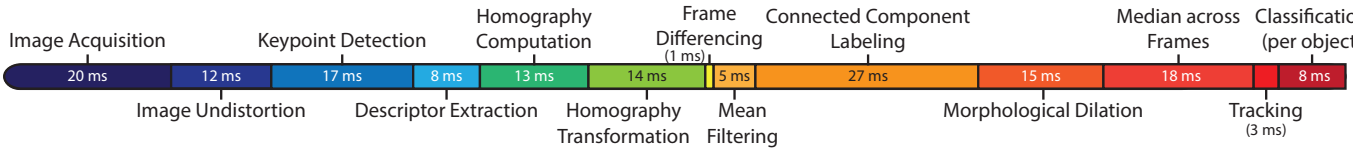

Fig. 4: Computation times, averaged over 100 video frames, are shown for each step of the framework. The times required for the completion of morphological dilation, tracking and classification steps depend on the number of moving objects in the frame. Classification time is reported for a single object, whereas dilation and tracking times have been averaged for a general case where more than 10 moving objects are present. The total time required for the processing of a frame is 150 ms to 200 ms depending on the number of objects (from 0 to 5), which corresponds to a video processing rate of 5 to 6.67 frames per second.

registration performance, which is an expected result since the transformation becomes more complex with faster movements.

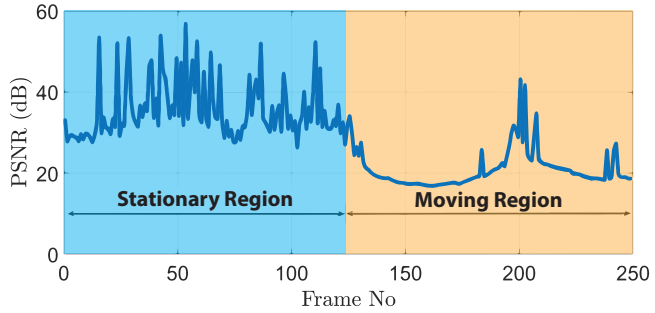

Fig. 5: Image registration process was assessed using peak signal-to-noise ratio over a 250-frame video. Nearly-perfect alignment was achieved when the camera is stationary. When it is moving, a minimum PSNR of 17 dB was observed.

The detection stage, without any feedback from tracking and classification steps, yields an F1 score of 0.91 (see Table I). This is a desirably high rate since the later phases of tracking and classification can compensate for misdetections. Tracking would likely filter out temporary misdetections, and persistent ones will be carried onto the classification stage, likely with the label of faulty detections. The comparison with subjects shows that although humans are quite precise while detecting movements, they miss several moving objects as they cannot focus on the entire video at once. Thus, the average human reference indicates that the proposed system meets its goal of exceeding the success of human operators.

TABLE I: Precision, recall and F_1 scores for moving object detection with no feedback from tracking and classification steps

	Precision	**Recall**	**F_1 Score**
Detection Algorithm	0.88	0.94	0.91
Human Average	1	0.71	0.83

Kalman tracking maintenance failure ratio is 8.77%, which means that our system can keep track of an object 91.23 frames out of 100 frames on average.

Classification performance is evaluated through a confusion matrix in Table II. The accuracy of the faulty detections class indicates classification step, in fact, improves the detection. For the combined outcomes of three classes, overall accuracy is 90.6%.

Lastly, we measured the real and calculated distances between the center points of frames and detected objects. We have seen that the difference is about 1%.

TABLE II: Confusion matrix of the classifier

		Prediction		
		Human	Vehicle	Faulty Detection
Actual	Human	**260**	6	21
	Vehicle	13	**111**	4
	Faulty Detection	6	3	**139**

V. Discussion

Quantitative analyses of our image registration, moving object detection and tracking performance have yielded greater average stationary state registration PSNR values and detection F1 scores than those from [18]. The overall classification accuracy was greater than in [2] and [3]. Considering major application differences, such as our use of feature based methods aided with different descriptors for image registration, our real-time approach using a sampled video with lower fps, use of different datasets and learning techniques, and possible differences in our camera setting such as UAV related hardware, it is difficult to evaluate the significance of these results. However, achieving higher than literature values can be a strong indicator that our framework provides successful and computationally cheap solutions for use in aerial surveillance related applications.

To enhance the proposed framework, some critical points can be improved. Image registration is a primary and critical process in this framework due to the use of a UAV that is in motion. In cases where there is an insufficient number of corner points in a scene, consecutive frames may be matched improperly. For such cases, area based image matching methods [19] can be applied. Also, more sophisticated methods, such as SIFT [20] or SURF [21], can be employed given the availability of resources with a high computational capacity.

To further reduce the number of detection failures, thresholding operation can be performed adaptively at the price of computation time. One simple approach might be selecting the threshold value with an adaptive model, such as Otsu's method [22], around the regions where an object has already been spotted. A more demanding approach can be determining the threshold after modeling the background in sliding patches across the images.

Parallax errors in the moving object detection is another issue that can be reduced with further studies. High objects, such as streetlights (see Fig. 3d), are detected as moving objects because of the relative motion with respect to the ground from the perspective of UAV. Our approach attempts to

eliminate such misdetections in classification step. In addition, as [23] suggests, the use of image gradients might improve the performance by achieving better parallax suppression. Besides, optical flow [24] might be incorporated into the moving object detection block at the price of computational power.

Classifier performance can be increased in various ways. Firstly, larger datasets in which large shadows are regarded as different classes can be used for training. Secondly, our preclassification method can be integrated into the deep CNN. For example, the output layer of deep CNN can be replaced with a support vector machine which takes velocity and size information of objects as additional inputs, which would yield a better learning of these features. Moreover, for the robustness of algorithm speed, multiple classifiers with different complexities can be trained. Simpler alternatives can be preferred to speed up, when the entire process slows down due to high number of objects; whereas sophisticated classifiers are used in the abundance of process time. Lastly, we employed SqueezeNet [13] for the framework to be fast, even on low-cost graphical processing units (GPU), whereas more complex CNNs can increase classification performance in the cost of computation power.

In this work, UAV has been assumed to be controlled with an external controller, which is generally the case for commercially available UAVs. An alternative implementation can unify the processing with the control of UAV. Furthermore, onboard implementations that work on UAV itself can eliminate the need of high communication capabilities between UAV and ground stations, so higher quality videos might be available for processing. In this work, classification was performed on a low-cost graphics processing unit (GPU). Other blocks of the framework can also be transferred to GPU for even higher speeds.

VI. Conclusion

In this study, we proposed a fast and robust framework that detects and tracks moving objects from mobile UAV videos, and classifies the objects as vehicles and humans in real-time, for low-altitude applications. An additional feature where the GPS coordinates of the classified objects are calculated by using telemetry data of the UAV has been included in the framework. The improvements presented in this work is mostly practical. Different from previous studies, our framework can perform multiple tasks in real-time. Novel ideas to increase the robustness and the speed of the system have been described throughout the paper. We believe our speed optimizations, detection system with feedback from tracking and classification steps, and SqueezeNet-based [13] fast classifier with a specific preclassification approach will stimulate further research and practical work.

Acknowledgments

We thank Ersin Yar, Bumin Kaan Aydın, Dr. Cem Tekin and Dr. Orhan Arıkan for comments and fruitful discussions throughout the project. This work was supported by HAVELSAN Inc. and Bilkent University, Electrical and Electronics Engineering Department. We acknowledge funding by TÜBİTAK BİDEB 2209-B grant.

References

[1] G. Máttyus, C. Benedek, and T. Szirányi, "Multi target tracking on aerial videos," 2010.

[2] Y. Iwashita, M. S. Ryoo, T. J. Fuchs, and C. Padgett, "Recognizing humans in motion: Trajectory-based aerial video analysis." in *BMVC*, vol. 1, no. 3, 2013, p. 6.

[3] O. Oreifej, R. Mehran, and M. Shah, "Human identity recognition in aerial images," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 709–716.

[4] Y. Dedeoğlu, "Moving object detection, tracking and classification for smart video surveillance," Ph.D. dissertation, Bilkent University, 2004.

[5] B. Möller and S. Posch, "An integrated analysis concept for errors in image registration," *Pattern Recognition and Image Analysis*, vol. 18, no. 2, pp. 201–206, 2008.

[6] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[7] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Computer Vision–ECCV 2006*, pp. 430–443, 2006.

[8] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1, pp. 43–72, 2005.

[9] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*. Ieee, 2012, pp. 510–517.

[10] N. Singla, "Motion detection based on frame difference method," *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1559–1565, 2014.

[11] C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo, "An improved moving object detection algorithm based on frame difference and edge detection," in *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*. IEEE, 2007, pp. 519–523.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[14] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes." in *ECCV (8)*, 2016, pp. 549–565.

[15] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems*. Springer Science & Business Media, 2011.

[16] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[18] A. Walha, A. Wali, and A. M. Alimi, "Moving object detection system in aerial video surveillance," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2013, pp. 310–320.

[19] J. Joglekar and S. S. Gedam, "Area based image matching methods—a survey," *Int. J. Emerg. Technol. Adv. Eng*, vol. 2, no. 1, pp. 130–136, 2012.

[20] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[23] V. Reilly and M. Shah, "Detecting, tracking, and recognizing activities in aerial video," 2012.

[24] K. Kale, S. Pawar, and P. Dhulekar, "Moving object tracking using optical flow and motion vector estimation," in *Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on*. IEEE, 2015, pp. 1–6.