2018 1st International Conference on Data Intelligence and Security

Hand Gesture Controlled Drones: An Open Source Library

Kathiravan Natarajan†
Student IEEE Member
Department of Computer Science
Texas A&M University – Commerce
Commerce, Texas 75429
sunkathirav@gmail.com

Truong-Huy D. Nguyen*
IEEE Member
Department of Computer and Information Sciences
Fordham University
Bronx, New York 10458
tnguyen88@fordham.edu

Mutlu Mete
IEEE Member
Department of Computer Science
Texas A&M University – Commerce
Commerce, Texas 75429
mutlu.mete@tamuc.edu

***Abstract*—Drones are conventionally controlled using joysticks, remote controllers, mobile applications, and embedded computers. A few significant issues with these approaches are that drone control is limited by the range of electromagnetic radiation and susceptible to interference noise. In this study we propose the use of hand gestures as a method to control drones. We investigate the use of computer vision methods to develop an intuitive way of agent-less communication between a drone and its operator. Computer vision-based methods rely on the ability of a drones camera to capture surrounding images and use pattern recognition to translate images to meaningful and/or actionable information. The proposed framework involves a few key parts toward an ultimate action to be taken. They are: image segregation from the video streams of front camera, creating a robust and reliable image recognition based on segregated images, and finally conversion of classified gestures into actionable drone movement, such as takeoff, landing, hovering and so forth. A set of five gestures are studied in this work. Haar feature-based AdaBoost classifier [1] is employed for gesture recognition. We also envisage safety of the operator and drone's action calculating the distance based on computer vision for this task. A series of experiments are conducted to measure gesture recognition accuracies considering the major scene variabilities, illumination, background, and distance. Classification accuracies show that well-lit, clear background, and within 3 ft gestures are recognized correctly over 90%. Limitations of current framework and feasible solutions for better gesture recognition are discussed, too. The software library we developed,and hand gesture datasets are open-sourced at project website.**

1. Introduction

Drones, also known as unmanned aerial vehicles, are on the rise in recreational and in a wide range of industrial applications, such as security, defense, agriculture, energy, insurance and hydrology. Drones are essentially special flying robots that perform functionalities like capturing images, recording videos and sensing multimodal data from its environment. There are two types of drones based on their shape and size, fixed-wing and multirotor. Because of their versatility and small size, multirotor drones can operate where humans cannot, collect multimodal data, and intervene in occasions. Moreover, with the use of a guard hull, multirotor drones are very sturdy in collisions, which make them even more valuable for exploring uncharted areas. At present, flying robots are used in different businesses like parcel delivery systems [2]. For example, companies like Amazon Prime and UPS are using multirotor drones to deliver their parcels. New York Police Department uses quadcopters in crime prevention [3]. For the purposes of agriculture monitoring, for instance, the use of multiple sensors such as video and thermal infrared cameras are of benefit [4]. Drones are especially useful in risky missions. For the sake of clarity in the rest of this work, we define *a drone* as a multirotor flying robot, excluding fixed-wings.

A visual camera is an indispensable sensor for current drones. The low cost, low power, small size of image capturing, and streaming devices make them a de facto feature for numerous drones in the market. Output of a drone's camera can be used in many ways depending of the applications. In a common scenario, the camera output is directed to the drone operator who may command the drone a new instruction based on the current visual environment via a remote controller, which serves as an agent between drone and its operator. In this work, we investigate an alternative method of controlling multirotor drones using hand gestures as the main channel of communication. We propose a framework that maps segregated images from video stream as one of five commands/gestures. The camera can capture visual instructors from the operator, which eliminates the control device, leading the way for agent-less communication.

Haar features serve as fundamental masks to capture gradient changes in images. Each block of mask can be scaled or rotated to capture predetermined targets. These advantages allow us to detect various gestures in many sizes. Therefore, a Haar feature-based AdaBoost classifier [1] is employed in action planner. Safety issues are also considered

†*Supported by Texas A&M University–Commerce Graduate School and Department of Computer Science*
∗*Corresponding author*

978-1-5386-5762-1/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDIS.2018.00035

while the drones automatically comply with the commands initiated by operator's gestures. This project also presents a case study for image recognition-driven autonomous drones.

Our key contributions in this project include

1) A novel framework of drone control based on hand gestures
2) A comparison of state-of-the-art computer vision approaches in hand gesture detection, applied on our hand gesture dataset
3) A discussion of key challenges and lessons learned from building the framework's hand gesture recognition component.

This project uses one of many mediocre drones in the market: Parrot AR.Drone 2.0 [5]. Both the software library and hand gesture datasets are open-sourced at [6].

2. Background

Before detailing our framework, we briefly summarize related works in drone control approaches and attempts in employing gesture detection for this purpose.

2.1. Drone Control

Most commercial drones available on the market come with specially designed controllers, either as a dedicated signal transmitter or software applications running on users' hand-held device (such as mobile phones or tablets). In both cases, the controller sends commands with detailed movement information such as *move the drone x units towards a certain direction* through wireless channels (e.g., Wi-Fi or Bluetooth). Notable products include the DJI drones (models Phantom, Inspire, Matrice, etc.) [7] and Parrot's drones (models AR. Drone, Bebop, DISCO, Swing, Mambo, etc.) [8].

Recently there have been commercial products that introduce hand gestures as a viable control mechanism. To capture the gestures, there are two approaches.

- Using specially designed gloves: The controller is mounted on a glove worn by users and detects in real time the yaw, pitch, and roll of the hand to translate into respective movements for the drone. Products include the Kd Interactive Aura Drone [9], and the MenKind Motion Control Drone [10].
- Using computer vision via the on-board camera. These devices use the on-board camera to detect in real time where the user's hand is and respond to it in intuitive ways. Products include the DJI Spark Drone [11].

The first approach above presents an attempt to add new control dimensions, thus allowing more degrees of freedom to the drone controller. Instead of pressing some predefined buttons, users can move their fingers or wave their hand(s) in specific ways that are recognized by sensors installed in the glove, which are then converted into digital commands. The transmission of commands is done over radio channels, so it is the same as the traditional control paradigm. The second approach on the other hand takes a more radical leap by employing real-time image analysis, which is done on the drone itself, to recognize commands instead of sending them over radio channels.

In academia, there have been similar attempts to investigate alternative methods to control drones using body parts, such as hand gestures or full body motions. Notably, Cauchard et al. [12] found that when interacting with drones using body language, drone operators feel natural using gestures like those used as with a pet or other people, such as beckoning or waving. As such, natural user interfaces (NUIs) present an appealing way to enhance the user experience when interacting with drones, as compared to the traditional way of a remote-control device. In building an NUI for drone control, there are two main directions fellow researchers are working towards: with and without the help of aiding devices.

The first involves the use of some third-party device that can recognize non-verbal gestures reliably, before mapping the detected gestures into suitable digital commands. Some such devices include the Leap Motion Controller[1] [13], [14] and the Microsoft Kinect [15], [16].

While Leap Motion Controller is designed specifically to capture hand motions, the Kinect can capture full body motion faithfully. While this approach yields high accuracy in gesture or body motion detection, they need to be connected to a computer to work, so portability is a limiting factor.

In the second direction, body movement is detected in real-time, using machine vision, to control the drone without any additional instrument. Researchers have examined the use of eye gazes [17], face poses, hand gestures, and the combination of them [18], [19].

2.2. Hand Gesture Studies

Image-based hand gesture recognition problems have been studied extensively for decades. Twenty-four basic signs of American Sign Language are detected and classified using a boosted cascade of classifiers trained using AdaBoost and informative Haar wavelet features. In this work, Dinh et al. [20] have proposed a new feature called *Double L* for best describing the hand gestures other than edge features, line features and edge surrounded features. Real time hand gesture detection based on bag of features and support vector machines were proposed in [21]. In training, scale invariance feature transform (SIFT) is used to extract the key-points for all training images, and vector quantization is used to map key-points from training image into bag of words after performing K-means clustering. These histograms act as feature vectors. SVM model is trained for the classification purposes. Experiments were carried out with a web camera.

1. https://www.leapmotion.com/

A study done by Dardas et al. [22] detects and tracks hand gestures in cluttered backgrounds as well as in different lighting conditions. It uses skin detection and hand posture contours comparison algorithms by subtracting faces and only recognizes hand gestures using Principal Component Analysis. In each training stage, different hand gesture images with various scales, angles, and lightings are trained. The training weights are calculated by projecting training images onto the eigen vectors. During testing, the images that contained hand gestures are projected onto the eigen vectors and the testing weights are calculated. Finally, Euclidean distances are calculated between training weights and test weights to classify hand gestures.

In another work, Hu moment features used by Meng et al. [23] proposed an algorithm for detecting the fingertip structure. First, the features which are the areas including skin region and the image, were made to differentiate the background in space of saturation, value of brightness, and hue from the skin region. Later, an algorithm to find the region of interest was implemented and fingertips were detected by approximating the contour. The seven-dimensional feature vector was created after the detection process. Finally, the distance marching criterion was used for the hand gesture recognition. This algorithm improved the accuracy by 2.7% when compared to Hu moment feature recognition.

Detecting hand gestures in real time is a challenging task due to a few reasons, including how people perform hand gestures. Molchanov et al. [24] recently addressed these challenges by a three dimensional recurrent convolutional neural network model with multi-modal input streams. The hypothesis is validated by testing multi-modal dynamic hand gesture dataset captured with depth, color and stereo infrared sensors. This system achieved an accuracy of 83.8% in the complex dynamic hand gesture set.

A multi-class classification approach based on Weighted Linear Discriminant Analysis and Gentle AdaBoost (GAB) algorithm was proposed by Tian et al. [25]. In this approach, Histogram of Oriented Gradient (HoG) features are extracted arbitrarily in random locations and a multi-class cascade classifier is trained for hand gesture detection.

3. Proposed Framework

In this section, we detail our framework of gesture-based drone control. The targeted drone types for this framework are multirotor helicopters equipped with a front-facing camera, such as the Parrot AR.Drone [5]. Figure 1 depicts one such drone with four rotors on the sides of the body in charge of lifting the drone off the ground and moving the drone in different directions. A camera is mounted at the front of the drone's body, which allows recording of the environment within its field of view. The framework is depicted in Figure 2.

The video stream is constantly recorded through the on-board camera of the drone, and then segmented into sequences of still images. Each image is then analyzed through the hand gesture recognition process, which includes three main steps: feature extraction, hand region identification, and finally gesture classification. A command mapper transforms the detected gesture into a command, such as *take off*, *land*, or *back off*. An action planner takes the command as its input and compute the corresponding course of primitive actions to satisfy the command. While the planner is operating, it also considers the surrounding environment to avoid collision and ensure the safety for both the drone and perceived obstacles.

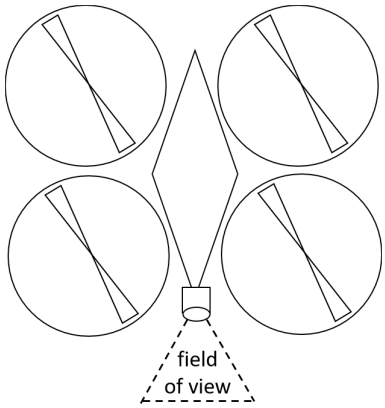

Figure 1. Stylized top-down view of a quadrotor drone, facing downwards with a camera mounted at the front of the drone.

The hand gestures we work on are shown in Figure 3. Note that gestures are recognized based on certain orientation of the user's hand, i.e., either right or left hand is used for each gesture. The set of all five gestures includes fist, palm, go symbol, v-shape, and little finger. These gestures are arbitrarily picked but we made sure to have a lot of unique haar features for each carefully chosen gesture and they are very common gestures in the society and easy to pose. The reason for using only 5 gestures is to provide all basic functionalities of the drone like moving the drone right, left, backward, forward and clicking pictures. Unquestionably, more functionalities can be implemented by training more hand gestures. But the scope of this paper is focused on achieving high accuracies in mediocre drones for those basic functionalities mentioned above.

We avoid three fingers and two fingers gestures, since they may be translated into similar Haar features, which may lead to many errors in classification step. Another example, the one finger gesture and the fingers crossed gesture may end up having similar Haar features. During the preliminary set of experiments, we decided to choose aforementioned hand gestures, assuming that they will have a new set of differentiated features for each gesture to be classified correctly. For example, the go symbol is expected to possess a separate set of Haar features when compared to the fist or the palm which in turn reduces the number of misclassified images and improves the accuracy. This does not mean Haar features end up with similar values if the gestures look alike. Therefore, we attempted to reduce one such possibility of misclassification by choosing significantly different-posing gestures. In the rest of this study, the go symbol, v-shape, and little finger gestures are indicated by GS, VS, and LF,

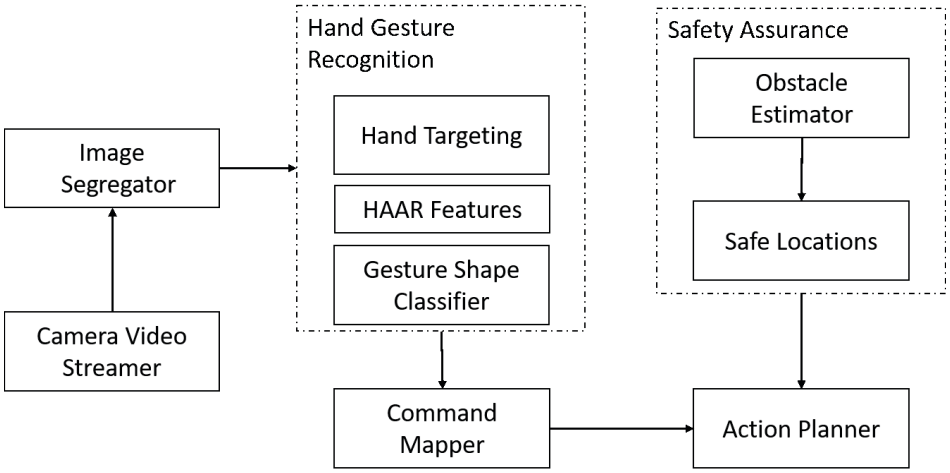

Figure 2. Gesture-based drone control framework.

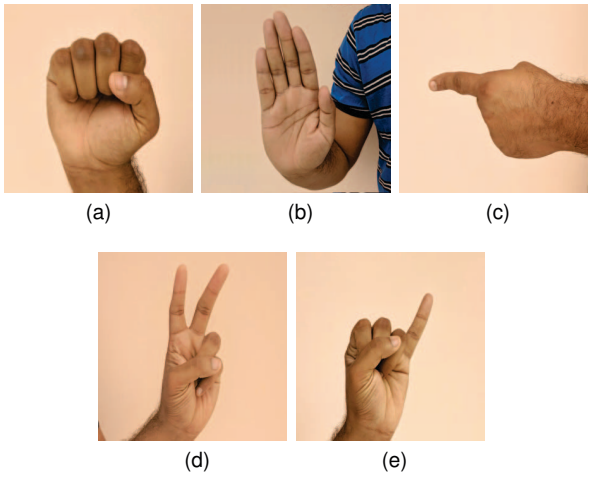

Figure 3. The hand gestures to be classified in this study: fist with right hand (3a), right hand palm (3b), left pointing left hand (go symbol)(3c), left hand V-shape (3d), and left hand little finger (3e).

respectively.

In order to implement the complete framework, there are a number of key challenges that we need to address, namely gesture recognition, visual variability of scene, and safety assurance of maneuver.

3.1. Gesture Recognition

In two related studies Viola and Jones [1], [26] introduced Haar feature-based cascade AdaBoost classifier exclusively for frontal face recognition. Their method builds a weak classifier using extracted Haar features compiled from various sub-windows/patch of the target image. AdaBoost (Adaptive Boosting) is a weak learning algorithm and was introduced in [27]. It classifies a feature vector exploiting many other subsequent learners. AdaBoost updates weights of each weak classifier at the end of each iteration in training. AdaBoost-based solutions require a set of real classifiers that learn from training dataset and map testing data to one of the classification labels.

We used Haar features to represent each image of dataset. Although Haar features was introduced in 1910 [28], it is not popularized for image recognition problems until a broad analysis by Papageorgiou et al. [29]. A Haar-based feature utilizes rectangular regions at various locations of the detection window by summing up the pixel intensities in each location of the detected window and calculates the difference between these summation values. These differences are then used to categorize the image. In our scenario, the feature extraction module uses the pattern generated by many local Haar features of a hand gesture. Later, the classifier maps feature vector of gestures either one of the existing gesture labels or as void. The reasons for choosing Haar classifiers over other algorithms are that Haar cascade has better detection rate than other feature descriptors like Hog [30] in less clear images and moreover, its implementation is simple, achieves more accuracy with less training images, and consumes less memory unlike GPU-enabled image classification system like Convolutional Neural networks [31].

3.2. Visual Variability of Scene

The proposed study is designed for a user to control a drone in daily life, not a special laboratory environment. For this very reason, we want to empirically measure the effects of scene variability while classification framework is kept unchanged. To this end, three different visual variables are introduced to be tested: illumination, background, and distance of target gesture. The *illumination* measures how well the scene is lit. In terms of illumination variable, a scene (experimental environment) is categorized in a binary way, dim lit or well lit. We did not use any special lighting tools while collecting images of the dataset. Instead, various

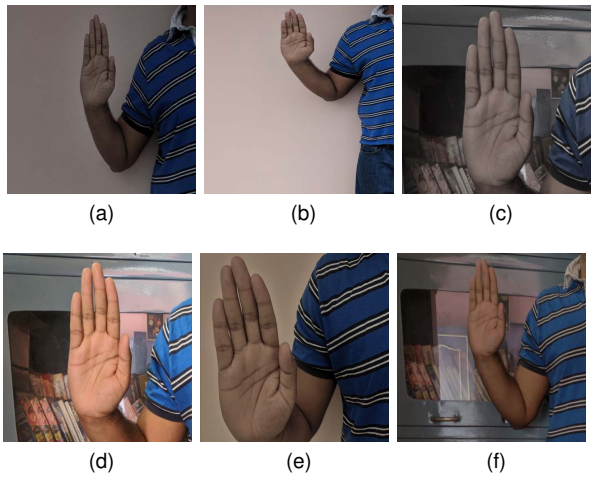

Figure 4. Demonstrating scene variability of the palm gesture. 4a: dim lit, clear background, more than 3 ft away, 4b: well lit, clear background, more than 3 ft away, 4c: dim lit, cluttered background, within 3 ft, 4d: well lit, cluttered background, within 3 ft, 4e: dim lit, clear background, within 3 ft, 4f: dim lit, cluttered background, more than 3 ft away.

test cases are captured under sun and/or everyday fluorescent lights. The variability of *background* is expressed with one of two categories, cluttered or clear (almost blank). A user in front of a loaded bookshelf, a natural scene, or many other objects are categorized as clutter scene, whereas, a gesture posed in front of walls or doors are considered as clear background. Cluttered background problems were detailed in [32]. The last scene condition is basically the *distance* between gesture posing hand of the user and drone's camera. The distance threshold is 3 ft. It means that while some gestures are presented within 3 ft, the others are tested more than 3 ft away. Figure 4 shows various scenes based on newly introduced conditions.

3.3. Safety

After a gesture is recognized and converted to a command, such as *move to the left*, the action planner on the drone kicks in to compute the most appropriate course of action that satisfies the recent command. In this process, it is imperative for the drone to carry out the action while ensuring safety to itself, surrounding objects, and the environment. Collision to any of these entities potentially causes serious damage to the parties involved, which is highly undesirable. In our framework, action planning module requires the drone to utilize its sensors (e.g., camera and proximity) to estimate the area where it can safely fly or hover to.

Collision avoidance is a topic addressed in robotics. Drones are much more susceptible to external factors that cause their movements to be unstable, such as wind or air flows. Collision avoidance in drones requires additional considerations for such factors. While some approaches rely on the on-board camera for this task [33], [34], [35], others propose the use of more advanced sensors, such as ultrasonic or laser range finders [36], [37]. One limitation of camera-based solutions is that they may perform poorly when there are optical noises, such as in low lighting or foggy environments. Using more non-vision based sensors helps alleviate this problem, but adds more load to the overall weight of the drone, which may not always be feasible.

TABLE 1. NUMBER OF IMAGES IN THE STUDY DATASET

Hand Gestures	Positives	Negatives
Fist	1570	900
Palm	1456	900
GS	1390	900
VS	1530	900
LF	1456	900

4. Experiments

Parrot AR.Drone 2.0 [5] is used throughout all the experiments. It is one of the early versions of the Wi-Fi controller drone, which is debuted by Parrot SA (Paris, France) in 2012. It costs around $130 as of December 2017. AR.Drone 2.0 is equipped with 720 x 720 pixels camera, ARM Cortex A8 1 GHz 32-bit processor, Wi-Fi connectivity, gyroscope, accelerometer, magnetometer, pressure sensor, and altitude ultrasound sensor. A stylized top-down view of the AR.Drone 2.0 is shown in Figure 1.

Gesture recognition experiments are carried out with a 2.60 GHz CPU, 16 GB memory Ubuntu 14.04.5 LTS (Trusty Tahr) operating system. Drone control software is developed using Python 2.7 with OpenCV 3.3.0, an open-source computer vision library [38].

Training images are collected at resolution of 720 x 720 pixels, which are same as the drone's front camera resolution. Positive training images are hand gestures images collected from drone's front camera. Meanwhile negative training images, also called background images or background image samples, are collected randomly with the help of image search engines, which do not contain any hand gesture images. Should the size of a negative image be greater than 720 x 720 pixels, it is down-sampled to size of positive images. The number of positive and negative training samples for each gesture is given in Table 1. A total of 8302 images are used in the experiments.

We benefit from OpenCv's embedded tool to mark bounding box and location of each gesture in positive training images. It should be noted that although all five gestures are posed with same user with same right/left hand, a same gesture appears at many different orientations and scales. In a preprocessing step, their location should be marked correctly to train a classifier. OpenCV also provides an integrated annotation tool to manually describe the objects to be detected by the classifier. We created an annotation file which contains a file structure to maintain association between positive images and the coordinates of the bounding rectangles of the gestures. Following this step, we extracted

features in OpenCV, which supports in creating vector representation of training images using Haar features. While generating feature vectors from the images, we specify the sample size as 20 x 20 pixels since Lienhart et al. [39] reported that 20 x 20 of sample size achieved the highest hit rate in a similar study. Upon extracting feature vectors, we train the boosted cascade of weak classifiers, AdaBoost, using all positive and negative feature vectors. Each of gesture classifiers are trained separately, which generates five different classification models. Once an image is streamed from the drone's camera to our software, each frame is mapped to the respective gesture or none. Training of each classifiers takes around 15 minutes because of the smaller window size (20 x 20) of Haar features extraction step. All training images and model files (in the form of .xml) are publicly available at project web site [6]. In the context of AdaBoost, each resulting .xml file serves as strong classifier, composed of the weighted sum of weak classifiers. The number of training stages for palm, fist, GS, VS, and LF gestures in Haar cascade classifier are reported as 4, 16, 8, 10, and 5, respectively.

5. Discussion of Experimental Results

Individual accuracy of each gesture is detailed in Table 2. This table also categorized how the classifier performs in variable scene conditions, which are described in Section 3.2. The accuracy measure reported in Table 2 is the ratio of the correctly classified gestures to the total number of same gesture. For example, in case of the palm gesture experiments with scene variables of DL, CTB, LT-3, 4593 of 5000 palm gestures are correctly identified.

The distance is observed as the most significant scene variable. The gestures posed within 3 ft outperform significantly the gestures posed more than 3 ft away. Referring to Table 2, regardless of illumination and background variability, the average accuracy of LT-3 experiments is 0.94 while that of MT-3 is 0.71. The decline of accuracy based on distance is found common amongst all gestures. One of a few sharp accuracy declines is seen in the classification of palm, where scene variable of distance is changed from LT-3 to MT-3. In this pairwise comparison, the accuracy drops from 0.97 to 0.70. The distance variable causes a comparatively mild diminishing of classification accuracy in the case of well-lit and clear background experiments, from 0.80 to 0.70.

Second and third significant scene variables are observed as background and illuminations, respectively. A cluttered background lessens accuracy in many pairwise comparisons, just as in within 3 ft, dim lit fist experiments (DL, CTB, LT-3: 0.89 while DL, CLT, LT-3: 0.91). Another example of similarly lessened accuracy is the gesture of go shape where various backgrounds of scenes are tested in well-lit and within 3 ft poses (WL, CTB, LT-3: 0.91 while WL, CLB, LT-3: 0.96).

Illumination, categorized as dim or well lit, is found the least significant scene variable. Expectedly, the effect of lighting condition is almost obvious amongst all gestures,

TABLE 2. CLASSIFICATION ACCURACIES FOR GESTURE DETECTION

Test Conditions	Palm	Fist	GS	VS	LF
DL, CTB, LT-3	0.92	0.89	0.86	0.84	0.86
DL, CTB, MT-3	0.66	0.70	0.60	0.65	0.69
DL, CLB, LT-3	0.97	0.91	0.87	0.90	0.88
DL, CLB, MT-3	0.70	0.74	0.69	0.65	0.59
WL, CTB, LT-3	0.90	0.89	0.91	0.86	0.81
WL, CTB, MT-3	0.69	0.81	0.73	0.66	0.70
WL, CLB, LT-3	**0.99**	**0.99**	**0.96**	**0.95**	**0.90**
WL, CLB, MT-3	0.84	0.81	0.80	0.80	0.76

DL: dim lit, WL: well lit, CTB: cluttered background, CLB: clear background, LT-3: within 3 ft, MT-3: more than 3 ft; GS: Go symbol, VS: V-shape, LF: little finger. The highest average accuracy settings are given in bold.

except in a few cases of little finger and fist. The accuracy of little finger is reduced from 0.86 to 0.81 in the case of DL, CTB, LT-3 vs WL, CTB, LT-3. In the same pairwise experiments of fist, changing the illumination variable from DL to WL does not help the accuracy increase (DL, CTB, LT-3: 0.89 while WL, CTB, LT-3: 0.89).

Overall best average classification accuracy is 0.95 and obtained with scene variables of WL, CLB, LT-3, as given at row #7 of Table 2. In summary, significance of scene variables is ordered as distance, background, and illumination, respectively.

A set of misclassified gestures is depicted in Figure 5. The go symbols of Figure 5a and Figure 5b are both classified as palm. The gesture of Figure 5c should have been recognized as little finger but our classifier incorrectly labels it as fist. Both Figure 5d and 5e are recognized as v-shape in tests. These mistakes are probably due to vertical edges in the background. As a last example of misclassification, Figure 5f is a v-shape gesture; however, it is recognized as fist.

The misclassified images give us a few insights into causes of the errors done in testing. First, the operator should be close enough to the drone for a better accuracy. This problem also involves the camera resolution of the drone and can be partially elevated with high resolution images or better cameras. We observed that Haar features are not immune to non-gesture related background patterns. This occurs because the proposed framework does not include the background removal procedure.

6. Conclusion

Our goal of this study is to enable the hand gesture-based control mechanism with maximum possible accuracy even in mediocre drones which can be easily outperformed by the state-of-art drones due to their inbuilt high camera resolutions like 4K, 8K, 16K and 64K etc. In this empirical study, we investigated more on software development for the AR Drones. We presented an image recognition-based communication framework to control drones with hand gestures. The framework is successfully tested using a mediocre

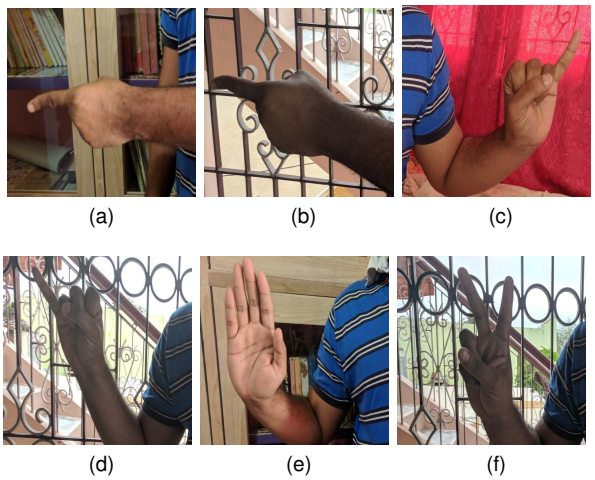

Figure 5. Misclassified gestures. See Section 5 for the further discussion.

drone, Parrot AR.Drone 2.0. A set of five gestures were carefully selected to build a dataset of 8302 images. Each image is represented by a set of Haar features in cascaded AdaBoost algorithm. Classification results showed that the distance between drone and its operator is the most important indicator of success. This applies all of five gestures. Experimental tests resulted in an average accuracy of 0.90 where operated posed gestures were within 3 ft, regardless of illumination and background variability of the scene. We found that the accuracy of the framework is highest once the operator poses within 3 ft, well lit, and clear background. This controlling distance can be further improved by utilizing better cameras such as those supporting 4K or 16K resolution in the drones, which allows capturing of images with good resolution at longer distance, or implementing the same framework on state-of-art drones with better imaging capabilities. With the available HD camera in mediocre drones, the hand gesture recognition in the distance between 3 and 5 ft is highly accurate, and this controlling distance can be improved by enabling high resolution cameras in drones. With the current hand gesture-based control mechanism, we envision that drones can be sent to any feasible distances and perform operations, before flying back to the controller for further close-ranged interactions. To explore the effects of different hand poses and deviation, an in-depth statistical analysis on the applicability of the framework in different environment settings is planned for future work.

Acknowledgments

The authors thank Texas A&M University–Commerce Graduate School and Department of Computer Science for the travel and publication support.

References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.

[2] V. Gatteschi, F. Lamberti, G. Paravati, A. Sanna, C. Demartini, A. Lisanti, and G. Venezia, "New frontiers of delivery services using drones: A prototype system exploiting a quadcopter for autonomous drug shipments," in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, vol. 2. IEEE, 2015, pp. 920–927.

[3] T. Moore, "Nypd considering using drones to fight crime - ny daily news," May 2014. [Online]. Available: http://www.nydailynews.com/new-york/nyc-crime/nypd-drones-fight-crime-article-1.1799980

[4] D. Turner, A. Lucieer, Z. Malenovskỳ, D. H. King, and S. A. Robinson, "Spatial co-registration of ultra-high resolution visible, multispectral and thermal images acquired with a micro-uav over antarctic moss beds," *Remote Sensing*, vol. 6, no. 5, pp. 4003–4024, 2014.

[5] Parrot, "Parrot ar.drone 2.0 power edition," Nov 2017. [Online]. Available: https://www.parrot.com/us/drones/parrot-ardrone-20-power-edition

[6] K. Natarajan, "Drone project web page," Dec 2017. [Online]. Available: https://github.com/KathiravanNatarajan/HandGestureControlledDrones

[7] DJI, "Dji, the future of possible," Dec 2017. [Online]. Available: https://www.dji.com/

[8] Parrot, "Parrot drones," Dec 2017. [Online]. Available: https://www.parrot.com

[9] Z. Stone, "Review: The Hand-Controlled Aura Drone Will Make You Jealous Of Your Kids," 2017. [Online]. Available: https://www.forbes.com/sites/zarastone/2017/11/20/review-the-hand-controlled-aura-drone-will-make-you-jealous-of-your-kids/

[10] S. Mughal, "Review: Motion Control Drone fly your drone the easy way," 2017. [Online]. Available: https://www.oxgadgets.com/2017/09/motion-control-drone.html

[11] J. Goldman, "DJI Spark review: Ups the ante on selfie drones," 2017. [Online]. Available: https://www.cnet.com/products/dji-spark/review/

[12] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "Drone & me: an exploration into natural human-drone interaction," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*. New York, New York, USA: ACM Press, 2015, pp. 361–365. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2750858.2805823

[13] B. Sanders, D. Vincenzi, and Y. Shen, "Investigation of Gesture Based UAV Control," in *Advances in Human Factors in Robots and Unmanned Systems*, Chen J., Ed. Springer, Cham, jul 2017, pp. 205–215. [Online]. Available: http://link.springer.com/10.1007/978-3-319-60384-1_20

[14] A. Sarkar, K. A. Patel, R. K. G. Ram, and G. K. Capoor, "Gesture control of drone using a motion controller," in *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*. IEEE, mar 2016, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7462401/

[15] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A Kinect-based natural interface for quadrotor control," *Entertainment Computing*, vol. 4, no. 3, pp. 179–186, aug 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952113000025

[16] K. Pfeil, S. L. Koh, and J. LaViola, "Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles," in *Proceedings of the 2013 international conference on Intelligent user interfaces - IUI '13*. New York, New York, USA: ACM Press, 2013, p. 257. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2449396.2449429

[17] J. P. Hansen, A. Alapetite, I. S. MacKenzie, and E. Møllenbach, "The use of gaze to control drones," in *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14*. New York, New York, USA: ACM Press, 2014, pp. 27–34. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2578153.2578156

[18] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella, "Human Control of UAVs using Face Pose Estimates and Hand Gestures," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction - HRI '14*. New York, New York, USA: ACM Press, 2014, pp. 252–253. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2559636.2559833

[19] V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, "HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 617–623. [Online]. Available: http://ieeexplore.ieee.org/document/6696415/

[20] T. B. Dinh, V. B. Dang, D. A. Duong, T. T. Nguyen, and D.-D. Le, "Hand gesture classification using boosted cascade of classifiers," in *Research, Innovation and Vision for the Future, 2006 International Conference on*. IEEE, 2006, pp. 139–144.

[21] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[22] N. H. Dardas and E. M. Petriu, "Hand gesture detection and recognition using principal component analysis," in *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[23] Guoqing-Meng and M. Wang, "Hand gesture recognition based on fingertip detection," in *2013 Fourth Global Congress on Intelligent Systems*, Dec 2013, pp. 107–111.

[24] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4207–4215.

[25] F. Tian, Q.-C. Hu, and T.-N. Zhang, "A hand gesture detection for multi-class cascade classifier based on gradient," in *Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2015 Fifth International Conference on*. IEEE, 2015, pp. 1364–1368.

[26] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[27] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[28] A. Haar, "Zur theorie der orthogonalen functionensysteme. inaugural," Ph.D. dissertation, Dissertation (Göttingen, 1909), 1–49. Math. Annal. 69, 331–271, 1910.

[29] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Computer vision, 1998. sixth international conference on*. IEEE, 1998, pp. 555–562.

[30] P. Negri, X. Clady, and L. Prevost, "Benchmarking haar and histograms of oriented gradients features applied to vehicle detection." in *ICINCO-RA (1)*, 2007, pp. 359–364.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] M. J. Bravo and H. Farid, "Object recognition in dense clutter," *Perception & psychophysics*, vol. 68, no. 6, pp. 911–918, 2006.

[33] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, may 2013, pp. 1750–1757. [Online]. Available: http://ieeexplore.ieee.org/document/6630807/

[34] H. Alvarez, L. M. Paz, J. Sturm, and D. Cremers, "Collision Avoidance for Quadrotors with a Monocular Camera." Springer, Cham, 2016, pp. 195–209. [Online]. Available: http://link.springer.com/10.1007/978-3-319-23778-7{_}14

[35] C. Fu, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, "Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 513–533, jan 2014. [Online]. Available: http://link.springer.com/10.1007/s10846-013-9918-3

[36] A. Moses, M. J. Rutherford, M. Kontitsis, and K. P. Valavanis, "UAV-borne X-band radar for collision avoidance," *Robotica*, vol. 32, no. 01, pp. 97–114, jan 2014. [Online]. Available: http://www.journals.cambridge.org/abstract{_}S0263574713000659

[37] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "Quadrotor Using Minimal Sensing For Autonomous Indoor Flight," *European Micro Air Vehicle Conference and Flight Competition (EMAV2007)*, 2007. [Online]. Available: https://infoscience.epfl.ch/record/111485/

[38] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[39] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," *Pattern Recognition*, pp. 297–304, 2003.