IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 29, 2020, accepted October 19, 2020, date of publication October 22, 2020, date of current version November 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033157

Real-Time Human-UAV Interaction: New Dataset and Two Novel Gesture-Based Interacting Systems

MOHAMED A. KASSAB[1], MOSTAFA AHMED[1], ALI MAHER[1], AND BAOCHANG ZHANG[1,2], (Senior Member, IEEE)

[1]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2]Shenzhen Academy of Aerospace Technology, Shenzhen 100083, China

Corresponding authors: Mohamed A. Kassab (mohamedkassab99@gmail.com) and Baochang Zhang (Bczhang@139.com)

This work was supported in part by the Shenzhen Science and Technology Program under Grant KQTD2016112515134654.

ABSTRACT Two novel gesture-based Human-UAV Interaction (HUI) systems are proposed to launch and control a UAV in real-time utilizing a monocular camera and a ground computer. The first proposal is an end-to-end static Gesture-Based Interaction (GBI) system based on classifying the interacted user poses while discarding the gesture interpreting component to boost the system performance up to 99 % with a speed of 28 fps. On the other hand, the second proposal is a dynamic one that adopts a simple model to detect three parts (face and two hands) of the interacted person and tracks them for a certain number of frames till a specific dynamic gesture is recognized. The proposed dynamic method is efficient, decreases the complexity, and speeds the interaction up to 27 fps comparing with the recent multimodel ones. Its backbone is a simplified Tiny-You Only Look Once (YOLO) network saves the resources and speeds the detection process up to 120 fps. Moreover, a comprehensive new gestures dataset was established to facilitate the learning process and aid the research work. A comparative study is carried out to show the performance and efficiency of the proposed dynamic HUI system in terms of detection accuracy and speed with the baseline detector on the human gesture public dataset. Finally, a non-expert volunteer examines the proposed HUIs by launching and driving a Bebop 2 micro UAV through a set of real flights.

INDEX TERMS Gesture dataset, gesture recognition, human-machine interaction, object tracking.

I. INTRODUCTION

Machines became an essential part of humans' daily life. Consequently, interacting and dealing with them pushes to develop and propose different suitable Human-Machine Interaction (HMI) systems that can detect the interacted user's actions and achieve the desired output in real-time. Systems that respond to external events in a timely fashion are defined as the real-time systems [1] in which a correct output or action is needed alongside time constraint with a certain degree of flexibility. According to that degree of flexibility, two categories were defined. The first category is the soft real-time system in which catastrophic consequences due to a missed time constraint will not happen. In contrast, the other category is the hard real-time system in which catastrophic consequences will happen due to a missed time constraint.

Interacting and controlling a flying UAV is a hard real-time system in which the UAV must respond to the interacted person in a timely fashion without a degree of flexibility in a missed time constraint. It must detect the different objectives of the interacted person, recognize, distinguish between them, and produce its correct action in real-time. Otherwise, catastrophic consequences (human injury, UAV damage, etc.) will happen [1]. Consequently, diverse researches have been proposed to improve the controlling task of the UAV flight control system [2]–[6]. These researches are mostly divided into two main groups [7]. In the first group, the received measurements data from the onboard sensors are compared with the user's objectives within the onboard computer which produces suitable commands to control the flying UAV [2]–[4]. In the other group, the commanded flight control system

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed A. Zaki Diab.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

adopted a ground computer to generate the control law based on the comparison between the navigation data transmitted from the onboard sensors to this ground computer with the interacted user's objectives through the data communication link [5], [6].

Both groups can be equipped with different onboard sensors (inertial measurement units, rate gyroscope units, pressure sensors, cameras, etc.). These onboard sensors are responsible for providing information about the state of the flying UAV and its surroundings. Nowadays, the camera became the default onboard UAV sensor, it's a cheap, passive, and weightless sensor which can provide great information about the UAV's surroundings. As a result, diverse researches have been built to investigate the benefits of using a camera sensor in different applications, for instance, UAV target tracking [8], [9], aerial surveillance [10], [11], UAV path planning [12], fire observation [13], [14], and HMI [15], [16]. These types of researches used image information for important and practical applications in our daily life. However, they still depend on different sensors alongside the camera sensor for controlling the UAV.

Building a gesture-based HUI system mainly depends on accurate visual gesture recognition component [26] that must correctly detect, recognize, and interpret human gestures in real-time. Detecting the interacted person's gestures is the first and the main component in such systems. The detecting process is an object detection problem. Many approaches have been presented to handle this problem including handcrafted features-based approaches and deep features-based approaches. Handcrafted features-based approaches are less complex than deep features-based approaches. However, they suffer from limited performance.

Convolutional Neural Network (CNN) is a deep features-based approach that achieves great performance in detection and classification tasks [38]–[40]. However, it requires more resources and heavy computations. Motivated by the massive number of required resources and the heavy computations of CNNs, a new dataset comprises ten classes (used in the learning task of the first proposed HUI system) has been built we called the ''one frame gesture'' dataset where each frame is a gesture class e.g. Up, Forward, etc. and a developed model of Tiny-YOLOv2 [41] (used in the second proposed HUI system) has been proposed. This proposed model succeeded in saving some system resources, increased the deep neural network efficiency, and achieved almost the same performance.

Human gestures are categorized into two types; static and dynamic gestures [27]–[30]. In the static gestures, there is no change in pose across time for a certain number of frames which means that every gesture should be stationary for a certain number of frames [22], [31]. On the dynamic gestures side, different poses are collected across time to form a gesture, which means that every gesture is a combination of different poses for a certain number of frames [32], [33]. This type is more complex than the static gestures because it's usually handled by multimodel methods [34]–[36]. Both static and dynamic gestures are important as natural human interacting methods that can express uninstrumented (a person who is not being equipped with any hardware to help in the process of interacting and controlling the machine) person's demands. Therefore, two novel gesture-based recognition systems are proposed which can be used for HUI systems to launch and control a UAV in real-time utilizing only one RGB camera sensor connected to a ground computer observing the interacted person's gestures to launch and control UAV in real-time while utilizing the onboard camera of the controlled UAV as a feedback sensor. Moreover, the two proposed interacting systems are dealing with an uninstrumented person.

The first proposed HUI system is a novel end-to-end interacting system based on static gesture recognition through classifying the interacted user poses and it abandoned the gesture interpreting component (this is the first HMI system that abandoned the gesture interpreting component) from the system which leads to highly efficient and accepted interaction performance, thus we called ''static end-to-end HUI system''.

The second proposed HUI system is a novel interacting system that can handle dynamic gesture which is preferred because it is more natural in human interaction. It is based on dynamic gesture recognition but without using the multimodel methods. It adopted a proposed developed model of tiny-YOLOv2 [41] to detect three parts (face and two hands) of the interacted person. As a result, it decreases the complexity, meanwhile improving efficiency and ensures high performance. Moreover, this system handles dynamic gesture as a tracking problem in which the three detected parts are tracked using Kalman Filter (KF) [42] tracker for a certain number of frames till a specific dynamic gesture is recognized, thus we called ''dynamic HUI system''.

Although the two proposed systems have a great dependency on a recognition model that may fail during the interaction (we solved this problem by propagating the default Hovering command when there are no classifications or no detections) and the distance of the WIFI data communication link that limits the range between the ground control computer and the interacted machine, they have six advantages over existing approaches: (1) They can be easily modified to interact with a massive number of applications rather than UAVs. (2) They depend on low-cost equipment and a reasonable amount of system resources; however, they achieved promising results in GBI methods. (3) They can control from another place as long as there is a communication link between the GBI ground computer and the interacted machine. (4) The interacted person does not need any special instruments to control the interacted machine. (5) The static HUI system is an end-to-end system that does not depend on the gesture interpreting component. (6) The dynamic HUI system handles dynamic gestures without using multimodel methods or special sensors.

The contributions of this paper are summarized as follows:

- Developing an effective model of Tiny-YOLOv2 that can achieve the same performance and save resources with more efficiency.
- Designing and implementing a robust and simple novel static end-to-end HUI system that abandons the gesture interpreting component from the system. This is the first HMI system that abandons the gesture interpreting component.
- Building a new dataset comprises ten classes represent ten gestures to help in the learning task of the proposed static end-to-end HUI system.
- Designing and implementing a robust and simple novel dynamic HUI system that can handle dynamic gesture which is preferred because it is more natural in human interaction without using the multimodel methods.
- Increasing the dynamic gesture recognition efficiency and performance by incorporating the KF tracker providing a reduction of both detected frames and miss detected objects.

The structure of this paper is outlined as follows: the related work is presented in Section II. The proposed dataset is designed in Section III. Section IV and V describe the proposed static end-to-end HUI system and the proposed dynamic HUI system respectively. The performance and efficiency of the proposed approaches are demonstrated in Section VI. Finally, Section VII states the conclusions.

II. RELATED WORK

Recently, significant HMI methods have been presented that depend on remote controllers, wearable sensors like Fly jacket [17], EMG sensor [18], or Hand glove [19]. However, practically these methods suffer from extra difficulties due to the special requirements to interact with the instrumented person. Therefore, these difficulties encourage researchers to investigate natural interacting methods like gestures [20]–[22] and speech recognition [23], [24] to deal with the uninstrumented person. Although the speech recognition method is a natural interacting method, it is not an applicable method due to the associated background noise next to the different accents of different people. As a result, it requires a compulsory grammar component to facilitate the complexity of the speech [25]. On the other hand, gesture has the natural form of communication between people and it does not require compulsory components, so it's the most suitable and intelligent method for HMI systems.

A diverse number of researches has been done to improve and propose different methods for interacting between humans' gestures and machines. Some of these different interacting methods utilize different sources of information to improve the interacting results, for instance, skeleton joints or depth, these methods are called multimodel methods [34], [43], [44]. In [34], Wang *et al.* used a frequency modulated continuous wave radar to obtain the hand gesture row data that facilitated hand gesture recognition task however their system suffer from dependency on a heavy active sensor (radar) whose signal can be jammed by another active device in comparison with the camera sensor.

Multimodel methods need special sensors to acquire the additional information. These special sensors maybe the Microsoft Kinect [33], [45]–[47], Intel Real sense [48]–[51], or Asus Xlion Pro [52]–[54]. Consequently, depending on special sensors makes multimodel methods harder to be applicable in all environments, especially with UAVs where the payload and power consumption are limited.

Recently, with the huge amount of surveillance and different machines' on-board RGB cameras, different types of interacting methods have emerged that depend only on image color information. Some of them used traditional handcrafted feature-based methods [3], [55], [56]. In [3], Nagi *et al.* presented an approach based on detecting the interacted user's face poses and hands to control a flying UAV. In [55], Monajjemi *et al.* presented a Human-UAV Interaction (HUI) system that can detect and recognize only three dynamic gesture which stimulates a flying UAV to come to the interacted user and follow him.

In [56], Nagi *et al.* presented a distributed consensus protocol that can be used in the controlling process of Human-Swarm Interaction (HSI). While, the majority used the deep feature-based methods [22], [35], [36]. In [22], Zhang *et al.* presented an approach for moving and control Automated Guided Vehicle (AGV) with different eight static gestures based on Mobile Net-SSD CNN [57]. In [35], an approach to detect and classify dynamic gestures using more than one three-dimensional CNN was presented. In [36], Santos *et al.* presented an approach to classify dynamic gestures based on two Resnet CNN [37].

All the aforementioned methods depend on using different sources of information to improve gesture recognition, using traditional handcrafted feature-based methods, or using only the image's color information alongside increasing the number of detection and recognition CNNs. Practically these methods suffer from extra difficulties because using different sources of information is not applicable in all environments, using the traditional handcrafted feature-based methods deals with a very limited number of distinguished gestures, and increasing the number of detection and recognition CNNs is not recommended in hard real-time scenarios because it requires more resources and heavy computations.

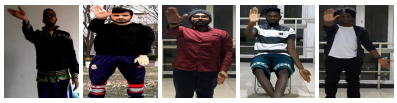
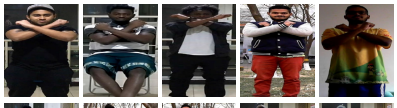
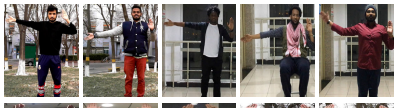
Therefore, two proposed HUI systems have been proposed that instead of depending on different sources of information, they only depend on one RGB camera sensor connected to a ground computer. Moreover, instead of using multimodel methods to handle dynamic gestures, only one simple proposed CNN model is used.

III. THE PROPOSED DATASET

A. DATASET COLLECTION

A "one frame gesture" dataset has been built from 38 videos taken by 12 volunteers in different environments indoor (22 videos) and outdoor (16 videos) and comprises ten classes, in which each frame is a gesture class e.g. Up,

TABLE 1. The different class labels and some examples from the proposed dataset.

Label	Class Name	Some Examples
0	Backward	
1	CCW	
2	CW	
3	Down	
4	Forward	
5	Takeoff / Land	
6	Left	
7	Right	
8	Up	

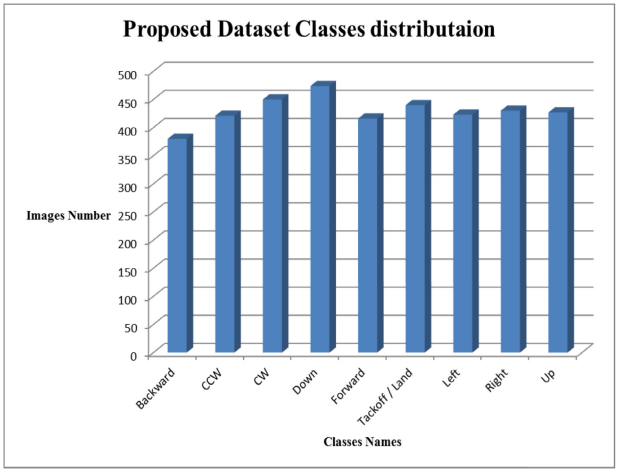


FIGURE 1. The distribution of the proposed dataset's samples reveals its balance.

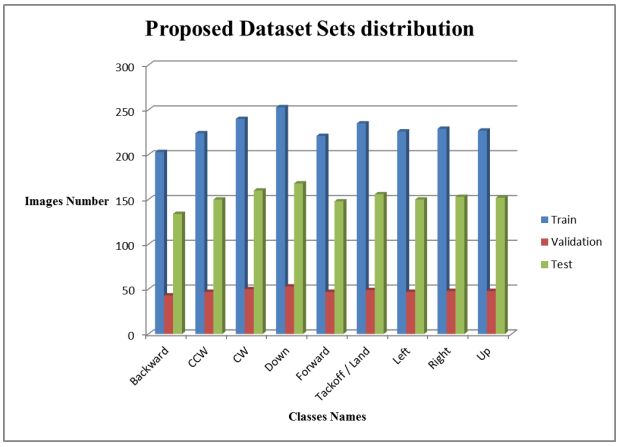


FIGURE 2. The number of images for training, validation, and testing samples for each class.

Forward, etc., each video duration was about 25 s with 1920 × 1080 image's resolution at 30 fps. Then every 5 frames, one frame from each video was selected. After removing the similar frames, we got 3861 images. To focus on the human body gesture in each image, the background of all images was cropped which produced an image's resolution of 420 × 1040 in JPG format.

Images were manually labeled with a code from 0 to 8 by some experts. Then these labels were revised by another team. Table 1 shows class labels, names, and some examples from the proposed dataset demonstrating each class with the corresponding human body gesture. Then, ten different human body gestures were generated by defining ten different classes. These classes include Backward, CCW (Counter Clock Wise), CW (Clock Wise), Down, Forward, Takeoff / Land, Left, Right, Up and the transition frames between any two successive gestures are represented by the Hovering class. Fig.1 shows the distribution of each class.

B. DATASET EVALUATION

The proposed dataset was divided into 2058 and 432 images for training and validation respectively. While the remaining 1371 images were used for the testing phase. This division process was performed according to each class distribution to ensure the overall balance as shown in Fig. 2. In the training phase, a Deep Learning GPU Training System (DIGITS) was used which allows a real-time network behavior visualization through an interactive browser-based interface. This training system was introduced by Nvidia in 2015 to develop, train, and visualize different Deep Neural Network (DNN) models. Fig. 3 shows three classes (Up, Down, and Forward) from the proposed dataset and the corresponding visualization of the first three convolutional layers' feature maps of each class.

Three deep classification frameworks were used for the proposed dataset evaluation. These deep classification frameworks were the Faster R-CNN backbone (RPN) [38], the VGG-16 [58], and the YOLOv2 backbone (Darknet-19) [41]. The results after testing these three CNN proves the effectiveness of our proposed dataset. Where the RPN achieved 100 % accuracy with 1 mean Average Precision (mAP), the VGG-16 achieved 98.76 % accuracy with 0.9999 mAP, and the Darknet-19 achieved 99.92 % accuracy

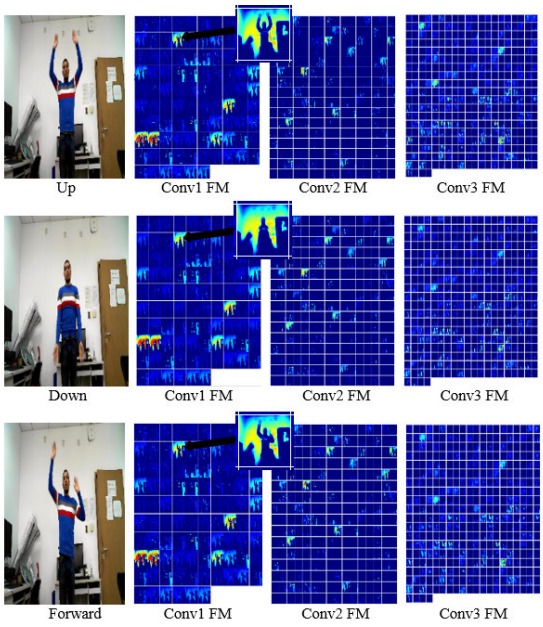


FIGURE 3. Samples (Up, Down, and Forward) from the proposed dataset's classes and the corresponding visualization of the first three convolutional layers' feature maps (FM).

with 1 mAP. The testing phase is explained in detail in Section VI.

IV. THE PROPOSED STATIC END-TO-END HUI SYSTEM

Traditional gesture-based HUI systems mainly consist of two components: a gesture recognition component and a gesture interpreting component. The first proposed HUI system is based on abandoning the gesture interpreting component from the system. Therefore, a novel static end-to-end interacting system has been introduced. It depends only on the gesture recognition component by classifying the interacted person's poses throw a deep CNN after training on the proposed one frame-gesture dataset (Section III). Consequently, the classification of each frame is used to propagate a command that will control the flying UAV. Therefore, choosing this deep CNN is very critical in this proposed system.

As concluded from the experimental results (Section VI-A), the RPN cannot be utilized for real-time applications because it classifies frames with 4.08 fps. While the VGG-16 classifies frames with 26.4 fps and the Darknet-19 classifies frames with 28.3 fps. Consequently, the last two models can be utilized in real-time applications.

As to the performance of these two models, the VGG-16 achieved 98.76 % accuracy and mAP = 0.9999, and the Darknet-19 achieved 99.92 % accuracy and mAP = 1. The total accuracy and the mAP of the Darknet-19 model are better than VGG-16. Then the Darknet-19 was utilized in this proposed system as the deep classification network.

The architecture of this proposed system is shown in Fig. 4. A camera sensor is connected to a ground computer having Intel® Core™ i7-7700HQ CPU @ 2.8 GHz × 8 with NVIDIA GPU GeForce GTX 1050 Ti which communicates with the Bebop 2 UAV (Bebop 2 UAV is designed to digitally stabilize, rectify the live video stream of its onboard camera then transmit it at 30 fps) via a wireless communication link. On this ground computer, there are three nodes: input camera node, deep classification node, and monitoring camera node. These three nodes were implemented in the Robot Operating System (ROS) framework [59], nodes in ROS can communicate with each other by publishing (sending) or subscribing (receiving) according to the ROS transport protocol. The monitoring camera node is used only to monitor the received frames from the UAV. These frames are used as feedback that will be seen by the user to decide the suitable gesture for the UAV controlling. The input camera node and the deep classification node form the gesture recognition component of this proposed system.

The ground computer's camera sensor observes the interacted person's gestures and provides frames at 30 fps to the ground computer, the input camera node subscribes these frames and publishes them. Then, the deep classification node subscribes them frame by frame, classifies each frame with a class, and uses this class as a posture to publish the corresponding command that will be transmitted via the wireless communication link to control the flying UAV. Ten commands are corresponding to the ten gestures from the proposed one frame-gesture dataset: Backward, CCW, CW, Down, Forward, Takeoff / Land, Left, Right, Up, and the default Hovering. Fig. 5 shows some frames from the proposed dataset with the corresponding controlling commands where each human body gesture indicates a different control command.

The Takeoff command will be propagated only if the gesture recognition component recognizes a certain gesture sequence from two successive frames. This sequence must be the Down class followed by the Takeoff class (e.g., see Fig. 6) to inform the proposed system that the user attends to start the interaction and overcome the False Positives (FP) problem.

Four rules were defined to ensure the robustness of this proposed system, which are:

1) The confidence score of accepted classification should be more than 90 %.
2) The commands must follow a specific order, which means that the controlling commands or the Landing command can be propagated only after the Takeoff command.
3) The last command is the propagated command until recognizing another valid one.
4) The default Hovering command will be propagated if there are no valid classifications or more than one classification.

During the interaction, the flying UAV sends frames (we use these frames as feedback) at 30 fps from its onboard camera via the wireless communication link to the ground computer. The monitoring camera node subscribes these frames and publishes them on the ground computer screen. Consequently, the user sees them and decides a suitable gesture to control the UAV.

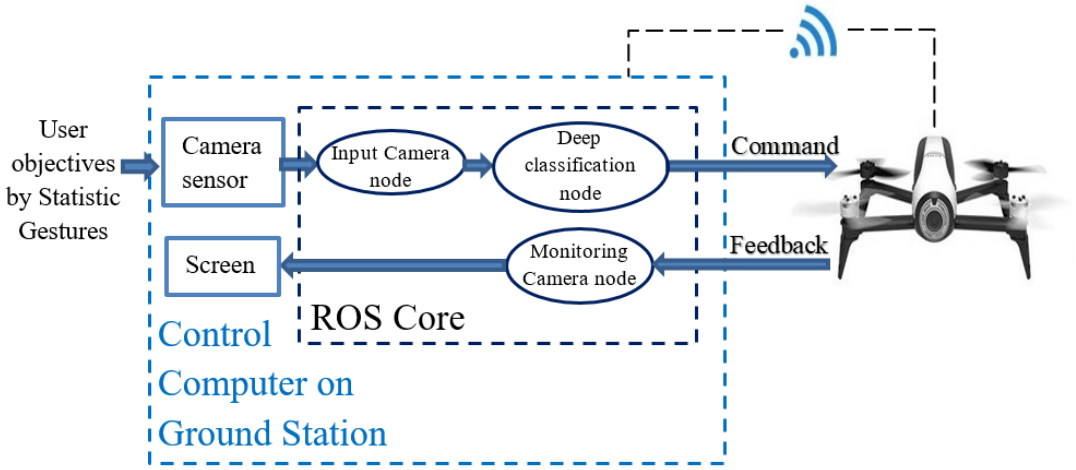


FIGURE 4. The proposed static gesture HUI system architecture with bebop 2 drone.

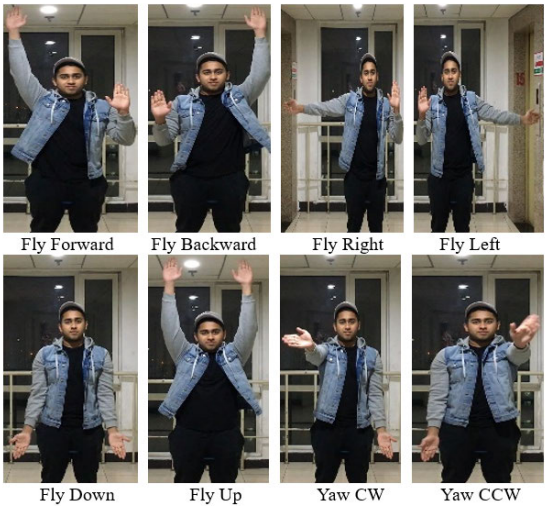


FIGURE 5. Some frames from our proposed dataset with the corresponding controlling commands.



FIGURE 6. Two successive frames to indicate the start of the interaction. The left is a sample of Down class while the right is a sample of Takeoff / Land class.

V. THE PROPOSED DYNAMIC HUI SYSTEM

A. OVERVIEW

Traditional dynamic gesture-based HUI systems are usually based on using the multimodel methods, or dealing with the instrumented person as illustrated in Section II. Practically these methods suffer from extra difficulties and not applicable in all environments. Therefore, the second proposed HUI system is based on a simple proposed object detection model. This proposed system can handle the dynamic gesture by detecting three parts (face and two hands) of the interacted person using our simple proposed developed model of Tiny-YOLOv2 [41]. Then tracking these detected parts using the simple KF [42] tracker for a certain number of frames till a specific dynamic gesture is recognized. Therefore, a novel dynamic interacting system has been introduced that can deal with the uninstrumented person without using the multimodel methods.

The architecture of this proposed system is shown in Fig. 7. Note: The camera sensor and the ground computer in this proposed system have the same properties as the first proposed static HUI system. Also, all nodes were implemented in ROS. However, in this proposed system, there are five ROS nodes: input camera node, deep detection node, tracker node, interpreter node, and monitoring camera node. The monitoring camera node is used only to monitor the received frames from the UAV as illustrated in the first proposed static HUI system (Section IV). The input camera node, the deep detection node, the tracker node, and the interpreter node form the gesture recognition component of this proposed system.

The ground computer's camera sensor observes the interacted person's gestures and provides frames at 30 fps to the ground computer, the input camera node subscribes these frames and publishes them. Then, the deep detection node subscribes them frame by frame, detects objects inside each

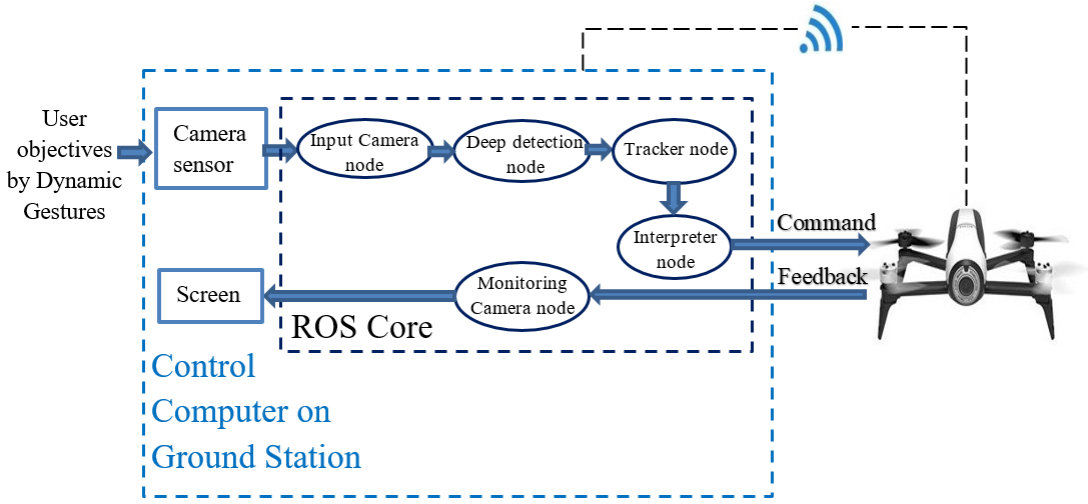


FIGURE 7. The proposed dynamic gesture HUI system architecture with bebop 2 drone.

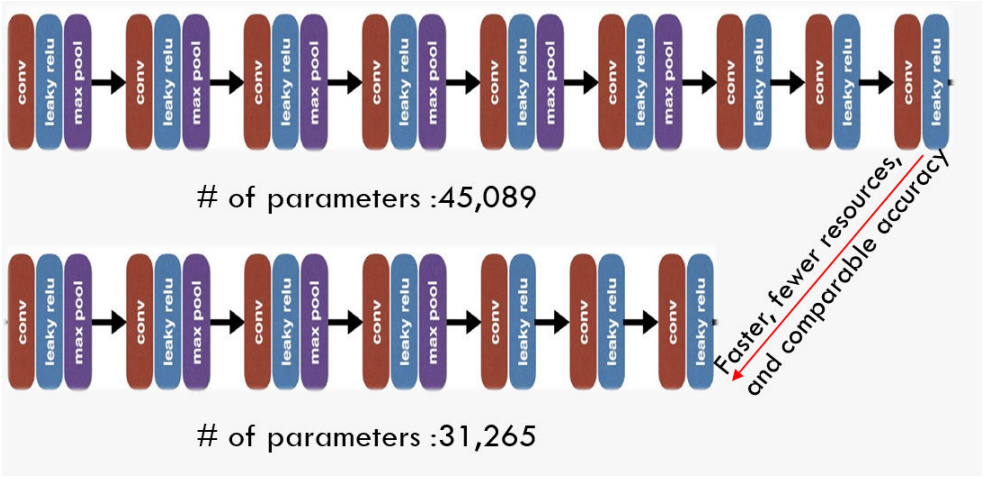


FIGURE 8. The Tiny-YOLOv2 and the proposed model architectures with the number of reduced parameters.

frame, and check these detections until finding three detections of one face and two hands to publish them. After that, the tracker node subscribes these three detections and publishes three tracks. Then, the interpreter node subscribes these tracks for every five frames to form a sequence of postures that represents a dynamic gesture. Consequently, the interpreter node publishes the corresponding command that will be transmitted via the wireless communication link to control the flying UAV.

There are ten commands in this proposed system, eight of them: Hovering, Fly forward, Fly backward, Fly right, Fly left, Fly up, Yaw CW, and Yaw CCW are corresponding to eight dynamic gestures while the Takeoff and the Landing commands are corresponding to static gestures (gesture from only one posture) because these two commands have special issues which are the Takeoff command is followed by the default Hovering command and no commands will be propagated after the Landing command except the Takeoff command.

During the interaction, the flying UAV sends frames (we use these frames as feedback) at 30 fps from its onboard camera via the wireless communication link to the ground computer for the same purpose as illustrated in the first static proposed HUI system (Section IV).

B. DYNAMIC GESTURE DEEP DETECTION NETWORK

To detect objects with high speed, and preserve the detection accuracy, a developed model of Tiny-YOLOv2 [41] has been proposed. The proposed model reduced the number of the required parameters and saved more system resources. Fig. 8 shows the Tiny-YOLOv2 and the proposed model architectures and indicates the number of reduced parameters. In Table 2, the comparison results between the two models demonstrate the resources

TABLE 2. Comparison results between the Tiny-YOLOv2 and the proposed model with the resources optimization in terms of the GPU RAM and the weight file size.

Method	GPU DRAM (GB)	Weight file size (MB)
Tiny-YOLOv2	1.12	64.2
Proposed model	0.853	9.9

TABLE 3. The proposed model architecture.

Layer	Filters	Size	Stride	Input	Output
C1	16	3×3	1	416×416×3	416×416×16
MP1		2×2	2	416×416×16	208×208×16
C2	32	3×3	1	208×208×16	208×208×32
MP2		2×2	2	208×208×32	104×104×32
C3	64	3×3	1	104×104×32	104×104×64
MP3		2×2	2	104×104×64	52×52×64
C4	128	3×3	1	52×52×64	52×52×128
MP4		2×2	2	52×52×128	26×26×128
C5	1024	3×3	1	26×26×128	26×26×1024
C6	1024	3×3	1	26×26×1024	26×26×1024
C7	14	1×1	1	26×26×1024	26×26×14

optimization in terms of the GPU RAM and the weight file size.

Tiny-YOLOv2 utilizes nine convolutional layers and six polling layers for downsampling, it needs 45089 parameters (see Fig. 8) and requires 1.12 GB of GPU DRAM and a size of 64.2 MB for weight file (see Table 2). It is fast, but it took about 11.36 ms for detecting objects in one frame on our ground computer. Consequently, it took about 56.8 ms for detecting objects in five frames to form a sequence of postures that represents a dynamic gesture. Therefore, it reached a speed of 17.26 fps for dynamic gesture recognition (see Table 12).

As shown in Fig. 8, the proposed model architecture is configured as follows: *C1, MP1, C2, MP2, C3, MP3, C4, MP4, C5, C6, C7*. Where *C* stands for the convolutional layer and *MP* for the max pooling layer. This model utilizes only seven convolutional layers and four max pooling layers for downsampling. The convolutional layers are activated by Leaky ReLU function. The kernel sizes for the first six convolutional layers are 3×3 with a stride of 1 and is 1×1 for the last one (*C7*). The total number of the feature maps is 16, 32, 64, 128, 1024, 1024, and 14 for *C1, C2, C3, C4, C5, C6,* and *C7* respectively as illustrated in Table 3. Consequently, it needs only 31265 parameters and requires 0.853 GB of GPU DRAM and a size of 9.9 MB for weight file as mentioned in Table 2.

However, it achieved more efficiency next to preserving almost the same performance as the Tiny-YOLOv2. It took about 8.33 ms for detecting objects in one frame. Consequently, it took about 41.65 ms for detecting objects in five frames. Therefore, it reached a speed of 23.37 fps for dynamic gesture recognition. Moreover, by reducing the polling layers, more spatial information will be preserved (like face and hand postures) which is essential for this interacting system. Therefore, the proposed model succeeded in saving some resources and increasing the deep CNN efficiency with achieving the same performance as illustrated in Table 11 and Table 12.

The human gesture dataset presented by [60] was collected by taking one frame every 5 frames from 56 videos taken at 30 fps in different environments indoor (18 videos) and outdoor (38 videos). Each video duration was about 30 s. Thus, it contains 6223 images of human gestures focusing on face and hand. We used it to train and test the Tiny-YOLOv2 [41] and the proposed model for detecting certain classes (face and hand). It was divided into 3733 images for the training phase where a cross-validation scheme was implicitly carried out [41] during this phase by utilizing nine classes' frames for training while utilizing the rest class's frames for validation and keep changing the validation class till utilizing all ten classes as a validation class and 2490 images for the testing phase for both models.

According to the testing results demonstrated in Table 11, the two models almost achieved a similar mAP. However, the proposed model achieved higher efficiency by increasing its speed to 120 fps. This speed will aid in dynamic gesture recognition. Therefore, this proposed model was used in the proposed dynamic HUI system as its deep detection network to detect human face and hand. These detections are represented as Bounding Boxes (BB) with labels. Each label represents a detected object class, and the BB represents a rectangular around this object with x_c, y_c, w, and h. Where x_c and y_c are the center values in the horizontal and the vertical directions respectively, w is the width, and h is the height.

C. DYNAMIC GESTURE TRACKER AND INTERPRETER NODES

Deep detection node publishes detections of the interacted person's face and hands. The tracker node subscribes these detections, observes them till it finds only one face and two hands to overcome FP gestures. Then the KF [42] tracker tracks these detections' centers (centers of both hands and face) to estimate their new locations according to the defined states in (1) and produces tracks for them. Using KF tracker to track these center points enables the dynamic HUI system to use tracks' center points not detections' center points in the gesture interpreting component. Consequently, a reduction in the number of the detector model processed frames is achieved which increases the system efficiency as illustrated in (Table 12) in addition to overcome the miss detection problem produced if the detector fails to detect objects inside frames while forming the dynamic gestures. Therefore, each object geometry state will be:

$$X_g = [u, v, \dot{u}, \dot{v}] \tag{1}$$

where (u, v) is the center of the estimated BB.

The interpreter node distinguishes between the two estimated states of both hands of the interacted person by indicating the hand with smaller u as the right hand (R_h) and the other as the left hand (L_h). Then, it collects every estimated center point from every n (set here to five) frames as a sequence of postures. These sequences of postures form our three

dynamic gestures: face dynamic gesture, right-hand dynamic gesture and left-hand dynamic gesture from (2)

$$G = [g_i \quad g_{i+1} \quad \dots \quad g_{n-1}]^T \tag{2}$$

where n is the number of frames to collect the dynamic gesture, and g_i is the gesture estimated center of the i^{th} frame obtained from (3) with $i = 0, 1, \dots, n-1$.

$$g_i = [u_i, v_i] \tag{3}$$

After obtaining three dynamic gestures: face dynamic gesture (G_f), right-hand dynamic gesture (G_{Rh}), and left-hand dynamic gesture (G_{Lh}), different commands are produced according to the dynamic gesture design (explained in detail in Section V-D). Then the interpreter node publishes these commands to control the flying UAV by transmitting via the wireless communication link between the ground computer and the UAV.

D. DYNAMIC GESTURE DESIGN

From the obtained three dynamic gestures (G_f, G_{Rh}, and G_Lh), three difference vectors are calculated. These calculated vectors indicate the behavior of each dynamic gesture by calculating the difference between every two successive frames. These three vectors are: face difference vector (D_f), right-hand difference vector (D_{Rh}), and left-hand difference vector (D_{Lh}) which calculated from (4):

$$D = [d_i \quad d_{i+1} \quad \dots \quad d_{n-2}]^T \tag{4}$$

where n is the number of frames to form the dynamic gesture, and d_i is the difference between the estimated state's center of the i^{th} frame and the previous one that was obtained from (5) with $i = 0, 1, \dots, n-2$ and (6).

$$d_i = [(u_{i+1} - u_i), (v_{i+1} - v_i)] \tag{5}$$

$$u_{i+1} - u_i = \begin{cases} +ve & \text{if } (u_{i+1} - u_i) > th \\ 0 & \text{if } -th \leq (u_{i+1} - u_i) \leq th \\ -ve & \text{if } (u_{i+1} - u_i) < -th \end{cases} \tag{6}$$

where (u_i, v_i) is the center of the estimated BB of the i^{th} frame and th is the threshold value which indicates the acceptable value that defines the stationary object, we adjusted th to a value of 3.

Therefore, a command is produced according to certain combinations from those difference calculated vectors. These combinations indicate dynamic gesture behaviors and can be obtained from the negative and the positive sign in the values of the three difference vectors as calculated from (5) and (6).

Four rules were defined to increase the system reliability, which stated as follows:

1) The interacted person's face should be stationary, which means no valid commands will be sent with a moving face.
2) The commands must follow a specific order, which means that the controlling commands or the Landing command can be propagated only after the Takeoff command.
3) The last command is the propagated command until recognizing another valid one.
4) The default Hovering command will be propagated if there are no detections or more than one face and two hands.

With the previously mentioned rules, the default command is the Hovering command that is propagated when the detected face moves in any direction. The other commands: Fly forward, Fly backward, Fly right, Fly left, Fly up, Yaw Clock Wise (CW), and Yaw Counter Clock Wise (CCW) are propagated only if a stationary face and one or two moving hands are detected. A certain command from them will be determined according to the two hands dynamic gesture behavior.

The commands: Fly forward, Fly right and Yaw CW are propagated only if the face and the left-hand dynamic gesture are stationary. If the right-hand dynamic gesture changes only in the negative vertical direction, the Fly forward command is propagated, if it changes only in the negative horizontal direction, the Fly right command is propagated, and if it changes in both negative horizontal and positive vertical directions, the Yaw CW command is propagated.

The commands: Fly backward, Fly left, and Yaw CCW are propagated only if the face and the right-hand dynamic gesture are stationary. If the left-hand dynamic gesture changes only in the negative vertical direction, the Fly backward command is propagated, if it changes only in the positive horizontal direction, the Fly left command is propagated, and if it changes in both positive horizontal and vertical directions, the Yaw CCW command is propagated.

The Fly up command is propagated if the face dynamic gestures are stationary and the two hands dynamic gestures change at the same time in the negative vertical direction.

Table 4 summarizes the combinations of the three difference vectors with the movement direction of each gesture and the corresponding commands where the hand's movement speed should be more than 0.25 m/s and less than 0.35 m/s in the direction of each gesture which allows the interpreter to form the dynamic gesture and the corresponding command during interacting. Fig. 9 shows real samples as a detailed example of the Fly forward command. The frames that indicate the different combinations of the three difference vectors are shown. These combinations generate the controlling command according to the changes in the dynamic gestures behaviors. Fig. 10 shows real samples of the frames that indicate the different combinations of the three difference vectors of the other controlling commands together in one figure.

Both Takeoff and Landing commands have a special issue. The Takeoff command must be followed by the default Hovering command, and there are no commands after the Landing command except the Takeoff command. Therefore, these two commands are propagated using only static gesture recognition, which means that these commands are propagated from only one posture. To implement these two static commands, we utilized the formula presented by [60] after we modified

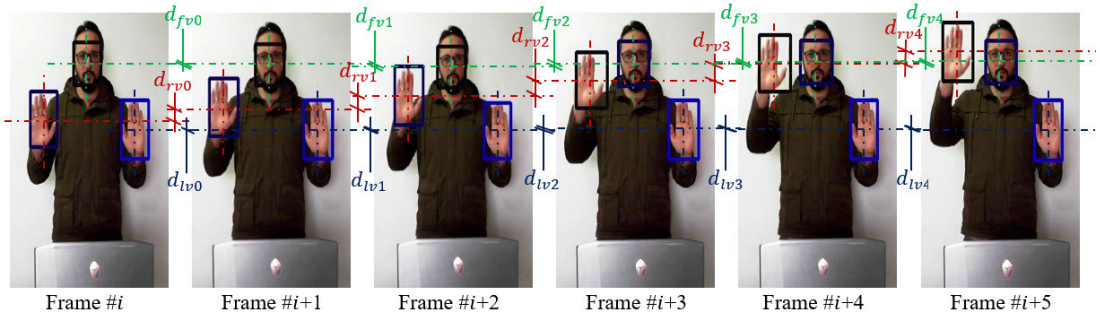


FIGURE 9. Real samples of the Fly forward command. A sequence of frames indicates the calculations of the three difference vectors, d_{rvi}, d_{fvi}, and d_{lvi} represent the difference between each vertical component from each BB' center of the i^{th} frame and the previous one for right-hand dynamic gesture, face dynamic gesture, and left-hand dynamic gesture respectively. It's clear that the change in the right-hand gesture is in the negative vertical direction as each d_{rvi} value is smaller than the negative value of *th* while all d_{fvi} and d_{lvi} values are bigger than the negative value of *th* and smaller than the positive value of *th*, so the last two are stationary. Note: only the vertical direction is demonstrated on the figure.

TABLE 4. The three dynamic gestures combinations and commands.

Face dynamic gesture	Right-hand dynamic gesture	Left-hand dynamic gesture	Movement direction	Command
Changing in any direction	Do not care	Do not care	Do not care	Hovering
Stationary	Changing only in $-ve$ vertical direction	Stationary	Upward	Fly forward
Stationary	Stationary	Changing only in $-ve$ vertical direction	Upward	Fly backward
Stationary	Changing only in $-ve$ horizontal direction	Stationary	Rightward	Fly right
Stationary	Stationary	Changing only in $+ve$ horizontal direction	Leftward	Fly left
Stationary	Changing only in $-ve$ vertical direction	Changing only in $-ve$ vertical direction	Upward	Fly up
Stationary	Changing in $-ve$ horizontal and $+ve$ vertical directions	Stationary	Downward with 45 degree angle to the right side	Yaw CW
Stationary	Stationary	Changing in $+ve$ horizontal and $+ve$ vertical directions	Downward with 45 angle to the left side	Yaw CCW

it to work with our proposed interacting system (see Fig. 11) where two ratios R_1 and R_2 are used to devise commands and calculated from (7) and (8):

$$R_1 = \frac{y_f}{y_{Rh}} \tag{7}$$

$$R_2 = \frac{y_f}{y_{Lh}} \tag{8}$$

where y_f, y_{Rh}, and y_{Lh} are the detected BB centers in the vertical component of the face, right hand, and left hand respectively.

Table 5 summarizes how to use the detected BB centers in the vertical component to calculate the two ratios: R_1 and R_2,

TABLE 5. Using the two ratios R_1 and R_2 to generate commands.

Hands position w.r.t. face position	R_1	R_2	Command
The two hands having the same value in the vertical direction and under the face.	0.2 : 0.4	0.2 : 0.4	Takeoff
The two hands are far and above the face in the vertical direction.	> 2.0	> 2.0	Landing

which used to propagate these two commands. The Takeoff command is propagated if the values of R_1 and R_2 are between 0.2 and 0.4, which means that the two hands having almost the same detected BB center's value in the vertical direction and under the face. The Landing command is propagated if the values of R_1 and R_2 are greater than 0.2, which means that the two hands are far from the face and above.

VI. EXPERIMENTAL RESULTS

This section is organized into three parts: experiments on classifications, experiments on detections and tracking, and experiments on real flights. Firstly, the experiments on classifications investigate the performance after training and testing three well known CNNs on our proposed one-frame gesture dataset and the comparison results between the proposed end-to-end static HUI system and the static HUI system presented by [60]. Secondly, the experiments on detections and tracking investigate the performance and the efficiency of the proposed dynamic HUI system, a comparison between our proposed developed model and the Tiny-YOLOv2 model [41] on a public dataset (human gesture dataset), and the comparison results between the proposed dynamic HUI system with three recent existing dynamic approaches. Finally, the experiments on real flights to prove our proposed HUI system in launching and controlling UAV.

A. EXPERIMENTS ON CLASSIFICATION

As mentioned in Section III-B, we divided the proposed dataset into 2058 and 432 images for training and validation respectively, while the remaining 1371 images were used for testing. The three deep CNNs: Faster R-CNN

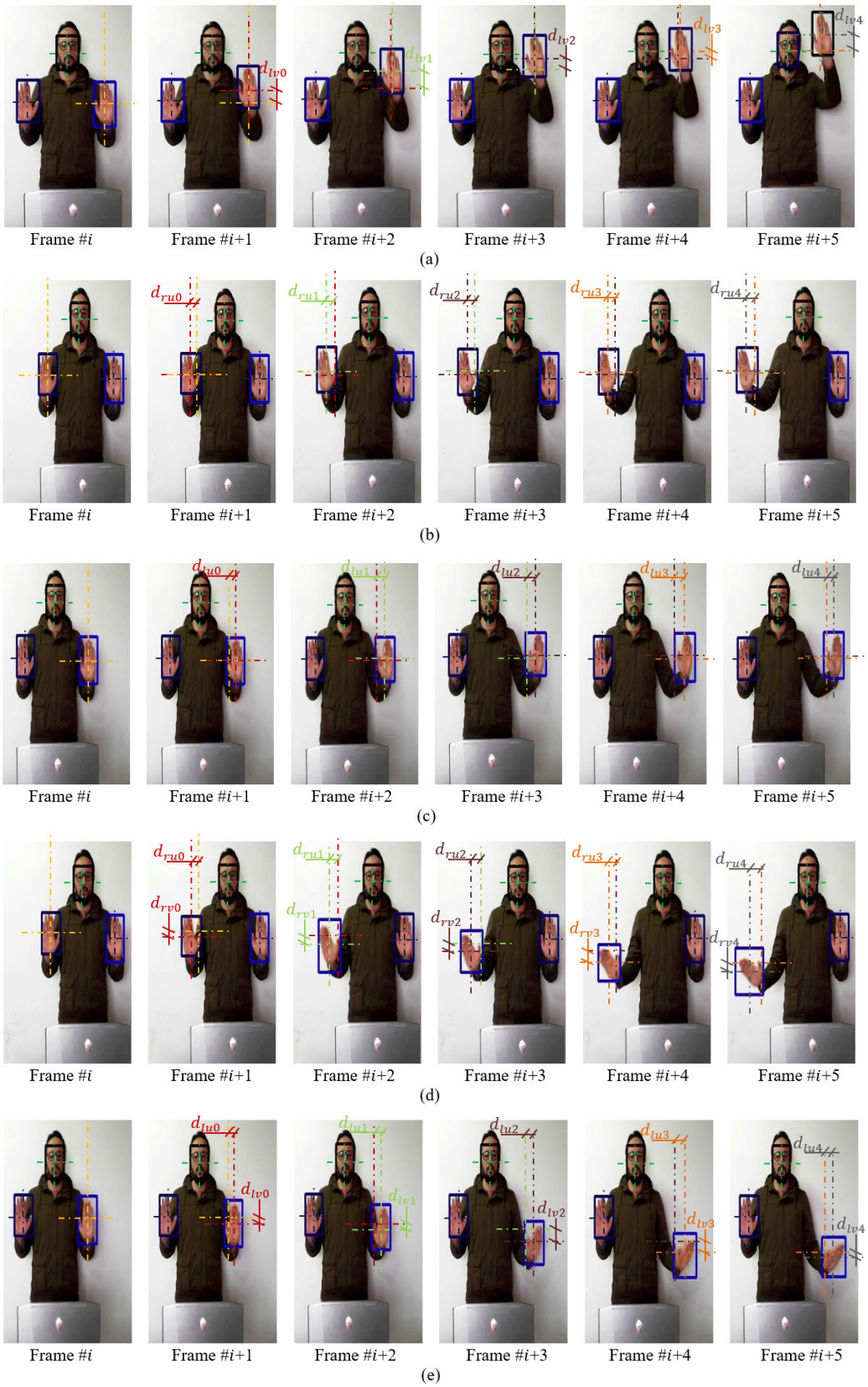


FIGURE 10. Real samples of the control commands. Note: only the changed directions are demonstrated on the figures while the stationary is not demonstrated. (a) The Fly backward command. (b) The Fly right command. (c) The Fly left command. (d) The Yaw CW command. (e) The Yaw CCW command.

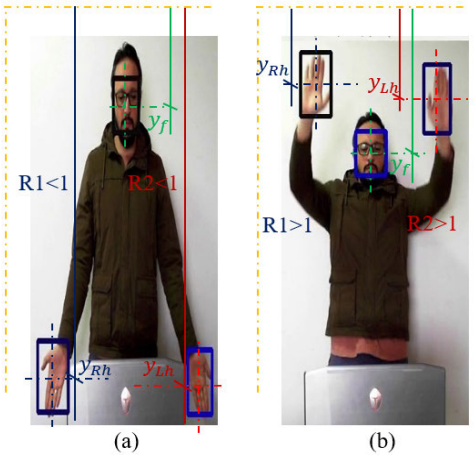


FIGURE 11. Real samples for obtaining the two static commands using ratios R_1 and R_2. (a) Taking off command when the two detected hands under the face so R_1 and R_2 less than 1. (b) Landing command when the two detected hands above the face so R_1 and R_2 more than 1.

TABLE 6. Confusion matrix for RPN on the proposed one frame-gesture dataset, "Bd" and "Fd" refer to Backward and Forward respectively.

Class	Bd	CCW	CW	Down	Fd	Land	Left	Right	Up
Bd	1	0	0	0	0	0	0	0	0
CCW	0	1	0	0	0	0	0	0	0
CW	0	0	1	0	0	0	0	0	0
Down	0	0	0	1	0	0	0	0	0
Fd	0	0	0	0	1	0	0	0	0
Land	0	0	0	0	0	1	0	0	0
Left	0	0	0	0	0	0	1	0	0
Right	0	0	0	0	0	0	0	1	0
Up	0	0	0	0	0	0	0	0	1

TABLE 7. Confusion matrix for VGG-16 on the proposed one frame-gesture dataset, "Bd" and "Fd" refer to Backward and Forward respectively.

Class	Bd	CCW	CW	Down	Fd	Land	Left	Right	Up
Bd	0.9925	0	0	0	0.0074	0	0	0	0
CCW	0	0.9933	0	0	0	0	0	0	0
CW	0	0	1	0	0	0	0	0	0
Down	0	0	0	1	0	0	0	0	0
Fd	0	0	0	0	0.9594	0	0	0	0
Land	0	0	0	0	0	1	0	0	0
Left	0	0	0	0	0	0	1	0	0
Right	0	0	0	0	0	0	0	1	0
Up	0	0	0	0	0	0	0	0	0.9407

backbone (RPN) [38], VGG-16 [58], and YOLOv2 backbone (Darknet-19) [41] were used for the evaluation. After testing, the results prove that the proposed dataset is effective.

As to performance, the RPN achieved 100 % accuracy and mAP = 1 (see Table 6), the VGG-16 achieved 98.76 % accuracy and mAP = 0.9999 (see Table 7), and the Darknet-19 achieved 99.92 % accuracy and mAP = 1 (see Table 8).

As to speed, the RPN classifies frames with 4.08 fps, the VGG-16 classifies frames with 26.4 fps, and the Darknet-19 classifies frames with 28.3 fps (see Table 9).

The comparison results between the proposed end-to-end static HUI system and the static HUI system presented by [60] are demonstrated in Table 10. In which, both system

TABLE 8. Confusion matrix for Darknet-19 on the proposed one frame-gesture dataset, "Bd" and "Fd" refer to Backward and Forward respectively.

Class	Bd	CCW	CW	Down	Fd	Land	Left	Right	Up
Bd	1	0	0	0	0	0	0	0	0
CCW	0	0.9933	0	0	0	0	0	0	0
CW	0	0	1	0	0	0	0	0	0
Down	0	0	0	1	0	0	0	0	0
Fd	0	0	0	0	1	0	0	0	0
Land	0	0	0	0	0	1	0	0	0
Left	0	0	0	0	0	0	1	0	0
Right	0	0	0	0	0	0	0	1	0
Up	0	0	0	0	0	0	0	0	1

performance and speed have been increased up to 99 % and 28 fps respectively.

B. EXPERIMENTS ON DETECTION AND TRACKING

We used the public human gesture dataset [60] to train and test the Tiny-YOLOv2 [41] and the proposed developed model for detecting certain classes (face and hand). We divided this dataset into 3733 images for the training process and 2490 images for the testing process for both models. The detection results after testing the two models according to the standard object detection evaluation protocol PASCAL VOC [61] are demonstrated in Table 11. The two models almost achieved a similar mAP. However, the proposed model achieved higher efficiency by increasing its speed up to 120 fps. Therefore, we used this proposed developed Tiny-YOLOv2 as the deep detection network of the proposed dynamic HUI system.

Table 12 demonstrates the speed calculations of the proposed dynamic HUI system. The system achieved a speed up to 27.52 fps based on our proposed developed Tiny-YOLOv2 with the KF [42] tracker component and preserved almost the same detecting accuracy with mAP = 0.9089. While it achieved a speed of 17.26 fps based on the Tiny-YOLOv2 [41], and a speed of 23.37 fps based on our proposed developed Tiny-YOLOv2 without using the KF tracker.

Table 13 illustrates the comparison results between the proposed dynamic HUI system with three recent existing dynamic approaches. Multimodel methods were used to handle dynamic gestures in [5], [16]. In [16], Zhang *et al.* used the Kinect sensor to utilize depth information with RGB information in recognizing only six dynamic gestures of a person who wears a 3D shutter glass. In [5], Chen *et al.* used the key skeleton points with RGB information in recognizing also six dynamic gestures of a person who wears yellow gloves. While in [55], Monajjemi *et al.* used only RGB information in recognizing the dynamic gestures of an uninstrumented person but only three dynamic gestures were recognized. The proposed dynamic HUI system used only RGB information in recognizing ten dynamic gestures of an uninstrumented person without utilizing special sensors.

C. EXPERIMENTS ON REAL FLIGHTS

The proposed HUI systems were tested in real interacting scenarios by non-expert users to launch and control the bebop

TABLE 9. Comparison results of total accuracy, mAP, and efficiency for RPN, VGG-16, and Darknet-19 on the proposed one frame-gesture dataset, "Bd", "Fd", and "Ta" refer to Backward, Forward, and Total accuracy respectively.

Method	Bd	CCW	CW	Down	Fd	Land	Left	Right	Up	Ta	mAP	Speed (fps)
RPN	1	1	1	1	1	1	1	1	1	100 %	1	4.083
VGG-16	0.9925	09933	1	1	0.9594	1	1	1	0.9407	98.76 %	0.9999	26.43
Darknet-19	1	0.9933	1	1	1	1	1	1	1	99.92 %	1	28.39

TABLE 10. Comparison results between the proposed end-to-end static HUI and the static HUI system presented by [60], "DL" refer to Deep Learning.

HUI system	Gesture's type	Based on	Method	Detection or classification time (ms)	Gesture interpreting time (ms)	Total time (ms)	Total accuracy	Speed (fps)
Static HUI System [60]	Static	Detection with DL	Tiny-YOLOv2	11.36	36.25	47.61	90 %	21
Proposed Static HUI System	Static	Classification with DL	YOLOv2	35.21	No interpreting	35.21	99 %	28

TABLE 11. Comparison result on human gesture dataset with the speed on the ground computer.

Method	Face	Hand	mAP	Speed (fps)
Tiny-YOLOv2	0.909	0.909	0.909	88
Proposed model	0.9089	0.9086	0.9087	120

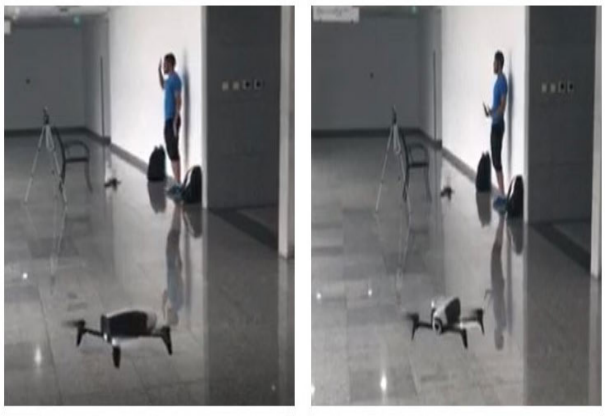


FIGURE 12. The proposed HUI systems being tested with Bebop 2 in indoor environment.

2 UAV in an environment without GPS (see Fig. 12) as presented by [47], [62]. We made indoor real flights in a room of these dimensions 6.5 m × 15 m × 3.5 m. A certain sequence of gestures was supposed as a test scenario. This sequence was Takeoff, Fly up, Fly forward, Fly left, Fly backward, Fly right, Yaw 180° CCW, Yaw 180° CW, and Land.

During the test, we collected the UAV's navigation data that were measured by its onboard navigation sensors. These collected navigation data were used to investigate the UAV's trajectory profile and its response to the proposed HUI systems. Fig. 13 shows the 3D plot of the x, y, and z navigation data for about 85 s of interacting with a non-expert user who performed the test scenario sequence.

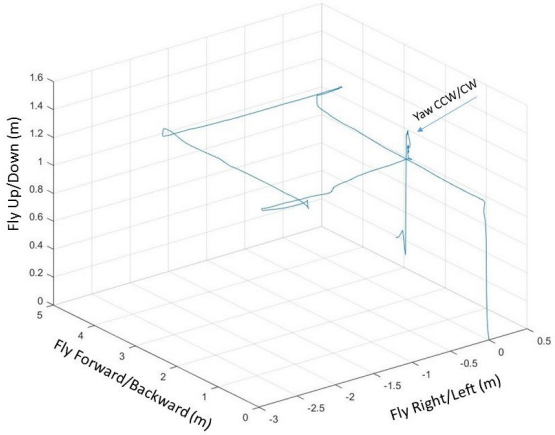


FIGURE 13. The 3D plot of the x, y, and z navigation data during a real flight test with a non-expert user.

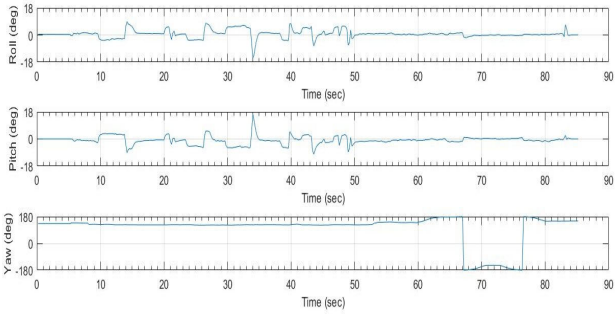


FIGURE 14. Illustration of the UAV attitude during the real flight test. Roll, Pitch, and Yaw are presented in degrees.

Fig. 14 shows the attitude of the UAV while performing the test scenario sequence. Yaw, pitch, and roll angles are measured by the UAV's onboard navigation sensors and transmitted to the ground computer via the wireless communication link. The pitch angle indicates the Fly forward / Fly backward commands, and the roll angle indicates the Fly right / Fly left commands.

TABLE 12. Comparison result demonstrates the performance and speed improvements for the dynamic HUI system.

Method	Dynamic gesture detection time (ms)	Tracking time (ms)	Dynamic gestures interpreting time (ms)	Total execution time (ms)	mAP	Speed (fps)
Tiny-YOLOv2	56.8181	No tracking	1.112	57.9301	0.909	17.2621
Proposed model	41.6666	No tracking	1.112	42.7786	0.9087	23.3761
Proposed model + KF	33.3333	1.879	1.112	36.3243	0.9089	27.5297

TABLE 13. Comparison results between the proposed dynamic HUI system with the existing dynamic approaches.

HMI system	Interacted machine	Used data	Based on DL	Number of gestures	Instrumented interacted person	Multimodel method	Special sensors	Real-time system type
HUI System [55]	UAV	RGB	No	3	No	No	No	Hard real-time system
H-3DTV Interaction System [16]	TV	Depth and RGB	No	6	Yes (3D Shutter glass)	Yes	Yes (Microsoft Kinect)	Soft real-time system
Intelligent HUI System [5]	UAV	Skeleton and RGB	Yes	6	Yes (Yellow gloves)	Yes	No	Hard real-time system
Proposed Dynamic HUI System	UAV	RGB	Yes	10	No	No	No	Hard real-time system

At time t = 67 s and 77 s, we can detect the changes in the yaw angle which correspond to the two commands of the Yaw 180° CCW and Yaw 180° CW respectively.

VII. CONCLUSION

Two novel GBI systems have been proposed to launch and control a UAV in real-time utilizing a monocular camera and a ground computer. The two proposed GBI systems can be modified to interact with a massive number of applications rather than the UAVs. Moreover, the interacted person is able to interact with the machine from another place as long as there is a communication link between the GBI ground computer and the interacted machine. The first proposal is an end-to-end static GBI system that has the advantage of discarding the gesture interpreting component to boost the system performance up to 99 % with a speed of 28 fps. On the other hand, the second proposal is a dynamic one which is preferred because it is more natural in human interaction without using the multimodel methods in dynamic gesture recognition. It adopts a simple model to detect three parts (face and two hands) of the interacted person and tracks them for a certain number of frames until a specific dynamic gesture is recognized. The proposed dynamic method is efficient next to decreasing the complexity and speeds the interaction up to 27 fps comparing with the recent multimodel ones. To propose these two systems, a new dataset and a developed simple model of the Tiny-YOLOv2 detection framework have been proposed. The proposed dataset comprises ten gesture classes that helps the researchers in GBI systems. The proposed developed model succeeded in saving some resources and increasing the deep neural network efficiency with achieving the same performance which can help in different deep learning approaches and can be utilized to work on the micro machines onboard computers due to its simplicity and efficiency.

ACKNOWLEDGMENT

The authors thank our colleagues Lakshantha Dissanayake and Dikshan who provided great help in collecting the proposed dataset and Dr. E. Safwat for his effort that greatly assisted the research.

REFERENCES

[1] P. A. Laplante, *Real-Time Systems Design and Analysis*. Hoboken, NJ, USA: Wiley, 2004.

[2] T. Sun, S. Nie, D.-Y. Yeung, and S. Shen, "Gesture-based piloting of an aerial robot using monocular vision," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5913–5920.

[3] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella, "Human control of uavs using face pose estimates and hand gestures," in *Proc. 9th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, 2014, pp. 1–2.

[4] M. A. Kassab, H. S. Taha, S. A. Shedied, and A. Maher, "A novel voting algorithm for redundant aircraft sensors," in *Proc. 11th World Congr. Intell. Control Autom.*, Jun. 2014, pp. 3741–3746.

[5] B. Chen, C. Hua, D. L. Y. He, and J. Han, "Intelligent Human–UAV interaction system with joint cross-validation over Action–Gesture recognition and scene understanding," *Appl. Sci.*, vol. 9, no. 16, p. 3277, Aug. 2019.

[6] J. Yang and H. Sun, "Control system, a method for controlling an uav, and a uav-kit," U.S. Patent 15 096 283, Oct. 20, 2016.

[7] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal, and S. Zhang, "Communication and networking of UAV-based systems: Classification and associated architectures," *J. Netw. Comput. Appl.*, vol. 84, pp. 93–108, Apr. 2017.

[8] L. Yang, Z. Liu, X. Wang, and Y. Xu, "An optimized image-based visual servo control for fixed-wing unmanned aerial vehicle target tracking with fixed camera," *IEEE Access*, vol. 7, pp. 68455–68468, 2019.

[9] C. Fu, Y. Zhang, Z. Huang, R. Duan, and Z. Xie, "Part-based background-aware tracking for uav with convolutional features," *IEEE access*, vol. 7, pp. 79997–80010, 2019.

[10] M. Mueller, G. Sharma, N. Smith, and B. Ghanem, "Persistent aerial tracking system for UAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1562–1569.

[11] X. Zhao, F. Pu, Z. Wang, H. Chen, and Z. Xu, "Detection, tracking, and geolocation of moving vehicle from uav using monocular camera," *IEEE Access*, vol. 7, pp. 101160–101170, 2019.

[12] J. Kwak and Y. Sung, "Autonomous uav flight control for gps-based navigation," *IEEE Access*, vol. 6, pp. 37947–37955, 2018.

[13] S. Kim, W. Lee, Y.-S. Park, H.-W. Lee, and Y.-T. Lee, "Forest fire monitoring system based on aerial image," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. Disaster Manage. (ICT-DM)*, Dec. 2016, pp. 1–6.

[14] L. Merino, F. Caballero, J. R. Martínez-de-Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *J. Intell. Robot. Syst.*, vol. 65, nos. 1–4, pp. 533–548, Jan. 2012.

[15] K. Cheng, N. Ye, R. Malekian, and R. Wang, "In-air gesture interaction: Real time hand posture recognition using passive rfid tags," *IEEE Access*, vol. 7, pp. 94460–94472, 2019.

[16] S. Zhang and S. Zhang, "A novel human-3dtv interaction system based on free hand gestures and a touch-based virtual interface," *IEEE Access*, vol. 7, pp. 165961–165973, 2019.

[17] C. Rognon, S. Mintchev, F. DellAgnola, A. Cherpillod, D. Atienza, and D. Floreano, "FlyJacket: An upper body soft exoskeleton for immersive drone control," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2362–2369, Jul. 2018.

[18] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: Detection, processing, classification and applications," *Biol. Procedures Online*, vol. 8, no. 1, pp. 11–35, Dec. 2006.

[19] V. Berezhnoy, D. Popov, I. Afanasyev, and N. Mavridis, "The hand-gesture-based control interface with wearable glove system," in *Proc. ICINCO*, 2018, pp. 458–465.

[20] G. Costante, E. Bellocchio, P. Valigi, and E. Ricci, "Personalizing vision-based gestural interfaces for HRI with UAVs: A transfer learning approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 3319–3326.

[21] C. Morato, K. N. Kaipa, B. Zhao, and S. K. Gupta, "Toward safe human robot collaboration by using multiple kinects based real-time human tracking," *J. Comput. Inf. Sci. Eng.*, vol. 14, no. 1, p. 151, Mar. 2014.

[22] J. Zhang, L. Peng, W. Feng, Z. Ju, and H. Liu, "Human-agv interaction: Real-time gesture detection using deep learning," in *Proc. Int. Conf. Intell. Robot. Appl.*, 2019, pp. 231–242.

[23] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at microsoft," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8604–8608.

[24] S. Pourmehr, V. M. Monajjemi, R. Vaughan, and G. Mori, "You two! Take off: Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 137–142.

[25] D. McNeill, *Hand Mind: What Gestures Reveal About Thought*. Chicago, IN, USA: University of Chicago Press, 1992.

[26] C. Hu, M. Q. Meng, P. X. Liu, and X. Wang, "Visual gesture recognition for human-machine interface of robot teleoperation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, vol. 2, 2003, pp. 1560–1565.

[27] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007.

[28] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2012.

[29] S. Saikia and S. Saharia, "A survey on vision-based dynamic gesture recognition," *Int. J. Comput. Appl.*, vol. 138, no. 1, p. 15, 2016.

[30] M. A. Goodrich and A. C. Schultz, *Human-Robot Interaction: A Surverys*. Boston, MA, USA: Now, 2008.

[31] C. Manresa, J. Varona, R. Mas, and F. J. Perales, "Hand tracking and gesture recognition for human-computer interaction," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 5, no. 3, pp. 96–104, Jul. 2005.

[32] M. Monajjemi, J. Bruce, S. A. Sadat, J. Wawerla, and R. Vaughan, "Uav, do you see me? Establishing mutual attention between an uninstrumented human and an outdoor uav in flight," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2015, pp. 3614–3620.

[33] G. Podevijn, R. O'Grady, Y. S. Nashed, and M. Dorigo, "Gesturing at subswarms: Towards direct human control of robot swarms," in *Proc. Conf. Towards Auto. Robot. Syst.*, 2013, pp. 390–403.

[34] Y. Wang, A. REN, M. Zhou, W. WANG, and X. YANG, "A novel detection and recognition method for continuous hand gesture using fmcw radar," *IEEE Access*, vol. 8, pp. 167264–167275, 2020.

[35] 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 networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4207–4215.

[36] C. C. D. Santos, J. L. A. Samatelo, and R. F. Vassallo, "Dynamic gesture recognition by using CNNs and star RGB: A temporal information condensation," *Neurocomputing*, vol. 400, pp. 238–254, Aug. 2020.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[41] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[42] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[43] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "ModDrop: Adaptive multi-modal gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1692–1706, Aug. 2016.

[44] S. Escalera, V. Athitsos, and I. Guyon, "Challenges in multi-modal gesture recognition," in *Gesture Recognition* (The Springer Series on Challenges in Machine Learning), S. Escalera, I. Guyon, and V. Athitsos, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-57021-1_1.

[45] J. L. Raheja, M. Minhas, D. Prashanth, T. Shah, and A. Chaudhary, "Robust gesture recognition using kinect: A comparison between DTW and HMM," *Optik*, vol. 126, nos. 11–12, pp. 1098–1104, Jun. 2015.

[46] M. Zhang, Z. Zhang, Y. Chang, E.-S. Aziz, S. Esche, and C. Chassapis, "Recent developments in game-based virtual reality educational laboratories using the microsoft kinect," *Int. J. Emerg. Technol. Learn.*, vol. 13, no. 1, pp. 138–159, 2018.

[47] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A kinect-based natural interface for quadrotor control," *Entertainment Comput.*, vol. 4, no. 3, pp. 179–186, Aug. 2013.

[48] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–9.

[49] R. Das and K. B. Shivakumar, "Augmented world: Real time gesture based image processing tool with intel real Sense technology," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 9, no. 1, pp. 63–84, Jan. 2016.

[50] B. Liao, J. Li, Z. Ju, and G. Ouyang, "Hand gesture recognition with generalized Hough transform and DC-CNN using realsense," in *Proc. 8th Int. Conf. Inf. Sci. Technol. (ICIST)*, Jun. 2018, pp. 84–90.

[51] R. Vilaça, J. Ramos, J. Sepúlveda, and J. S. Esteves, "Mobile platform motion control system based on human gestures," *Int. J. Mechatronics Appl. Mech.*, vol. 2017, no. 1, pp. 267–273, 2017.

[52] X. Wei, M. Wang, and L. Xiao, "Research of human gesture recognition algorithm based on multi-layer perceptron," in *Proc. Conf.*, 2017, Art. no. 020146.

[53] Y. Yu, X. Wang, Z. Zhong, and Y. Zhang, "ROS-based UAV control using hand gesture recognition," in *Proc. 29th Chin. Control Decision Conf. (CCDC)*, May 2017, pp. 6795–6799.

[54] A. Tellaeche, J. Kildal, and I. Maurtua, "A flexible system for gesture based human-robot interaction," *Procedia CIRP*, vol. 72, pp. 57–62, Dec. 2018.

[55] M. Monajjemi, S. Mohaimenianpour, and R. Vaughan, "UAV, come to me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4410–4417.

[56] J. Nagi, H. Ngo, L. M. Gambardella, and G. A. Di Caro, "Wisdom of the swarm for cooperative decision-making in human-swarm interaction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1802–1808.

[57] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[59] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system," in *Proc. ICRA*, Kobe, Japan, 2009, vol. 3, nos. 3–2, p. 5.

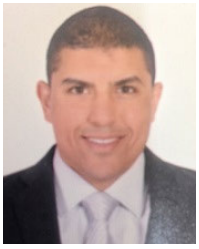
[60] A. Maher, C. Li, H. Hu, and B. Zhang, "Realtime human-UAV interaction using deep learning," in *Biometric Recognition* (Lecture Notes in Computer Science), vol. 10568, J. Zhou *et al.*, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-69923-3_55.
[61] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[62] A. Mashood, H. Noura, I. Jawhar, and N. Mohamed, "A gesture based kinect for quadrotor control," in *Proc. Int. Conf. Inf. Commun. Technol. Res. (ICTRC)*, May 2015, pp. 298–301.



MOHAMED A. KASSAB is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His current research interests include object detection, object tracking, and gesture recognition.



MOSTAFA AHMED is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His current research interests include object detection and gesture recognition.



ALI MAHER received the Ph.D. degree in pattern recognition and intelligent systems from Beihang University, Beijing, China, in 2018.



BAOCHANG ZHANG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, 2001, and 2006, respectively. From 2006 to 2008, he was a Research Fellow with The Chinese University of Hong Kong, Hong Kong, and Griffith University, Brisbane, QLD, Australia. He is currently an Associate Professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. He is also with the Shenzhen Academy of Aerospace Technology, Shenzhen, China. He was supported by the Program for New Century Excellent Talents in University, Ministry of Education, China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.