Fully Autonomous UAV-based Action Recognition System Using Aerial Imagery

Peng, Han, Razi, Abolfazl

Abstract—Human action recognition is an important topic in artificial intelligence with a wide range of applications including surveillance systems, search-and-rescue operations, human-computer interaction, etc. However, most of the current action recognition systems utilize videos captured by stationary cameras. Another emerging technology is the use of unmanned ground and aerial vehicles (UAV/UGV) for different tasks such as transportation, smart agriculture, traffic control, border patrolling, wildlife monitoring, etc. This technology has become more popular in recent years due to its affordability, high maneuverability, and limited human interventions. However, there does not exist an efficient action recognition algorithm for UAV-based monitoring platforms.

This paper considers UAV-based video action recognition by addressing the key issues of aerial imaging systems such as camera motion and vibration, low resolution, and tiny human size. In particular, we propose an automated deep learning-based action recognition system which includes the three stages of video stabilization using the SURF feature selection and Lucas-Kanade method, human action area detection using faster region-based convolutional neural networks (R-CNN), and action recognition. We propose a novel structure that extends and modifies the InceptionResNet-v2 architecture by combining a 3D CNN architecture and a residual network for action recognition. We achieve an average accuracy of 85.83% for the entire-video-level recognition when applying our algorithm to the popular UCF-ARG aerial imaging dataset. This accuracy significantly improves upon the state-of-the-art accuracy by a margin of 17%.

Index Terms—Drone video, human detection, action recognition, deep learning, unmanned aerial systems.

I. Introduction

Video-based action recognition, an integral part of a class of AI platforms, is typically designed for capturing human behaviors and using them in support of decision making systems [1], [2]. The applications of video-based action recognition span a wide range including security system [3], [4], entertainment [5], visual surveillance [6], virtual reality (VR) [7], athletes training [8], human-computer interaction [9], smart healthcare [10], etc. Most traditional video-based action recognition systems use stationary ground cameras to collect video information.

Recently, the use of unmanned aerial vehicles (UAVs) has become commonplace in many applications; and video-based action recognition systems is not an exception. In most of aerial monitoring systems, autonomous UAVs make on-the-fly decisions by processing the captured imagery. For instance, most commercial UAVs nowadays are equipped with the collision avoidance feature and change their motion paths when encountering an obstacle [11], [12].

Researchers have already tried to develop human action recognition systems using stationary cameras in various fields such as sports [13], surveillance [6], unmanned vehicles [14], etc. However, the UAV-based human action recognition has not yet received the deserved attention from the research community. Implementing UAV-based human action recognition systems can revolutionize the current practice in many applications, since drones provide several advantages over the ground-based monitoring systems. Some of the advantages include flexible and faster access, on-demand video streaming with adjustable resolution, focus and angle of view, and less human intervention and lower risk in harsh and extreme environments, only to name a few. That is the main reason that the UAV-based monitoring systems are experiencing an exponential growth in recent years [12], [15]–[19].

Recognizing human actions by processing captured video frames using static platforms is known to be a challenging task due to its computational complexity, the intrinsic variability between the actions of the same class, challenges related to determining the start and end points of each action, and dealing with mixed actions, and complications of background removal. This task becomes even more challenging when applied to UAV-based monitoring systems due to facing additional problems such as motion-related blurriness, camera vibration, varying angles of view, and tiny object sizes.

This paper proposes an end-to-end system for UAV-based action recognition by solving the aforementioned issues. More specifically, we use non-overlapping 16-frame video segments for clip-based action classification purpose. In this regard, we extract labeled 16-frame video segments from the benchmark UCF-ARG dataset [20] to develop a training dataset for action recognition. Likewise, we use clip-level classification along with majority voting for the ultimate video-level action recognition.

The contribution of the proposed works is two-fold: i) we proposed a fully autonomous UAV-based human action recognition system that enables the UAVs to precisely detect and recognize human actions while accommodating aerial imaging artifacts; ii) we introduced a novel architecture for neural networks that combines a 3D convolutional network with a residual network, which substantially improves the performance of the 3D CNNs [21]. More specifically, our proposed method when applied to the UCF-ARG dataset, achieves the classification success rate (CSR) of 73.72% for the clip-level 5-class action recognition problem. This translates to entire-video-level accuracy of 85.83% which shows a substantial improvement over the current state of the art methods with 68% accuracy [16].

The rest of this paper is organized as follows: In section 2, related works are discussed. Section 3 elaborates on the details of the proposed end-to-end action recognition algorithms using aerial videos. Section 4 demonstrates the efficiency of the proposed method by providing comparative results followed by concluding remarks in section 5.

II. Related Works

Action recognition algorithms, based on their utilized training and interpretation methods, can be divided into two main categories, namely *conventional* and *deep learning* (DL) methods.

Conventional methods typically include 3 main sequential steps of feature extraction, feature representation, and action classification. The feature extraction step is the process of extracting key information or indicators from video frames, which can represent an action. Two main approaches of feature extraction include global and local feature extraction methods. In global feature extraction, the shape of a moving object is considered as a holistic part, and the global features are extracted through object localization, background tracking, and the *region of interest* (ROI) encoding. Two famous implementations of the global feature extraction methods include *motion energy images* (MEI) and *motion history image* (MHI) [22], where in the former, the human motion is captured by accumulating the contrast of the pixel values between the human object and the background during an action into one image, while in the latter, the temporal information is also captured by weighting the pixel values based on their time [23]. The main advantage of these methods is the convenience of reducing video-based analysis into a much simpler problem of image classification. The local feature extraction methods typically include sequential steps of i) detecting the local spatio-temporal interest points first, ii) calculating the local patches around these points, and iii) using the identified local patches as representative features. There exist three main local feature extraction methods including *space-time interest points* [24], *cuboid* [25], and *dense sampling* [26]. Feature representation is used to represent and describe the extracted features in a unified way that is normalized, distinguishable, robust, and invariant to background clutter, scale, and rotation. This process is also called feature encoding. The two popular feature representation methods include *bag of vision word* (BoV) [27] and *fisher vector* [28]. Once the features are extracted and encoded, the last stage of conventional action recognition algorithms is the action classification, which is performed using standard classification methods including *support vector machine* (SVM), *logistic regression*, and *K nearest neighbors* (KNN). The *improved dense trajectories* (IDT) [26] which combines dense sampling with the BoV method can be considered as the state-of-the-art among traditional action recognition methods.

In recent years, deep learning methods have become more popular in many regression and classification tasks, due to their superior performance in capturing intricate relations through stacked hidden layers, generalizability, affordable computation costs with GPUs, and eliminating the need for handcrafted feature extraction methods [29]–[32].

There exist three mainstream methods in using DL for video-based action recognition. The first approach is using two-stream methods [33], where two separate CNN structures are used to extract temporal and spatial features from the video, and then the results are integrated to classify the action. The second approach is 3D convolutional neural networks (*C3D networks*) [21] by considering videos as 3D input, where the time is the 3rd dimension. In this approach, a 3D convolutional neural network is used to process videos with no pre-processing. The third mainstream approach is using the *long short-term recurrent convolutional neural network* (LRCN) [34], where a CNN architecture is used to extract spatial features from the image sequences and output fixed-length vectors, and then a *long short term memory* (LSTM) is used to learn from the sequenced information.

There are some recent and ongoing researches that achieve even higher action recognition accuracies building upon these works, including (i) *fusion stream* [35] which develops two-stream method by inserting multiple fusion layers into both spatial and temporal streams instead of fusing the results at the last step, (ii) *hidden two stream* [36], (iii) spatio-temporal residual networks (ST-ResNet) [37] which adds residual connections into both spatial and temporal domains so it can captures partial-temporal information in both streams separately, (iv) *temporal segment networks* (TSN) [38], (v) *pseudo 3D CovNet* [39] which uses the idea of decoupling 3D CovNet into two parts: a 2D spatial convolution filter to extract spatial information and a 1D temporal convolution filter to extract temporal information and then uses P3D to replace the residual unit in ResNet, (vi) *temporal 3D ConvNets* (T3D) [40], and (vii) *two-stream Inflated 3D* (I3D) *I3D* [41]. However, these methods perform reasonably well for stationary platforms but do not accommodate the key requirements of aerial imaging when subjected to shaking, vibration, and varying angle of views.

The above-mentioned methods follow the dominant trend of DL-based methods in using raw video frames as their input without any preprocessing. In a different line of research, human action can also be captured by monitoring the human skeleton pose. To mimic the skeleton motion, it is sufficient to trace the position of joints (key points) in consecutive frames, which is the core idea behind the skeleton-based action recognition algorithms [42]–[44]. This approach substantially reduces the dimensionality of the input data and the computation complexity of the processing method by converting video frames into motion trajectories of key points. However, it has its own challenges, such as the need for accurate human detection and skeleton extraction methods that becomes troublesome in the presence of complex backgrounds, multiple subjects, etc. In particular, it becomes prohibitively challenging for UAV-based video streaming when the human body size is tiny and the joints overlap due to the improper angle of view. Although human action recognition and UAV-based monitoring systems are both well-studied topics, there are very few works that bridge these two distinct research areas to develop an end-to-end solution for efficient UAV-based action recognition. Ghazal et al. [15] employed SVM algorithm to classify actions based on the high-level and stationary features extracted from key video frames using a CNN architecture. They also extract conceptual features from the first and last video frames for the human detection part. However, they use their own dataset which is not available for comparison. Burghouts et al. [17] proposed a focus-of-attention mechanism to perform the human action recognition that includes tracking, human detection and a per-track analysis. However, their method with its basic configuration achieves only 57% accuracy for the UCF-ARG dataset when the entire video is utilized. Hazar et al [16] used a two-phase method along with the scene stabilization algorithm. Their method involves human detection through human vs nonhuman modeling as well as the human action modeling, where the modeling part is performed offline and the recognition part is performed on the fly in the inference phase. However, this method is not good enough since it achieves only 68% accuracy for the UCF-ARG dataset. Another work that uses UCF-ARG dataset is [45]. Their focus is to enhance the action recognition accuracy noting the difficulty of collecting aerial images by adding two additional sources of videos including (i) video games, and (ii) generated fake aerial images using conditional Wasserstein generative adversarial networks. They achieve the average classification accuracy of 67.9% using the aerial images collected by the authors called YouTube-Aerial dataset. However, their performance drops to 35.92% when classifying 10 actions using the UCF-ARF dataset.

As mentioned earlier, the majority of action recognition methods are suitable for stationary cameras and fail in solving the issues of aerial imaging such as video instability, small object size, and varying angle of view. The few recent attempts for developing UAV-based human action recognition systems have considered this problem and produced some interesting results, but their performance is still far from satisfactory highlighting the need for more efficient methods. In this paper, we propose a fully autonomous human action recognition system to boost the performance of the action recognition accuracy for aerial videos using video stabilization, human detection, and deep neural networks.

III. Methods

Our end-to-end solution includes three steps of (i) video stabilization, (ii) human detection, and (iii) human action recognition, while taking necessary considerations to solve challenges associated with UAV-based video streaming (see Figure 1). Due to the flight instability and motion dynamics of commercial UAVs, aerial videos typically suffer from issues like vibration and camera motion. Therefore, using video stabilization is a critical need to obtain a stabled video appropriate for further processing. Furthermore, videos captured by drones usually include large areas that contain no information and the targets of interests (humans in this context) are tiny and barely noticeable. The human detection stage should be powerful enough to exclude unnecessary regions and video segments in order to focus only on the areas with recognizable human objects. In the following sections, we elucidate the details of each step.

A. Video stabilization

We use a frame-by-frame method for video stabilization. The overall idea is to extract key points from one frame and then finding the corresponding nodes in the following frame that exhibit consistent spatial shift, and this continues until we reach the last frame. Then we quantify the averaged frame by frame motions in terms of 3D trajectories represented by a transformation. The obtained cumulative trajectory is smoothed out to represent the actual motions, while the difference between the original and smoothed trajectories are considered video instability and is used to eliminate the vibration from the video by proper shifting. The following are the details of each step.

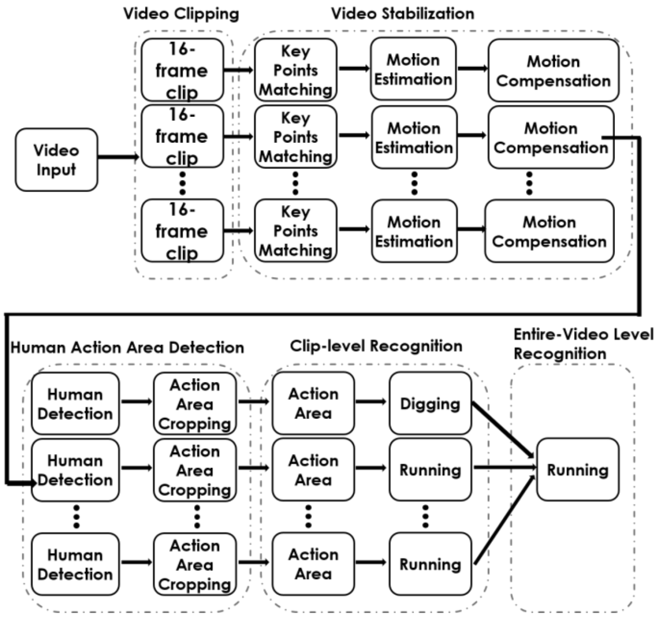


Fig. (1) Conceptual block diagram of the proposed UAVbased human action recognition.

1) Keypoints extraction using the speeded up robust features (SURF): We first use the SURF method to extract key points from the frames. SURF [46] is a low-complexity image feature detection method that is an accelerated version of the *scaleinvariant feature transform* (SIFT) [47] to extract local image descriptors. Using a light-weight algorithm such as SURF is highly desirable for one-the-fly feature extraction by the drones.

2) Track keypoints to next frame: In the next step, we use the *Lucas-Kanade optical flow* [48] to find the mapping between keypoints among the consecutive frames and quantify their relative motions. This method assumes that the displacement of the image content between two adjacent moments (frames) is small and approximately constant near the point p under consideration. Therefore, the optical flow equation can be applied to all pixels in a window centered at p and the local image flow (velocity) vector (V_x,V_y) can be written in a matrix form $Av = b$, where we have

$$A = \begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{bmatrix}, v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, b = \begin{bmatrix} -I_t(p_1) \\ -I_t(p_2) \\ \vdots \\ -I_t(p_n) \end{bmatrix}, \quad (1)$$

and $p_1,p_2,...,p_n$ are the pixels inside the window, and $I_x(p_i),I_y(p_i),I_t(p_i)$ denote the partial derivatives of the image I with respect to position x, y and time t, evaluated at point p_i at the current time.

This over-determined system which has more equations than unknowns, can be solved using the least squares method by

$$\mathrm{v} = (A^TA)^{-1}A^Tb \quad (2)$$

which provides the following results for the averaged relative 2D motions between the two consecutive frames:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(p_i)^2 & \sum_i I_x(p_i)I_y(p_i) \\ \sum_i I_y(p_i)I_x(p_i) & \sum_i I_y(p_i)^2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\sum_i I_x(p_i)I_t(p_i) \\ -\sum_i I_y(p_i)I_t(p_i) \end{bmatrix} \quad (3)$$

where the summations are over n points $p_1,p_2,...,p_n$.

3) Motion Estimation: Suppose that F_i, and F_{i+1} are the two adjacent frames, and $P_i = \{p_1^i, p_2^i, \cdots, p_n^i\}$ and $P_{i+1} = \{p_1^{i+1}, p_2^{i+1}, \cdots, p_n^{i+1}\}$ are the set of matched points between frames F_i and F_{i+1}, respectively, and T is the transformation matrix between the two frames. Then, we can state:

$$\begin{bmatrix} \mathbf{x}_j^{i+1} \\ y_j^{i+1} \\ 1 \end{bmatrix} = T_i \begin{bmatrix} x_j^i \\ y_j^i \\ 1 \end{bmatrix} \quad (4)$$

$$T_i = \begin{bmatrix} S_i \cdot \cos\Delta\theta_i & -S_i \cdot \sin\Delta\theta_i & \Delta x_i \\ S_i \cdot \sin\Delta\theta_i & S_i \cdot \cos\Delta\theta_i & \Delta y_i \\ 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where (x^i_j,y_j^i) is the coordinate of point p^i_j, $\Delta\theta_i$ is the rotation angle, S_i is the scale factor, and Δx_i and Δy_i are the translation motion vectors in horizontal and vertical directions between frames F_i, and F_{i+1}. No motion in the 3rd dimension is assumed ($z_i = 1$). Here, we use the cumulative sum of Δx_i, Δy_i, and $\Delta\theta_i$ to produce the motion trajectories specified by $x_i = \sum_{k=1}^i \Delta x_k, y_i = \sum_{k=1}^i \Delta y_k, \theta_i = \sum_{k=1}^i \Delta\theta_k$.

4) Trajectory Smoothing: Finally, we use a *Hanning* window to smooth out the obtained cumulative motion trajectories. Hanning window has no side lobes and is defined as:

$$w[n] = 0.5\left[1 - \cos\left(\frac{2\pi n}{N}\right)\right] = \sin^2\left(\frac{\pi n}{N}\right), \quad (6)$$

where N is the window size. The following algorithm summarizes the stabilization algorithm.

Algorithm 1: Video stabilization Using a method based on SURF and Lucas-Kanade optical flow.

```
Input: Unstable video Output:
Stabilized video
Initialization:
1. Read the first frame as PreviousIMG; 2. detect
the keypoint using SURF as PreviousPts;
3. Set i = 1; set $nF$ = the number of frames.
4. while $i$ 6= $nF$ do
    5. Set $i \leftarrow i + 1$
    6. Read the $i_{th}$ frame as CurrentIMG;
    7. Track and match PreviousPts from PreviousIMGto
       obtain CurrentPts in CurrentIMG using Lucas-Kanade
       optical flow;
    8. Calculate $T_i$, the $i_{th}$ transformation matrix between
       PreviousPts and CurrentPts using equation (5);
    9. Find keypoints in the $i^{th}$ frame using SURF as
         PreviousPts;
   end
10. Compute trajectory using cumulative sum
    oftransformations as trajectory;
11. Smooth out the trajectory using convolutioncalculation
    using Hanning window;
12. Calculate the difference between the original and
    the smoothed trajectories and apply the difference to
    transformation matrix;
13. Apply the new transformation matrix to eachframes and
    generate stable video;
```

14. Return stabilized video.

B. Human action area detection

Since the video taken by a UAV contains lots of irrelevant information, to achieve higher accuracies it is desirable to focus on areas where the humans are located. In deep learning, there are two main types of object detection methods: onestage methods and two-stage methods. The two-stage methods firstly identify a large pool of candidate regions which may or may not contain the object and then use a classification method (e.g. CNN) to classify these regions of interest (ROI) to verify if there exists an object or not. Two-stage methods require a longer time but achieve better results. The most successful implementations of the two-stage methods use fast R-CNN [49] and faster R-CNN [50]. The one-stage methods (e.g., single shot multi-box detector (SSD) [51] and you only look once (YOLOV3) [52],) use a similar approach, but skip the region proposal stage and execute the detection process directly over a dense sampling of possible locations. This approach requires only a single pass through the neural network and predicts all the bounding boxes at the same time. Thus the one-stage methods can realize a faster detection speed but with a lower accuracy. In our problem, the detection accuracy is considered more important than the execution speed, therefore a two-stage method is more suitable. Therefore, we applied a two-stage method based the faster R-CNN for human detection.

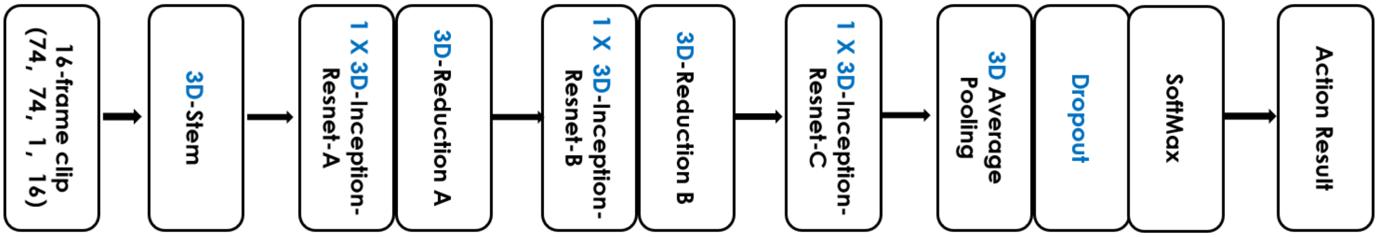

16-frame clip (74, 74, 1, 16)
3D-Stem
1 X 3D-Inception-Resnet-A
3D-Reduction A
1 X 3D-Inception-Resnet-B
3D-Reduction B
1 X 3D-Inception-Resnet-C
3D Average Pooling
Dropout
SoftMax
Action Result

Fig. (2) The architecture of the modified 3D InceptionResNet-v2 networks. The modified blocks and parameters are shown with blue color.

C. Action recognition

The last stage of the proposed end-to-end system deals with the action recognition. Inspired by the success of *InceptionResNet-v2* [53] in image classification, we developed a new architecture for action recognition by modifying and extending the Inception-ResNet-v2 method to a 3D version, which called Inception-ResNet-3D. Inception-ResNet-v2 is a combination of two recent networks, namely the *residual connections* [54] and the *Inception architecture* [55]. The Inception-ResNet3D network architecture is shown in Fig 2, which extends the original 2D architecture to a 3D version by using 3D convolutional neural network, and reducing the filter sizes while maintaining the overall architecture (i.e., the number and the order of the layers, pooling approaches, activation functions, etc.) unchanged.

Note that the Inception-ResNet-V2 has three main parts including (i) stem block, (ii) Inception-ResNet A,B,C blocks, and (iii) reduction A,B,C blocks. In the 3D stem block of the revised version (Figure 2), all kernels with dimensions (h,w) are extended to 3D kernels of size (h,w,d), where the 3r dimensional d equals to the first two dimensions, i.e. $h = w = d$. Also, the number of output filters in the convolution layers are divided by 8. For instance, the number of filters 32 is replaced with 4 in the first layer of the 3D stem block.

The interior module of the network includes 3D-InceptionResnet-A, 3D-Inception-Resnet-B, and 3D-Inception-Resnet-C blocks as shown in Figure 3. Instead of using 5 InceptionResnet-A, 10 Inception-Resnet-B, and 5 Inception-Resnet-C for each block, we use only 1 of each type per block to reduce the parameter size.

For the 3D reduction A and B blocks, all the kernels of size (h,w) are extended to 3D versions (h,w,d). Unlike the Inception Resnet A,B,C blocks, here the size of the 3rd dimension here is $d = 1$. The number of filters for each block is reduced by 8. Finally, the average pooling size is $(2,2,2)$ in our 3D-Inception-ResNet network.

IV. Experiments

In this section, the performance of the proposed method for fully autonomous UAV-based human activity recognition is assessed. We first describe the used dataset, and then provide a quantitative analysis of the proposed approach along with comparisons with the state of the art methods.

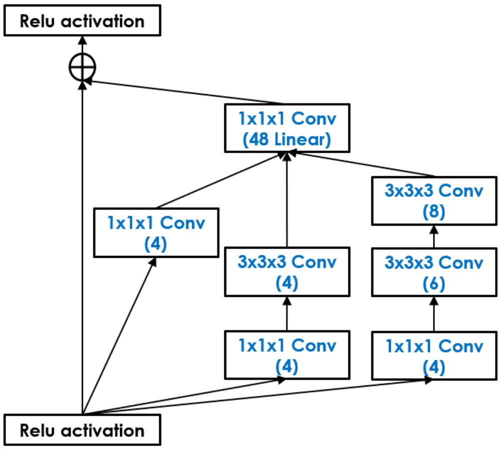
A. Dataset Description

UCF-ARG dataset is a benchmark dataset for action recognition which includes a set of hard-to-distinguish tasks based on videos taken by stationary ground and mobile aerial cameras [20]. More specifically, this dataset includes actions performed by 12 actors recorded by a ground camera, a rooftop camera at a height of 100 feet, and a UAV-mounted camera. Here, we only use the aerial videos. This dataset contains 10 human action classes: *boxing, carrying, clapping, digging, jogging, open-close trunk, running, throwing, walking,* and *waving*. Except for the *open-close trunk*, all other actions are performed 4 times by each actor in different directions, while the *open-close trunk* is performed only 3 times by each actor. Therefore, we have 48 videos for each action (36 for the actions of type *open-close trunk*). Since most of the former research projects on human detection and human action recognition, focused on 5 classes: *digging, running, throwing, walking,* and *waving* [16], [17], [56], we choose the same set of classes for a fair and meaningful comparison. Also, we have shorter videos for the *running* class which we compensate for it by using data augmentation as explained later.

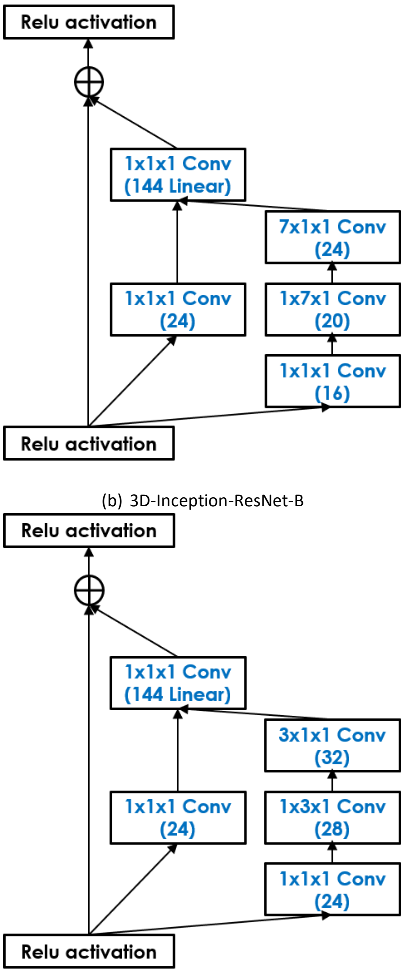
B. Implementation Details

1) Video stabilization: For the SURF feature point detection method, we used the python cv2 package with the hessian Threshold 500. For the *Lucas-Kanade optical flow* method, we use the iterative Lucas-Kanade algorithm with pyramids, following the implementation details presented in [57]. Since the video clips include only 16 frames, we wouldn't expect that clips include multiple camera motion episodes. Therefore, we set the *Hanning* window size to 16, which is equal to the number of frames to realize a global stabilization rather than local short-term stabilization.

2) Human detection: A pre-trained faster R-CNN model with implementation details of a version previously applied to the COCO dataset [58] is used for human detection. We consider that a clip contains a valid action, if a human subject is detected in at least one frame. Otherwise, the clip is not used for the further processing of action detection. The size and the width-height ratio of the *rectangular* box containing a human subject can be different from one frame to another. To avoid the issue of bias to the object size that can reduce the performance of the human action recognition, we extend the detected *rectangular* box to the smallest *square* box that encompasses all the identified *rectangular* boxes.This concept is visualized in Fig. 4. We crop the video frames to include only the extracted boxes for enhanced accuracy in the action recognition. Therefore, the action recognition stage that is applied to the reproduced 16-frame video clips is scale-invariant and rotation-invariant with respect to the human objects. If more than one humans are detected, we choose the one closer to the center of the frame.



(a) 3D-Inception-ResNet-A



(b) 3D-Inception-ResNet-B

(c) 3D-Inception-ResNet-C

Fig. (3) The schematic for interior grid modules of the Inception-ResNet-3D network.

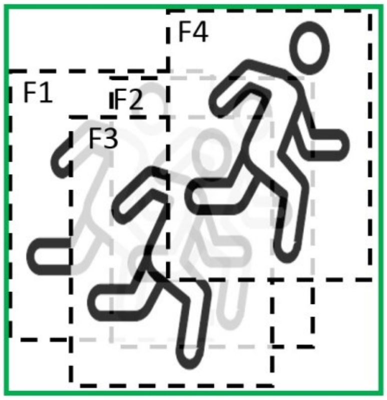


Fig. (4) Human detection: the smallest square box containing the identified rectangular boxes in frames F1 to F4 are used to cop the video frames for the subsequent action recognition stage.

3) Data augmentation: Since the data size is limited and noting the imbalanced number of recordings for different classes (e.g., the number of video clips for the *running* classes is about $1/3$ of other classes because of shorter videos), we implemented data augmentation before applying the Inception-ResNet-3D network. More specifically, to solve the imbalanced number of samples, we produced two extra sets of video clips for the *running* class by adding videos with altered blurriness and sharpness effects. Likewise, to increase the number of samples, we produce two new video clips by horizontal flipping and adding Gaussian Noise with zero mean and unit variance to the original video frames. Consequently, the number of training samples increased by a factor of 6 for the *running* class and doubled for other classes.

4) Human action recognition: To realize a uniform input size, we resized the video frames into (74 ,74) since the detected human boxes typically fit into 74 pixel by 74 pixel image segments. For a fair comparison with other works [16], [17], a *leave-one-out cross-validation* (LOO CV) is used to split the dataset into training and test sets, where each test is corresponding to one person. The dropout rate of 0.5 is used in our method. The popular *Adam* optimizer is used with the learning rate of 0.001 and *decay* = 0. The network is trained for the clip-level samples. The entire-video-level action recognition is based on applying the majority voting to the obtained clip-level labels.

C. Result

Fig 5 presents the video stabilization stage for two exemplary actions: *digging* and *throwing*. The top row in each sub-figure shows the original frames while the bottom row shows the stabilized frames. The green lines points to a fixed point close to the human object in the video frames. We can see that the human moves considerably with respect to the intersections of the green lines from one frame to another in the original video, while remaining stationary in the stabilized video frames. This indicates that the video stabilization part is successful in eliminating the camera vibration and motion effects. The figures illustrate that a small vibration in the UAV camera can translate to a large shift of the human object in the entire video which can be multiple times of a human size. Since this effect is not clearly visible in the original videos, we show the zoomed-in version of the *throwing* action. The shift in the human object location can cause a severe performance degradation in the action classification if not properly addressed. Therefore, video stabilization is a necessary step of the proposed method for aerial imaging.

Table I presents the action recognition accuracy of the proposed method as well as the two state of the art methods [16], [17] which use the same dataset (UCF-ARG). The results are provided per test (i.e. a sample recording) for the proposed method and Hazar et al. [16], but only the average accuracy is available for Burghouts et al [17]. The results confirm that the proposed method improves upon both methods with a significant margin. If we consider the clip-level results, our method achieves an average accuracy of 68% which is significantly higher than the Burghouts et al [17] method with accuracy 57% and the Hazar et al. [16] method with accuracy 68%. The achieved gain is even higher if we consider the ultimate result of the entire-video-level action recognition which achieves the high accuracy of 85.83%. It significantly improves the state of the art results by more than 17% increase in the action recognition accuracy. It is noteworthy that our method also outperforms [45] that achieves an accuracy of 35.92%. However, this comparison is not fair since they use 10 action classes (unlike 5 classes for our method), and also a different training/test splitting method.

Similar results are presented in Table II in terms of average action recognition accuracy per class for the three methods. Again the achieved gain is higher for the entire-video-level classification as expected. Our method consistently outperforms the competitor methods with a significant margin. The only exception is the *running class*, where our method with accuracy 89.58% slightly under-performs the [17] method with an accuracy of 91%. Perhaps a better video augmentation method can help improve the accuracy for this class. The higher gain is achieved for the *waving* class that improves the best method by 33.58%.

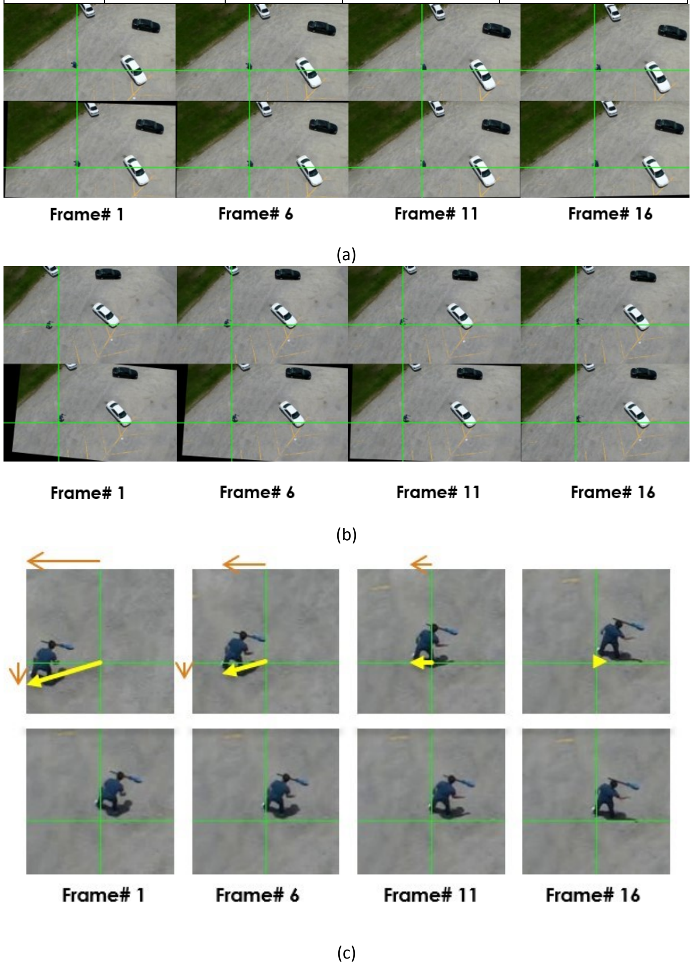
Fig 6 shows the confusion matrix for all classes, where we can see that the *running, walking,* and *waving* actions are well classified with the accuracy above 90%. However, the *digging* and *throwing* actions relatively lower accuracies of 81% and 73%. The potential reason is that these two classes may have some similar action components in the utilized 16-frame video clips.

V. Conclusion and discussion

Noting the lack of an effective action recognition algorithm which is capable of accommodating specific requirements of aerial monitoring systems, in this paper, we proposed an endto-end system for UAV-based action recognition. The proposed method includes three stages of video stabilization, action area detection, and action recognition, where the video stabilization solves the camera motion and vibration issue, and the action area detection deals with the small human sizes in aerial images. Also, the classification algorithm is trained for top-view

TABLE (I) Comparison of the performance of the proposed method with the state of the art in terms of action recognition accuracy per test video.

Test set	Burghouts et al. [17]	Hazar et al. [16]	Our Method	
			16-frame level	Entire-Video level
1		65%	68.7%	75%
2		55%	71.5%	80%
3		75%	77.5%	90%
4		55%	71.7%	80%
5		85%	75.9%	95%
6		35%	78.2%	80%
7		60%	68.2%	85%
8		70%	72.5%	90%
9		60%	66.2%	75%
10		85%	77.4%	95%
11		75%	82.1%	90%
12		90%	82.7%	95%
average	57%	68%	73.72%	85.83%



(a)

(b)

(c)

Fig. (5) Two examples of video stabilization result for actions: (a) *digging*, (b) *throwing* (original size), (c) *throwing* (zoomed-in view). In each sub-figure, the top row shows sample frames of the original 16-frame clip, and the bottom row shows the stabilized video clip. The relative human position to a fixed point (represented by the intersection of green lines) varies from one frame to another in the original video clip, while remaining constant in the stabilized video.

images, where action recognition is more challenging than the front-view and side-view. Our experiment results show that our algorithm achieves a very high accuracy of 85.83% when applied to the benchmark UCF-ARG dataset. This accuracy is significantly higher than the previously reported accuracy of 68% (by a margin of 17%), therefore it is appropriate for aerial monitoring systems for action recognition tasks.

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976 – 990, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885609002704

[2] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in timesequential images using hidden markov model," in *Proceedings 1992 IEEE Computer Society conference on computer vision and pattern recognition*. IEEE, 1992, pp. 379–385.

[3] S. Han, M. Achar, S. Lee, and F. Pena-Mora, "Empirical assessment of a˜ rgb-d sensor on motion capture and action recognition for construction worker monitoring," *Visualization in Engineering*, vol. 1, no. 1, p. 6, 2013.

TABLE (II) Comparison of the performance of the proposed method with the state of the art in terms of action recognition accuracy per class.

Test set	Burghouts et al. [17]	Hazar et al. [16]	Our Method	
			16-frame level	Entire-Video level
digging	50%	79%	68.91%	81.25%
running	91%	67%	77.39%	89.58%
throwing	33%	69%	62.57%	72.92%
walking	75%	67%	83.83%	95.83%
waving	33%	56%	75.91%	89.58%
average	57%	68%	73.72%	85.83%

[4] A. Bobick and J. Davis, “Real-time recognition of activity using temporal templates,” in *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV’96*. IEEE, 1996, pp. 39–42.

[5] V. Kruger, D. Kragic, A. Ude, and C. Geib, “The meaning of action: A¨ review on action recognition and mapping,” *Advanced robotics*, vol. 21, no. 13, pp. 1473–1501, 2007.

[6] S. Danafar and N. Gheissari, “Action recognition for surveillance applications using optic flow and svm,” in *Asian Conference on Computer Vision*. Springer, 2007, pp. 457–466.

[7] E. A. Suma, B. Lange, A. S. Rizzo, D. M. Krum, and M. Bolas, “Faast: The flexible action and articulated skeleton toolkit,” in *2011 IEEE Virtual Reality Conference*. IEEE, 2011, pp. 247–248.

[8] W.-L. Lu and J. J. Little, “Simultaneous tracking and action recognition using the pca-hog descriptor,” in *The 3rd Canadian Conference on Computer and Robot Vision (CRV’06)*. IEEE, 2006, pp. 6–6.

[9] S. S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: a survey,” *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[10] A. Jalal, S. Kamal, and D. Kim, “A depth video sensor-based lifelogging human activity recognition system for elderly care in smart indoor environments,” *Sensors*, vol. 14, no. 7, pp. 11735–11759, 2014.

[11] S. Islam and A. Razi, “A path planning algorithm for collective monitoring using autonomous drones,” in *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2019, pp. 1–6.

[12] S. Islam, Q. Huang, F. Afghah, P. Fule, and A. Razi, “Fire frontline monitoring by enabling uav-based virtual reality with adaptive imaging rate,” in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 368–372.

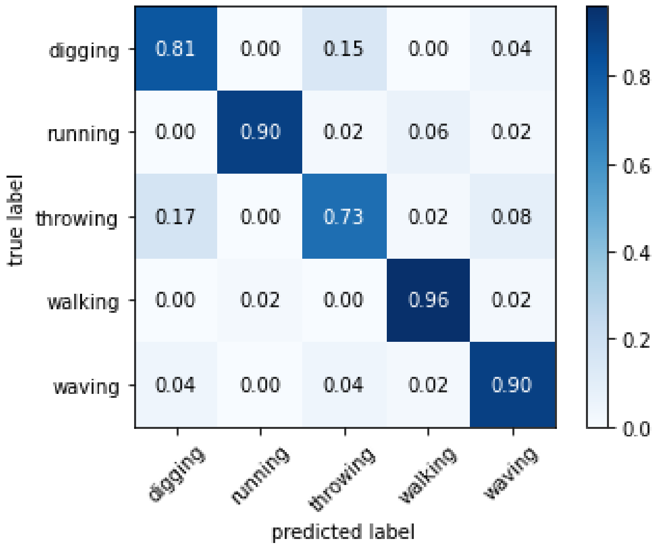
[13] K. Soomro and A. R. Zamir, “Action recognition in realistic sports videos,” in *Computer vision in sports*. Springer, 2014, pp. 181–208.

[14] E. Frazzoli, M. A. Dahleh, and E. Feron, “Real-time motion planning for agile autonomous vehicles,” *Journal of guidance, control, and dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[15] G. Shamsipour and S. Pirasteh, “Artificial intelligence and convolutional neural network for recognition of human interaction by video from drone,” 2019.

[16] H. Mliki, F. Bouhlel, and M. Hammami, “Human activity recognition from uav-captured video sequences,” *Pattern Recognition*, vol. 100, p. 107140, 2020.

[17] G. Burghouts, A. van Eekeren, and J. Dijk, “Focus-of-attention for human activity recognition from uavs,” in *Electro-Optical and Infrared Systems: Technology and Applications XI*, vol. 9249. International Society for Optics and Photonics, 2014, p. 92490T.



	digging	running	throwing	walking	waving
digging	0.81	0.00	0.15	0.00	0.04
running	0.00	0.90	0.02	0.06	0.02
throwing	0.17	0.00	0.73	0.02	0.08
walking	0.00	0.02	0.00	0.96	0.02
waving	0.04	0.00	0.04	0.02	0.90

Fig. (6) The confusion matrix of the multi-level action classification for the proposed method.

[18] A. Razi, “Optimal measurement policy for linear measurement systems with applications to uav network topology prediction,” *IEEE Transactions on Vehicular Technology*, 2019.

[19] H. Peng, A. Razi, F. Afghah, and J. Ashdown, “A unified framework for joint mobility prediction and object profiling of drones in uav networks,” *Journal of Communications and Networks*, vol. 20, no. 5, pp. 434–442, 2018.

[20] M. S. A. Nagendran, D. Harper. (2010) Ucf-arg dataset, university of central florida. [Online]. Available: (http://crcv.ucf.edu/data/UCFARG.php

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[22] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 3, pp. 257–267, 2001.

[23] R. V. Babu and K. Ramakrishnan, “Compressed domain human motion recognition using motion history information,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03).*, vol. 3. IEEE, 2003, pp. III–41.

[24] I. Laptev, “On space-time interest points,” *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[25] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition´ via sparse spatio-temporal features,” in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. IEEE, 2005, pp. 65–72.

[26] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 3551–3558.

[27] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, June 2006, pp. 2169–2178.

[28] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[29] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[30] ——, “State-of-the-art in artificial neural network applications: A survey,” *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[31] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[32] D. R. Nayak, A. Mahapatra, and P. Mishra, “A survey on rainfall prediction using artificial neural network,” *International Journal of Computer Applications*, vol. 72, no. 16, 2013.

[33] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 568–576. [Online]. Available: http://papers.nips.cc/paper/5353-twostream-convolutional-networks-for-action-recognition-in-videos.pdf

[34] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[35] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[36] Y. Zhu, Z. Lan, S. Newsam, and A. Hauptmann, “Hidden two-stream convolutional networks for action recognition,” 04 2017.

[37] C. Feichtenhofer, A. Pinz, and R. P. Wildes, “Spatiotemporal multiplier networks for video action recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4768–4777.

[38] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” vol. 9912, 10 2016.

[39] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3d residual networks,” in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[40] A. Diba, M. Fayyaz, V. Sharma, A. Karami, M. Arzani, L. Van Gool, and R. Yousefzadeh, “Temporal 3d convnets: New architecture and transfer learning for video classification,” 11 2017.

[41] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” 07 2017, pp. 4724–4733.

[42] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12026–12035.

[43] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, “An attention enhanced graph convolutional lstm network for skeleton-based action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1227–1236.

[44] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Skeleton-based action recognition with directed graph neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921.

[45] W. Sultani and M. Shah, “Human action recognition in drone videos using a few aerial training examples,” *arXiv preprint arXiv:1910.10027*, 2019.

[46] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[47] D. G. Lowe *et al.*, “Object recognition from local scale-invariant features.” in *iccv*, vol. 99, no. 2, 1999, pp. 1150–1157.

[48] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision,” 1981.

[49] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[50] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[52] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[53] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[56] N. AlDahoul, M. Sabri, A. Qalid, and A. M. Mansoor, “Real-time human detection for aerial captured video sequences via deep models,” *Computational intelligence and neuroscience*, vol. 2018, 2018.

[57] J.-Y. Bouguet *et al.*, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm.”

[58] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in´ context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.