Dogfight: Detecting Drones from Drones Videos

Muhammad Waseem Ashraf[1], Waqas Sultani[1], Mubarak Shah[2]
[1]Intelligent Machines Lab, Information Technology University, Pakistan
[2]Center for Research in Computer Vision, University of Central Florida, USA
mohammadwaseem043@gmail.com, waqas.sultani@itu.edu.pk, shah@crcv.ucf.edu

Abstract

As airborne vehicles are becoming more autonomous and ubiquitous, it has become vital to develop the capability to detect the objects in their surroundings. This paper attempts to address the problem of drones detection from other flying drones. The erratic movement of the source and target drones, small size, arbitrary shape, large intensity variations, and occlusion make this problem quite challenging. In this scenario, region-proposal based methods are not able to capture sufficient discriminative foreground-background information. Also, due to the extremely small size and complex motion of the source and target drones, feature aggregation based methods are unable to perform well. To handle this, instead of using region-proposal based methods, we propose to use a two-stage segmentation-based approach employing spatio-temporal attention cues. During the first stage, given the overlapping frame regions, detailed contextual information is captured over convolution feature maps using pyramid pooling. After that pixel and channel-wise attention is enforced on the feature maps to ensure accurate drone localization. In the second stage, first stage detections are verified and new probable drone locations are explored. To discover new drone locations, motion boundaries are used. This is followed by tracking candidate drone detections for a few frames, cuboid formation, extraction of the 3D convolution feature map, and drones detection within each cuboid. The proposed approach is evaluated on two publicly available drone detection datasets and outperforms several competitive baselines.

1. Introduction

Drones are actively being used in several daily life applications such as agriculture [8], wildfire fighting [34], inventory applications [19], cinematography [16] and surveillance [47, 28]. Due to a large-scale application of drones, recently computer vision researchers have put forward several new techniques for object detection [10], tracking [28], agriculture monitoring [8] and human action recognition [1, 2] in the imagery obtained through drones. In addition to detecting different objects from a drone video, it is also important to detect the drone itself from a video captured by another drone to avoid drone attacks [3], drone collisions [33] and safe multi-drone flights [24, 32].

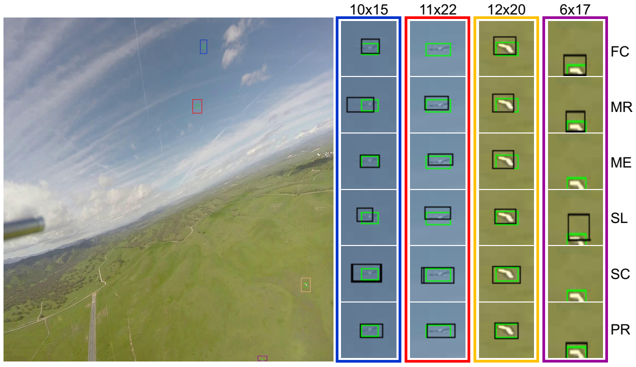

Figure 1. A comparison of our approach (PR) with state-of-the-art object detectors: FCOS (FC) [36], Mask-RCNN (MR) [13], MEGA (ME) [7], SLSA (SL) [39], and SCRDet (SC) [42]. In this frame (1080×1920), there are four drones of sizes: 10×15, 11×22, 12×20, 6×17. The green bounding box represents the ground truth and output of detectors are shown in black colors. For the clarity, we have not shown all false positives. The proposed approach provides better drone localization, reduces false positives and improves recall.

Detection of ground objects and aerial drones from drone videos is a very challenging problem due to a large and abrupt camera motion, arbitrary drone shape and view changes, occlusion, and more importantly small object size. Although a lot of recent research has been conducted to detect and track ground objects and to detect human action using drones [10, 47, 28, 1, 2, 35], limited work is being done to detect drones from drone videos [24, 33]. To tackle the problem of drone detection, Li et al., [24] proposed a new drone to drone detection dataset and employed handcrafted features for background estimation and foreground moving objects detection. Similarly, Rozantsev et al., [33] introduced a new challenging dataset of drones and air-crafts. They employed regression-based approaches to achieve object-centric stabilization and perform cuboid classification for detection purposes.

Usually flying drones occupy a few pixels in the video frames. For instance, the average drone size respectively is 0.05% and 0.07% of the average frame size in drone detection datasets proposed in [24] and [33]. Note that this is much smaller than that of PASCAL VOC (22.62%) and ImageNet (19.94%). Small objects including drones usually appear in the cluttered background and are oriented in different directions which makes its detection quite difficult. This issue was also pointed out by Huang et al., in [17], where they demonstrated that the mean Average Precision (mAP) of small objects is much lower than that of larger objects. Furthermore, the object detection performance further worsens in the videos [7]. To address this, Noh et al., [29] proposed a feature-level super-resolution-based approach that utilizes high-resolution target features for supervising a low-resolution model. However, this would require the availability of both low and high-resolution drone images which are difficult to obtain in drone videos where the drone is already flying at a far distance. Similarly, Yang et al., [42] employed a region proposal-based multi-level feature fusion approach to detect small objects and introduce a new loss function to handle rotated objects. However, due to very small size objects, less salient, and cluttered backgrounds i.e., clouds, buildings, etc., it is difficult to obtain well-localized region proposals, specifically in drone detection datasets. Through adversarial learning, Wu et al., [40] proposed to learn domain-specific features employing metadata (flying altitudes, weather, and view angles). Given the recent low prices of drones, it is more useful to use an RGB camera for detection and collision avoidance purposes instead of relying on the expensive hardware for metadata collection. Authors in [48, 27, 39] proposed to use convolution feature aggregations across video frames to achieve improved video object detection. Our experimental results and analysis reveal that although feature aggregations [48, 27, 39] techniques work well for large video object detection, for drone detection explicit motion information is more useful.

In this paper, we propose a two-stage segmentation-based approach to detect drones in cluttered backgrounds. The first stage uses only the appearance cues while the second stage exploits spatio-temporal cues. Given a video frame, we divide it into overlapping frame regions. Each frame region is passed through deep residual networks [14] to obtain the convolution feature maps which are then followed by pyramid pooling layers [44] to embed the contextual information. After that pixel-wise and channel-wise attention is employed on the convolution feature maps to discriminate drone boundaries from the background and achieve improved drone localization. The purpose of the second stage is to discover missing detections, remove false detections, and confirm the true positive detections by employing motion information. To discover the missing drones, we employ motion boundaries to find probable drone locations. Given the detections from the first stage and motion boundaries locations, we track each location forward and backward for a few (eight) frames. After that, cuboids are extracted across those tracks and are fed to a 3D convolutional neural network [5] for spatio-temporal feature extraction. This is followed by pyramid pooling layers. Similar to the first stage, we employ pixel and channel-wise attention in the second stage as well to get the improved localization. The proposed approach significantly outperforms several competitive baselines. In the experimental section, we validate the efficacy of each step of the proposed approach. The rest of the paper is organized as follows. Section 2 provides a brief overview of the related developments in small object detection including drones in video and images. Section 3 deals with our proposed methodology and Section 4 covers experimental results. Finally, Section 5 concludes the paper.

2. Related work

Object Detection: Recent years have witnessed tremendous strides in improving object recognition accuracy [13, 30, 31, 26, 4, 22] on complex benchmark datasets. Law et al. [22] proposed a single-stage network where they detect an object bounding box as a pair of key points. Also, the authors introduced a novel corner pooling scheme for better corners localization. To address the foreground-background class imbalance in object detection datasets, Lin et al. [26] presented a modified cross-entropy loss that reduces the weights of the loss assigned to well-classified examples. Similarly, Li et al., [25] pointed out the underperformance of object detectors for long-tail distribution datasets. They proposed a novel balanced group-softmax to modulate the training process and ensure that the classifier is sufficiently trained for all the classes. Cai et al., [4] put forward an idea of multi-stage object detection where a sequence of detectors trained with increasing intersection over union (IoU) thresholds to achieve improved object localization. To further improve object detection accuracy, authors in [12] introduced a new feature fusion technique through multi-scale features summation in a feature pyramid. Although impressive detection results, most of these detectors have low detection and localization accuracy for small objects.

Small Object Detection: To address the problem of small, cluttered, and oriented objects, Li et al., [23] proposed to use generative adversarial networks to decrease the gap of feature representation of small and large objects. To further improve feature representation for small objects, authors in [29] proposed to jointly use the low and high-resolution feature maps of the same images during the training by match-

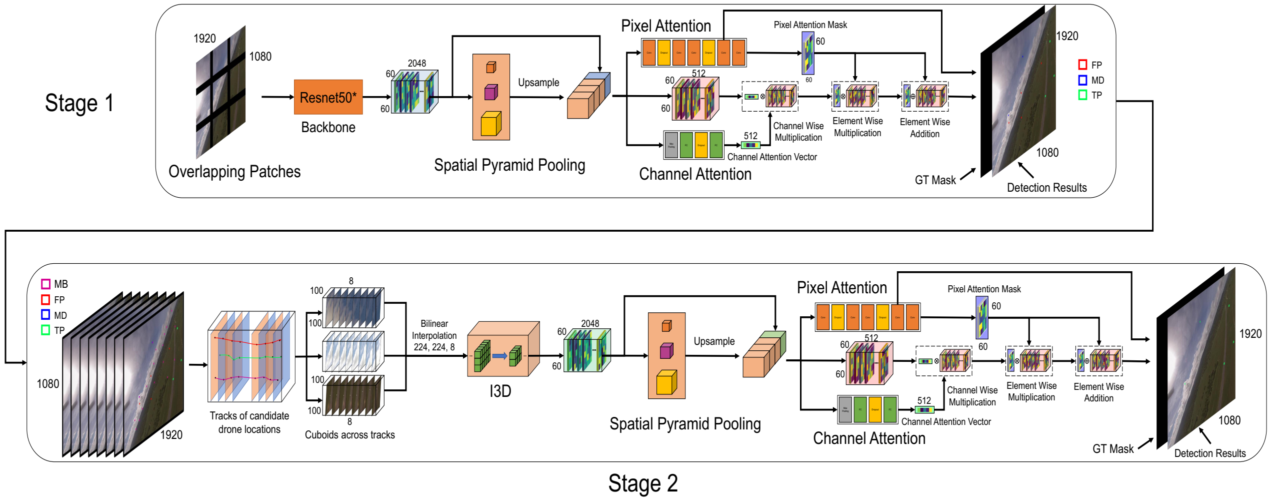

Figure 2. Our pipeline is divided into two stages. Stage-1 extracts Resnet50* features from the overlapping regions of each frame followed by pyramid pooling to retain global and local contextual information. Channel-wise and pixel-wise attention help in learning better localization of drones. Resnet50* refers to the modifications that we have applied (ref Section 3.1). Stage-2 combines spatial information with temporal data of the videos. Detections from stage-1 along with candidate regions discovered using motion boundaries are used as candidate regions where UAV can exist. All the proposals are tracked for 8 frames in a forward and backward manner to generate cuboids of size 224×224×8. Each cuboid is passed through the I3D network followed by the attention network to accurately locate drones within each cuboid. In figure MD, TP, FP, and MB corresponds to missed detection, true positive, false positive, motion boundaries respectively.

ing the relative receptive fields between the pairs. Yang et al., [42] proposed a multi-layer feature fusion technique along with attention networks to achieve improved object detection for cluttered and rotated objects. However, most of these methods are region proposal based which are unable to capture sufficient foreground-background information specifically when the objects are extremely small (such as 0.05% or 0.07% of the average image size as in the case of drone detection). A related problem to ours is the detection of ground objects from drone images such as [41] which proposed to detect clustered objects in aerial images. However, while detecting drones from a drone, target objects (drones) undergo fast, abrupt, and highly unpredictable movements as compared to objects (car, human) on the ground. Target objects can have arbitrary changing shapes due to their rotation around any axis as compared to the restricted movements of ground objects. Furthermore, as compared to ground objects, drones have much smaller sizes, and the target objects appear/disappear more frequently due to clouds.

Video Object Detection: Historically, taking a page from object detection, several video-based object detectors were introduced, including [48, 27, 39, 7]. The object detection accuracy degrades in the video due to several factors including motion blur, occlusion, and out of focus, etc,. To address this Zhu et al., [48] proposed to include the feature representation of each frame by aggregation of nearby frames features using optical flow. Authors in [27] improve feature aggregation speed using the Bottleneck-LSTM layer. Instead of aggregating features from nearby frames, Wu et al., [39] proposed to use whole sequence-level features aggregation for improved video object detection. Recently, Chen et al., [7] utilized both the global semantic and local localization information employing memory aggregation network and have shown impressive results on several video object detection benchmarks. Although impressive results, most of these methods are tested on standard video object detection datasets where the object covers a significant portion of the video frame. Furthermore, motion is much more complex in drone videos as compared to standard video object detection datasets.

Drones Detection: Recent low prices of drones will bring more and more UAVs to the sky. To keep the prices and weight of UAVs small, it is useful to detect other flying drones using simple RGB cameras instead of expensive radar systems. Therefore researchers have addressed the problem of drone detection for different applications. After obtaining a large number of spatio-temporal (s-t) tubes at different spatial resolutions, Rozantsev et al., [33] employs two CNN models to obtain coarse and fine motion stabilization in each s-t tube respectively. Next, they obtain UAVs detection by classifying each s-t tube using a third CNN. There are several differences between our method and [33]. 1) Instead of using computationally expensive region-based sliding windows at several scales, we employ an efficient fully convolutional segmentation-based approach, 2) as compared to [33], our approach does not need perfectly drone-centered cuboids and learn rich spatio-temporal information using I3D, and 3) we employ attention networks to improve feature representation for improved localization. Similarly, authors in [24, 43] detected moving drones by subtracting background images and then identified UAVs using deeply learned classifiers. Furthermore, they use Kalman filtering to get improved detection. Their method uses a lot of parameters and thresholds which make it irreproducible. Also, their reliance on background subtraction for moving object detection produces a lot of false positives. Instead of using RGB images, some researchers have also proposed to employ depth maps to achieve 3D localization [6, 38]. However, depth maps are quite expensive to obtain in real-world scenarios and add the extra payload to UAVs.

3. Proposed method

Our goal is to detect and localize drones in the video frames which are captured by other flying drones. Our proposed approach to tackle this challenging problem is based on the three observations: (1) due to very small drone size, region proposal based method may not be able to capture enough discriminative foreground-background information, therefore the bottom-up segmentation based approach which classifies each pixel is preferable; (2) the model should learn subtle visual differences between a drone and the background (clouds, etc); (3) due to large abrupt motion of target and source drones, the feature aggregation methods may not be sufficient and we need to use explicitly optical flow information as has been successfully employed in several action recognition works [11]. In the following, we first discuss details of the segmentation network (Sec. 3.1), followed by the attention networks used to get improved localization (Sec 3.2). Finally, we discuss how we use motion information to discover missing detection and thus improve recall.

3.1. Stage-1: Exploiting Spatial Cues

We start with appearance-based pixel classification to accurately localize drones. For spatial feature computation, we resort to deep residual networks [14]. However, given the extremely small size of drones (0.05% or 0.07% of image size), obtaining good discriminative features by utilizing the whole images is not possible. The standard 2D CNN networks such as Resnet50 require a fixed size input image (473×473). Therefore image resizing from high-resolution image to low resolution (1080×1920 to 473×473) for feature computation can further decrease the spatial resolution of the drone to one or two pixels. Secondly, as the network goes deeper, we lose local information. To address this, we use two steps: 1) To avoid resizing the image, we divide each frame into overlapping regions, 2) We modify Resnet50 to keep local information intact as we go deep in the network. Specifically, we extract features from all four blocks of Resnet50 [14] and concatenate them together after spatial resizing of the first block to avoid dimension mismatch. Finally, we use 1×1 convolution to get back the original dimension. We call modified Resnet50 as Resnet50*. Inspired by the use of pyramid pooling [45] in several applications we employ pyramid pooling in our framework. Specifically, after obtaining features from Resnet50*, we apply the pyramid pooling using four different kernel sizes and concatenate those multi-scale features after up-sampling.

In experiments, we observe that although the above network provides decent drone detection, in several cases, it is unable to accurately detect and localize the drones. Therefore, to make feature maps more focused on the foreground, we use pixel and channel-wise attentions. Pixel and channel-wise attention networks are described in the next section.

3.2. Attention networks

Assuming the drone of size 16×11 (the average drone size in [24]), by missing only a few pixels from both sides, the intersection over union (IOU) drops to below 0.5. Therefore, it is crucial to get accurate localization for true drone detection. To achieve this, we introduce detailed pixel-wise and channel-wise attention on the convolution feature maps. Recently, several attention networks [37, 15, 42, 9] have been introduced for different computer vision applications.

Channel-wise attention: Inspired by [15], we use a channel-wise attention network to automatically learn to give more weights to informative feature channels and suppress less informative ones. The architectural details of channel-wise attention networks are given in Figure 3 (a). This attention is achieved through channel-wise multiplication of attention vector with convolution feature maps.

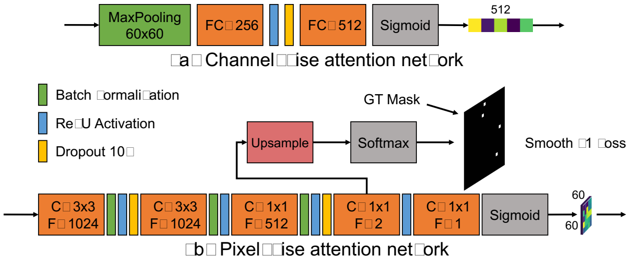

Figure 3. Architectural details of (a) channel-wise and (b) pixel-wise attention network, where ‘FC’ represents fully connected layer with number of units in (a), ‘C’ and ‘F’ represent convolution and number of filters respectively in (b).

Pixel-wise attention: Similar to channel-wise attention vector, we generate pixel-wise attention matrix to assign

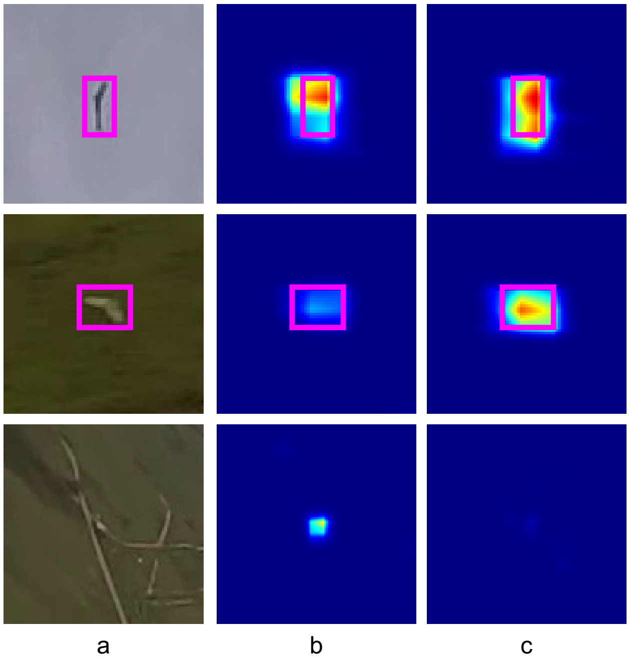

Figure 4. Role of attention. (a) Input image. (b) without and (c) with attention networks. Top two rows shows the examples where attention networks helps the network to learn to give more weights to the pixels associated with drones and the last row represents the case where attention network suppress the non-drone pixels.

more weights to spatial location which corresponds to drones and less weight to non-drone regions similar to [9, 42]. The architectural details of the pixel attention network are given in Figure 3 (b). To suppress background information, we perform element-wise multiplication of pixel attention mask with all convolution maps channels. This is followed by the addition of attention masks to give high weights to regions containing useful information.

In the experiments, we observe that attention networks significantly help in achieving better drone localization. Note that the complete stage-1 is trained end-to-end where attention is learned automatically through network architecture, training data, and losses. Figure 4 demonstrates the difference in the network outputs which training with and without attention networks.

3.3. Losses

Drone detection datasets have two major challenges: There exists a large drone versus non-drone class imbalance i.e., the majority of pixels belong to the background and only a few pixels (if any) occupy the drone. Secondly, due to extremely small size drones, the only difference of even 1 or 2 pixels between the detected box and ground truth brings down the IoU score to less than 0.5. Therefore, we use multiple losses to train our network. Specifically, to address the class imbalance, focal loss [26] is used. and Distance-IOU loss [46] is employed to achieve better IOU localization. Distance-IOU not only minimizes the IOU between the ground truth and detected bounding boxes but also reduces the distance between centers of two boxes. Finally, we use smooth-L1 loss [31] to jointly train pixel-wise attention network as shown in Figure 3 (b).

3.4. Stage2: Exploiting Spatiotemporal Cues

The purpose of this stage is to confirm true detections, reject the false detections and discover the missing detections of the stage-1. To find out the new probable drone locations, we use motion gradients which is explained below.

Motion boundaries: Drone can be characterized by locations undergoing motion. However, since drone detection datasets involve moving camera, simple optical flow magnitude cannot be much useful. Therefore, we propose to use optical flow gradients to capture the change in motion. Specifically, given every three frames of a video, we first stabilize them using key-points detection and then compute forward and backward optical flow. After that maximum motion gradients across all three frames are computed as follows:

$$G = max(\sqrt{u_x^2 + u_y^2}, \sqrt{v_x^2 + v_y^2}), \tag{1}$$

$$M = max(G_{0\to1}, G_{1\to2}, G_{2\to1}, G_{1\to0}), \tag{2}$$

where u_x, v_x, u_y, and v_y are optical flow gradients along x and y axis respectively, M show motion boundaries and $G_{0\to1}, G_{1\to2}, G_{2\to1}, G_{1\to0}$ represents motion gradients between frames, 0→1, 1→2, 2→1, 1→0 respectively.

There are two limitations of motion boundaries: The motion boundaries provide high magnitude across the boundaries of drones and in most cases, do not completely cover drones. Secondly, due to the underlying approximation of optical flow calculation, usually, the maximum of optical flow gradient magnitude does not match exactly with the moving drone. To address these issues, we dilate the motion boundaries and then apply conditional random field [20] to get the better localization of candidate drone regions.

Cuboids formation: Given detections from stage-1 and newly discovered locations obtained using motion boundaries, our next step is to extract spatio-temporal features from all candidate drone locations. To this end, we initialize the correlation tracker at each candidate location (including stage-1 detections and newly discovered locations). Due to small drone and complex camera motions, trajectories tend to drift away from their initial location within a few frames, therefore, we restrict trajectory length to eight

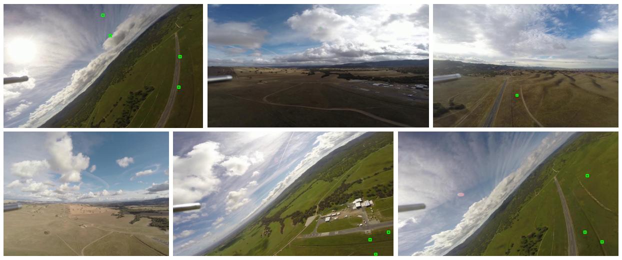

Figure 5. Samples frames from NPS-drone [24] dataset. The green boxes enclose drones.

frames. Specifically, given the candidate drone location, tracking is done three frames forward and four frames backward. Note that tracking is done after the motion stabilization of the corresponding eight frames. To capture contextual information across candidate location and to compensate for the trajectory drifting, N × N patches are extracted from video frames across each track, resulting in a cuboid of size N×N×8. Finally, to extract spatio-temporal features from each cuboid, we employ the Inflated-3D (I3D) network [5]. We choose I3D due to its fast speed, small memory consumption, and excellent capability of capturing detailed spatio-temporal characteristics. To make the size of the cuboid consistent with that of the standard I3D network input dimensions, we use bilinear interpolation on each patch to resize the cuboid from N×N×8 to 224×224×8. 3D Convolution features are extracted from the third last layer of the I3D network which has dimensions of 14×14×480. To have consistency with the stage-1, we use bilinear interpolation to resize the feature maps to 60 ×60×480. This is followed by 2D convolution layers to convert 60×60×480 to 60×60×2048 feature maps. Experimentally, we have also tried patch superresolution and feature map superresolution instead of resizing using bilinear interpolation, however, we did not observe any performance improvement.

Finally, spatio-temporal convolution feature maps of each cuboid are aggregated across different scales using spatial pyramid pooling. This is followed by attention networks and network losses as discussed in Section 3.2

4. Experiments

The main goal of our experiments is to verify that the proposed approach accurately localize drone in moving camera videos. To this end, we perform extensive experiments on two challenging drone datasets and compare our approach with several competitive approaches, and analyze different components of the proposed approach.

4.1. Implementation details

In this section, we provide implementation details of our approach. During stage-1, we divide each frame into nine overlapping patches for NPS-Drones dataset [24] and into four overlapping patches for FL-Drones dataset [33]. For both datasets, stage-1 is trained end to end from scratch, while for stage-2, a pre-trained I3D network with frozen weights is used for feature extractions. For the NPS-Drones dataset, a fixed 100×100 patch size is extracted across the drone to make a cuboid while in the FL-Drones dataset size of the patch corresponds to the size of the drone. In all the experiments, we use Adam [18] optimizer with an initial learning rate of 0.001 without the decay parameter.

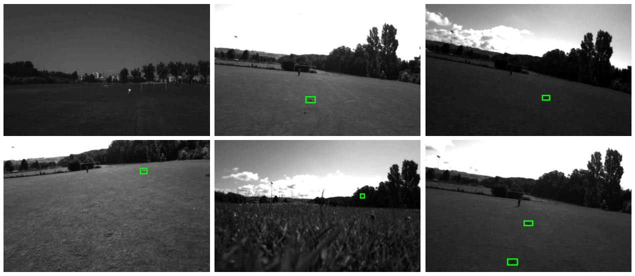

Figure 6. Samples frames from FL-drone [33] dataset. The green boxes enclose drones.

A simple correlational tracker is employed during cuboid formation. We remove candidate locations (provided by motion boundaries), which are smaller or bigger than the maximum and minimum size of drones in the training data respectively. To improve training, we have also used hard-negative mining. In a post-processing step, we remove detections that only appear for a single frame.

Evaluation metrics: Following the related works, we evaluate the performance of our approach and baselines using precision, recall, F1-score, and average precision (AP) where every frame is treated as an individual image for evaluation. We evaluate all the methods on every 4th frame of the testing data.

4.2. Datasets

We evaluate our approach on two drones datasets. Each of them is briefly introduced below.

NPS-Drones [24]: This dataset is published by Naval Postgraduate School (NPS) and is publicly available[1]. The dataset contains 50 videos that were recorded at HD resolutions (1920×1080 and 1280×760) using GoPro-3 cameras mounted on a custom delta-wing airframe. The minimum, average and maximum size of drones are 10 × 8, 16.2 × 11.6, and 65 × 21, respectively. The total number of frames in the dataset are 70250. In the experiments, the first 40 videos are used for training and validation and the last 10 videos are used for the testing.

FL-Drones [33]: The second drone dataset used in the paper was introduced by Rozantsev et al. [33]. This dataset

[1] https://engineering.purdue.edu/~bouman/UAV_Dataset/

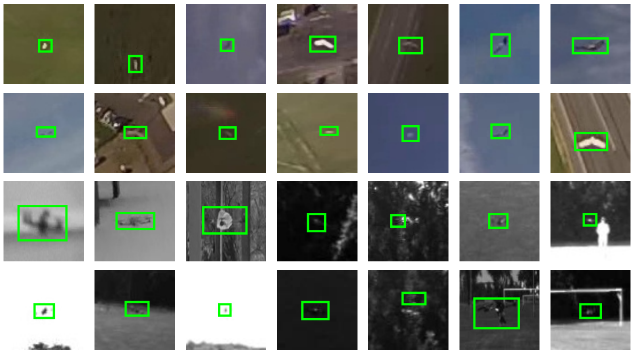

Figure 7. This figure shows the variability of drone shape and size in two datasets: NPS-drone [24] (first two rows) and FL-drone [33] (last two rows) datasets. The green boxes present the ground truth bounding boxes.

is quite challenging due to extreme illumination, pose, and size changes. This dataset contains indoor and outdoor samples and the flying drones have variable shapes, and their shape is barely retained even in consecutive frames. Drones get mixed up in the background due to small size and intense lighting conditions coupled with complex background. The minimum, average, and maximum sizes of drones are: 9×9, 25.5×16.4, and 259×197, respectively and frames resolutions are 640×480 and 752×480. This dataset contains 14 videos with a total of 38948 frames. As suggested by the authors [33], half of the data is used for the training, and the other half is used for the testing.

Sample frames and drones from both datasets are shown in Figure 5, Figure 6 and, Figure 7. Note that there are several frames in both datasets without any drone. It is interesting to know that the majority of original annotations released by the authors of [33, 24] are not precise and bounding boxes are much bigger than that of actual drones. To address this, we have re-annotated both datasets again. Given that number of frames in both datasets are more than 100K, re-annotation took significant time. The examples of improved annotations are shown in Figure 8. These improved annotations will be released.

Method	Precision	Recall	F1 score	AP
SCRDet-H [42]	0.81	0.74	0.77	0.65
SCRDet-R [42]	0.79	0.71	0.75	0.61
FCOS [36]	0.88	0.84	0.86	0.83
Mask-RCNN [13]	0.66	**0.91**	0.76	**0.89**
MEGA [7]	0.88	0.82	0.85	0.83
SLSA [39]	0.47	0.67	0.55	0.46
Proposed	**0.92**	**0.91**	**0.92**	**0.89**

Table 1. Quantitative comparison of the proposed approach with several state of the art approaches on the NPS dataset [24].

.

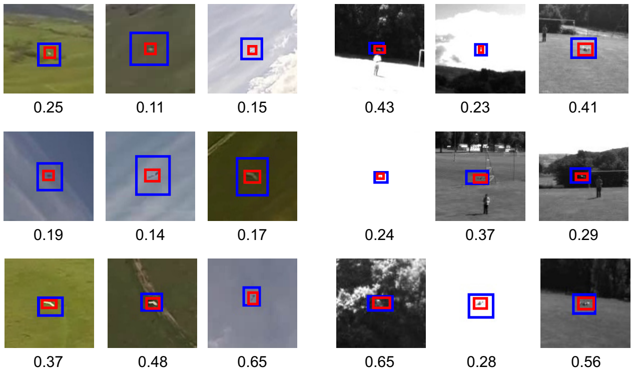

Figure 8. In the figure, we show the examples of our improved annotations. The blue boxes show the originally released annotations by the authors and the red boxes show our annotations done. The first three columns show the images from the NPS-drone dataset [24] and the last three columns are images from the FL-drone dataset [33]. Under each image, we show the IOU between new and old annotations.

Method	Precision	Recall	F1 score	AP
SCRDet-H [42]	0.54	0.62	0.58	0.52
SCRDet-R [42]	0.55	0.62	0.58	0.52
FCOS [36]	0.69	0.70	0.69	0.62
Mask-RCNN [13]	0.76	0.68	0.72	0.68
MEGA [7]	0.71	0.72	0.71	0.65
SLSA [39]	0.57	0.72	0.64	0.61
Proposed	**0.84**	**0.76**	**0.80**	**0.72**

Table 2. Quantitative comparison of the proposed approach with several state of the art approaches on the FL-Drones dataset [33].

.

4.3. Comparison with the State-of-the-art

We compare the proposed approach with recently proposed approaches such as fully convolutional one-stage object detector [36], cluttered and rotated small object detector [42], instance segmentation [13], and video object detectors [7, 39]. For comparison purposes, all the baseline methods are finetuned using the pre-trained weights available with public codes. Specifically, SLSA [39], FCOS [36], MEGA [7], and SCRDet [42] are finetuned using corresponding papers' pretrained weights and MASK-RCNN [13] is finetuned using Resnet50 weights trained on ImageNet. I3D has the weights obtained through training on Kinetics. Due to computational power limitations, all of the methods that we compared with are trained for iterations ranging from 80, 000 to 100, 000 using a single Nvidia 1080Ti GPU.

The quantitative comparison of proposed approach with the baselines over both drone detection datasets are shown in Table 1 and Table 2. The results demonstrate that our approach significantly outperforms the recent baselines on different evaluation metrics.

4.4. Ablation studies

Component analysis: We analyze different components of the proposed approach to verify their effectiveness. The experimental results in Table 3 enforce that each component of our approach is important and contributes toward final accuracy.

Method	Precision	Recall	F1 score
CE	0.87	0.82	0.85
$\mathcal{L}$	0.88	0.84	0.86
$\mathcal{L}+\mathcal{CA}$	0.88	0.86	0.87
$\mathcal{L}+\mathcal{PA}$	0.89	0.85	0.87
Stage-1	0.92	0.88	0.90
Stage-1+Stage-2	**0.92**	**0.91**	**0.92**

Table 3. Ablation study of different components of our method on NPS dataset. The first row shows the proposed approach employing only cross-entropy loss (CE). The second row represents results incorporating focal loss and Distance-IOU loss ($\mathcal{L}$). The third row demonstrates results employing channel-wise attention ($\mathcal{L}+\mathcal{CA}$). After that we show results using pixel-wise attention ($\mathcal{L}+\mathcal{PA}$). Finally, the last two rows show results after stage-1 and stage-2 respectively.

Two Stage Approach: As shown in Table 3 and Fig 9, we observe that two-stage approach produce much better detection results as compared to 1-stage approach by discovering and classifying difficult drone locations. We have also tried multiple frames in 1-stage similar to [21] but achieved very low detection accuracy (below F1-score=40). We believe that due to the large and complex target and source drone motion, the 1-stage multiple frames approach is unable to learn to detect drones.

Failure cases: Since the proposed approach tries to detect drones based on appearance and motion cues, the drones with slow motion and indistinguishable shape are hard to detect. Figure 10 show failure cases of our approach.

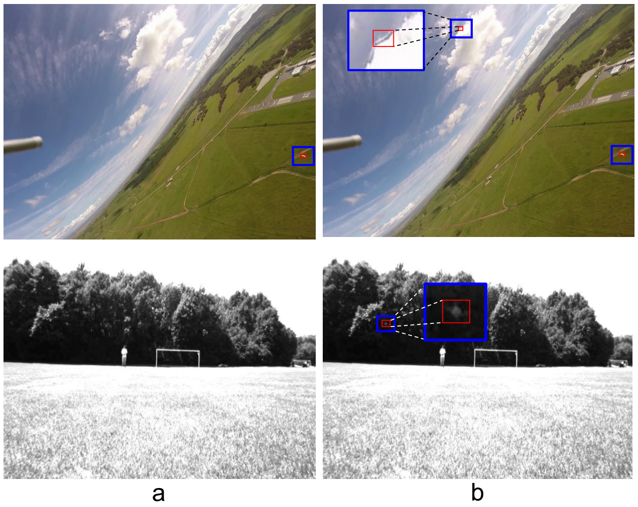

Figure 9. Qualitative comparison of one stage versus two stage detection results. (a) shows the detection results for stage-1 and (b) represents the detection results of two stage approach. Red boxes represent detection and blues boxes are just for better visualizations. The first stage misses one drone in each example.

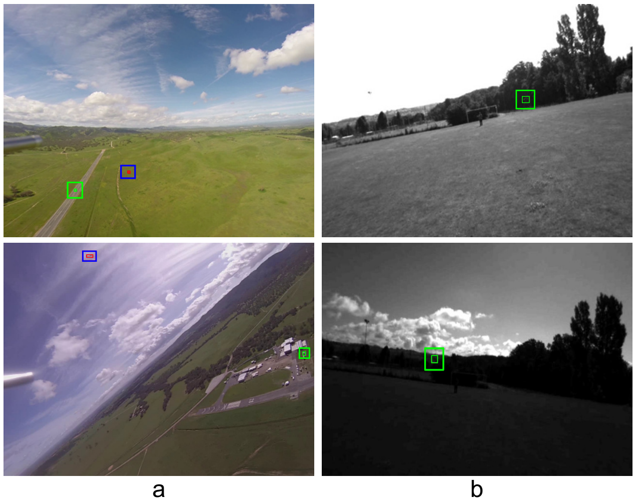

Figure 10. Failure cases of our method. Miss detections and correct detections are shown by green and red boxes respectively. Outer boxes are just for better visualizations. The drones with indistinguishable shape and motion are hard to detect. (a) and (b) represent samples from NPS-Drones [24] and FL-Drones [33] datasets respectively.

5. Conclusion

One of the most important challenges associated with drone use is collision avoidance or safe multi-drone flights. Therefore, it is crucial to develop robust computer vision methods that can detect and enable collision avoidance using inexpensive cameras. We have presented a two-stage approach for drone detection from other flying drones employing spatio-temporal cues. Instead of relying on region proposal-based methods, we have used a segmentation-based approach for accurate drone detection using pixel and channel-wise attention. In addition to using appearance information, we have also exploited motion information between frames to get a better recall. We observe that for drones to drones videos, the two-stage approach performs better than the one-stage approach. Thorough comparisons with the state of the arts and detailed ablation studies validate the framework and the ideas proposed in this work.

Acknowledgements: We would like to thank Mohsen Ali and Sarfaraz Hussein for their insightful discussions.

References

[1] D. H. Arjun Nagendran and M. Shah. Ucf-arg data set. 1

[2] M. Barekatain, M. Martí, H. Shih, S. Murray, K. Nakayama, Y. Matsuo, and H. Prendinger. Okutama-action: An aerial view video dataset for concurrent human action detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017. 1

[3] K. L. Best, J. Schmid, S. Tierney, J. Awan, N. M. Beyene, M. A. Holliday, R. Khan, and K. Lee. *How to Analyze the Cyber Threat from Drones: Background, Analysis Frameworks, and Analysis Tools*. RAND Corporation, Santa Monica, CA, 2020. 1

[4] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 6

[6] A. Carrio, S. Vemprala, A. Ripoll, S. Saripalli, and P. Campoy. Drone detection using depth maps. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1034–1037. IEEE, 2018. 4

[7] Y. Chen, Y. Cao, H. Hu, and L. Wang. Memory enhanced global-local aggregation for video object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10337–10346, 2020. 1, 2, 3, 7

[8] M. T. Chiu, X. Xu, Y. Wei, Z. Huang, A. G. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose, D. Wilson, A. Tudor, N. Hovakimyan, T. S. Huang, and H. Shi. Agriculture-vision: A large aerial image database for agricultural pattern analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[9] J. Choe and H. Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4, 5

[10] D. Du and P. Z. et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019. 1

[11] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman. What have we learned from deep representations for action recognition? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[12] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan. Augfpn: Improving multi-scale feature learning for object detection, 2019. 2

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 7

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 4

[15] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 4

[16] C. Huang, F. Gao, J. Pan, Z. Yang, W. Qiu, P. Chen, X. Yang, S. Shen, and K. Cheng. Act: An autonomous drone cinematography system for action scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7039–7046, 2018. 1

[17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] W. Kwon, J. H. Park, M. Lee, J. Her, S. H. Kim, and J. W. Seo. Robust autonomous navigation of unmanned aerial vehicles (uavs) for warehouses' inventory application. *IEEE Robotics and Automation Letters*, 2020. 1

[20] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, 2001. 5

[21] R. LaLonde, D. Zhang, and M. Shah. Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4003–4012, 2018. 8

[22] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 2

[23] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[24] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman. Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs). In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. 1, 2, 4, 6, 7, 8

[25] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang, J. Li, and J. Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 5

[27] M. Liu and M. Zhu. Mobile video object detection with temporally-aware feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5686–5695, 2018. 2, 3

[28] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In B. Leibe, J. Matas, N. Sebe,

and M. Welling, editors, *Computer Vision – ECCV 2016*. Springer International Publishing, 2016. 1

[29] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9725–9734, 2019. 2

[30] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[31] 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*, pages 91–99, 2015. 2, 5

[32] A. Rozantsev, V. Lepetit, and P. Fua. Flying objects detection from a single moving camera. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[33] A. Rozantsev, V. Lepetit, and P. Fua. Detecting flying objects using a single moving camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):879–892, 2017. 1, 2, 3, 4, 6, 7, 8

[34] D. Saikin, T. Báča, M. Gurtner, and M. Saska. Wildfire fighting by unmanned aerial system exploiting its time-varying mass. *IEEE Robotics and Automation Letters*, PP:1–1, 02 2020. 1

[35] W. Sultani and M. Shah. Human action recognition in drone videos using a few aerial training examples, 2020. 1

[36] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 1, 7

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2017. 4

[38] M. Vrba, D. Heřt, and M. Saska. Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system. *IEEE Robotics and Automation Letters*, 4(4):3402–3409, 2019. 4

[39] H. Wu, Y. Chen, N. Wang, and Z. Zhang. Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9217–9225, 2019. 1, 2, 3, 7

[40] Z. Wu, K. Suresh, P. Narayanan, H. Xu, H. Kwon, and Z. Wang. Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[41] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling. Clustered object detection in aerial images. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[42] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8232–8241, 2019. 1, 2, 3, 4, 5, 7

[43] D. H. Ye, J. Li, Q. Chen, J. Wachs, and C. Bouman. Deep learning for moving object detection and tracking from a single camera in unmanned aerial vehicles (uavs). *Electronic Imaging*, 2018:4661–4666, 01 2018. 4

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2

[45] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017. 4

[46] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-iou loss: Faster and better learning for bounding box regression. volume 34, pages 12993–13000, 02 2020. 5

[47] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. 2018. 1

[48] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 2, 3