

Flying Object Detection for Automatic UAV Recognition

Lars Sommer^{2,1} Arne Schumann¹ Thomas Müller¹ Tobias Schuchert¹ Jürgen Beyerer^{1,2}

¹Fraunhofer IOSB
Fraunhoferstrasse 1
76131 Karlsruhe, Germany

²Vision and Fusion Lab
Karlsruhe Institute of Technology KIT
Adenauerring 4, 76131 Karlsruhe, Germany

firstname.lastname@iosb.fraunhofer.de

Abstract

With the increasing use of unmanned aerial vehicles (UAVs) by consumers, automatic UAV detection systems have become increasingly important for security services. In such a system, video imagery is a core modality for the detection task, because it can cover large areas and is very cost-effective to acquire. Many detection systems consist of two parts: flying object detection and subsequent object classification. In this work, we investigate the suitability of a number of flying object detection approaches for the task of UAV detection based on video data from static and moving cameras. We compare approaches based on image differencing with object proposal detectors which are learned from data. Finally, we classify each detection by a convolutional neural network (CNN) into the classes UAV or clutter. Our approach is evaluated on six sequences of challenging real world data which contain multiple UAVs, birds, and background motion.

1. Introduction

In recent years unmanned aerial vehicles (UAVs) have conquered the consumer market. The increasing use of UAVs by hobbyists has led to new challenges in several areas where it may pose a threat to safety or privacy. This includes locations, such as airports, prisons, and borders or mass events, such as sports events or public demonstrations. In order for security services to be able to react to the presence of UAVs, an automated detection system is required. Such systems can rely on a number of signals in order to detect UAVs, including audio, UAV control signals, or video data. Video data in particular allows to detect and localize UAVs at large distances. However, detecting UAVs in video data poses several challenges, e.g., varying object dimensions from less than ten to hundreds of pixels, moving background, varying illumination conditions, and weak contrast



Figure 1. Detecting UAVs in image data poses several challenges such as very small object size, moving background, e.g., sea and clouds, varying illumination conditions, or weak contrast between the UAV and the background.

to the background, see Figure 1.

In this work, we focus on the detection of UAVs that is based solely on image data recorded by static or moving cameras. We separate the task into two steps: detection of flying objects and classification of such detections into whether they represent an UAV or not. A depiction of this popular pipeline is given in Figure 2. Our main focus in this work lies on a thorough evaluation of a number of suitable approaches for the first stage, flying object detection. We compare two groups of approaches, image difference based detectors and object proposal methods which are learned from data. We identify the advantages of different methods on our target datasets, investigate core parameters and formulate recommendations. Our aim for each detector is to have very few misses, because misses cannot be recovered in the classifier stage. Simultaneously we aim to keep the number of false positive detections as low as possible. We further evaluate the detectors' performances at different object resolutions. We are particularly interested in very small object sizes as these are required for a successful early



Figure 2. The established UAV detection pipeline. In the first detector stage flying objects are detected. A classifier in the second stage can then add semantics and reduce the number of false positive detections. The focus of this work is on the first stage. Note that missed detections from the first stage cannot readily be recovered in the second stage.

warning system. Finally, we train a convolutional neural network (CNN) classifier to help further reduce the number of false positive detections.

2. Related Work

In literature, there exists a large variety of techniques to detect UAVs [6]. In this paper, we limit our discussion to approaches based on visual-optical cameras and computer vision approaches. In general, these approaches consist of two steps as illustrated in Figure 2. First, moving objects are detected followed by a classification step to remove false alarms.

Ganti and Kim [1] propose frame differencing of two consecutive frames to detect moving objects. Then, SURF features are used to distinguish whether the object is a drone or not. In [2], the authors also propose two-frame differencing to detect moving objects. Then, a coherence score is computed for each blob to mitigate the number of false alarms. In [4], a background estimation and structural adaptive change detection process is initially applied to detect movements and other changes in the observed scene. Then, a background model is computed based on the local density of changes. The resulting detections are combined to trajectories, which are analyzed in order to filter out false alarms.

An alternative approach is proposed by Rozantsev et al. [6]. They use a regression-based approach for motion stabilization of image patches to allow an effective classification on spatio-temporal image cubes extracted by a multi-scale sliding window approach.

3. Flying Object Detection

As described in Section 2, many UAV detection approaches comprise a detection module to detect candidate regions that are likely to contain an UAV followed by a classification module to remove false positive detections. The performance of such approaches strongly depends on the detection module, because missed detections are irrecoverable and detections that only partially overlap with relevant objects might be misclassified. It is thus crucial for the detector stage to generate very few misses and for the resulting detections to have good alignment with the actual ob-

ject. In the following Section, we describe the methodology of three different detection techniques. The performance of these techniques and their strengths and weaknesses of each technique are then discussed in Section 5.

3.1. Image Differencing

Frame differencing and background subtraction are widely used to detect moving objects in video sequences from static cameras. In case of moving cameras, the camera motion is initially compensated by image alignment and only the overlapping image region is considered for moving object detection. The rationale of frame differencing and background subtraction based methods is to detect changes, e.g., moving objects, from the difference between the current frame and consecutive frames or a background model, respectively. Two- and three-frame differencing requires no computation of a background model. Background subtraction based methods allow instead the detection of slow moving or still objects whose positions partially overlap between consecutive frames.

The difference image D for direct two-frame differencing is calculated as

$$D(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \quad , \quad (1)$$

where $D(x, y)$ is the difference in intensity at pixel (x, y) and I_t and I_{t-1} denote the intensity values of frame t and frame $t - 1$. High difference values indicate areas or blobs where a change, e.g. motion, has occurred. However, this method produces two blobs for each moving object: one blob at its position in the current frame (t) and an additional blob at its position in the previous frame ($t - 1$).

Three-frame differencing can be used in order to avoid this effect. The difference image for three-frame differencing is given by the minimum of the difference between frame t and $t - 1$ and the difference image between frame t and $t + 1$:

$$D(x, y) = \min(|I_t(x, y) - I_{t-1}(x, y)|, |I_t(x, y) - I_{t+1}(x, y)|) \quad . \quad (2)$$

This formulation suppresses blobs which only occur in one of the two difference images.

In case of background subtraction, the difference image between an image I_t and the corresponding background model I_{BG} is calculated as

$$D(x, y) = |I_t(x, y) - I_{BG}(x, y)|. \quad (3)$$

Several approaches exist to compute a background model. We employ the straightforward approach of calculating the pixel-wise intensity median of consecutive frames or an entire image sequence to obtain our background image.

After computing the difference image, a fixed threshold value is used to distinguish the pixels into foreground (moving objects) and background pixels. Higher threshold values result in more missed detections whereas lower threshold values cause more false positive detections and less accurate localization. Morphological operations are applied on the thresholded image to remove single pixel detections and to fill in holes or gaps in object contours. We apply an opening operation with kernel size 3, followed by a closing with kernel size 10. Finally, connected component analysis is performed to cluster neighboring pixels. The bounding box around each cluster output is considered as a detection.

All three methods could be extended to work with moving cameras by stabilizing the camera image prior to the differencing. However, for UAV detection the major part of the image will often be sky. This leaves very few image clues for a successful stabilization or registration of images. Our aim is to develop UAV detection methods which are generally applicable and have as few restrictions as possible. We thus abstain from including stabilization methods in our pipeline.

3.2. Locally Adaptive Change Detection (LACD)

The detection of UAVs is based on the approach described in [4]. In this algorithm, a background estimation and structural adaptive change detection process detects movements and other changes in the recorded image sequence. In a learning phase (before any UAVs occur), the background model is initialized using a density computation of detections. In the successive operational phase the current situation is compared with the obtained background model in order to calculate the final detection result. In this procedure, present image noise is compensated and learned movements in the scene background are masked out. The found detections are then assembled to trajectories which are analyzed in order to filter out some false alarms stemming from, e.g., grass or leaves moving in the wind.

The learning phase needs sufficient image data without UAVs to build up an appropriate background model (representing the reference or past situation) with which the current situation can be compared. If an image sequence does not provide this directly (because UAVs are present throughout), we generate this needed data using a simple

temporal median method. First, all images are sorted temporally and pixel-wise, so that the first image of the sequence contains in its pixels all minimal gray values or color values of the sequence, the second image all minimal values of the rest of the sequence and so on. Finally, the last image represents all maximum values. Some of the first and last images of the generated sequence accumulate flying object depictions, whereas the images in the middle of the sequence are free from this, showing scene background.

3.3. Object Proposals

In contrast to the methods described in the previous sections, object proposal methods do not rely on motion to detect objects. Instead, these methods generate a set of candidate regions, which are most likely to contain an object. In the following, we apply the deep learning based Region Proposal Network (RPN) [5] which has been shown to clearly outperform object proposal methods based on hand-crafted features.

The RPN comprises a fully convolutional network (set of convolutional layers) and a small network which is shifted in a sliding window approach over the output of the last convolutional layer. The small network is composed of a 3×3 convolutional layer, whose output is fed into a classification layer and a bounding box regression layer. The classification layer provides a confidence value about the presence of an object, which is used to rank the proposals. The bounding box regression layer provides the corresponding coordinates for each proposal. Anchor boxes centered at the sliding window are used as regression reference.

For our experiments, we train a new RPN for each of five video sequences. For this, we use every fifth frame of four video sequences, which contains a drone, as training data and evaluate the RPN on all frames of the remaining fifth video sequence. All RPNs are trained end-to-end using the parameter settings proposed in [5], except for the anchor box sizes and the minimal proposal size to account for small drones. We reduce the minimal dimension of considered proposals to 8. The anchor base size is set to 2. In our experiments, we investigate two different architectures for the fully convolutional network of the RPN: VGG-16 [8] with 13 convolutional layers and CaffeNet [3] with 5 convolutional layers. Models pre-trained on the ImageNet dataset [7] are used to initialize the convolutional layers. We use the Caffe framework for all trainings.

In contrast to the methods described in Sections 3.1 and 3.2, the RPN is trained on a certain domain. Thus, re-training for new scenes that vary from the training domain might be necessary. However, the RPN is applicable in case of moving cameras as the RPN is applied to single images. Furthermore, the RPN can be integrated into a deep learning based detection framework to speed up the runtime [5].

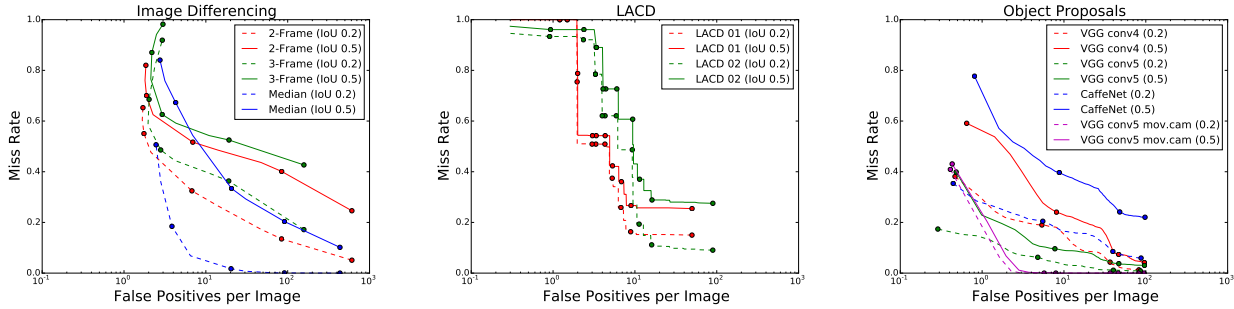


Figure 3. Performance of all three types of detectors. The image differencing methods were evaluated by varying their threshold (markers at 0.05, 0.1, 0.2, 0.3). For LACD we varied the minimum required track length from 0 to 1000 (markers each 100 steps). The proposal networks were evaluated by varying the maximum number of accepted proposals from 1 to 100 (markers at 1, 10, 30, 100).

4. Flying Object Classification

We investigate the potential of the classification stage by training a CNN classifier on the dataset. This is a challenging task, as only very little data is available. In order to maximize our training data, we train in a leave-one-out fashion where we reserve one sequence for testing and train on the remainder of the data. We iterate this process so that every sequence is evaluated once and average our results. We enhance the existing annotations by annotating all birds in the data and randomly sample background patches from the video sequences to serve as our negative class. We choose the CaffeNet model for this training task as well. Fine-tuning pre-existing weights is important for stable training with the small amount of data. We also apply data augmentation by randomly flipping and translating our training samples¹. The RPN described in Section 3.3 can be directly integrated with a CNN classifier and trained end-to-end. For a fair comparison with the other methods we keep the RPN trained separately and independent of the classifier.

5. Evaluation

We evaluate the flying object detectors on a dataset of five static camera videos and one moving camera video. Impressions of the data are given in Figure 5. We use the popular intersection-over-union (IoU) criterion to match detections to ground truth. Our main evaluation metrics are the number of false positive detections created by a detector and the number of missed detections. The number of false positives is of secondary importance, as these can be eliminated by the subsequent classification stage. Our main objective is to avoid misses.

5.1. Overall Performance

The overall performance of all three types of detectors is given in Figure 3. We compare the average number of false positives in each image to the number of misses generated across all static camera sequences. We evaluate once by using the established IoU value of 0.5. However, this threshold requires very good alignment between detections and ground truth. Weak alignment can be dealt with in the classification stage. We thus also evaluate with an IoU threshold of 0.2 which gives a more accurate idea how many UAVs have been entirely missed.

The detection approaches based on image differencing are evaluated by varying the threshold which is applied to the difference image. We choose this threshold t as a factor to the maximum possible distance value, i.e., $t \in [0, 1]$, $t_{abs} = 255 * t$. In our evaluation we vary t from 0.05 to 0.3. Among the three methods, the median image approach generates the least amount of misses. In particular for the IoU 0.2 setting a perfect result of 0 misses can be found for a value of $t = 0.06$ which comes at the cost of almost 100 false positive detections per image. The two- and three-frame difference methods perform notably weaker. Three-frame differencing performs particularly poorly when the UAV motion is very slow.

The LACD approach generates tracks as outputs. We evaluate two settings: one in which the learning phase of the detector is placed at the beginning of each video sequence (red) and one where the learning sequence is determined as described in Section 3.2 (green). For evaluation we varied the minimum accepted track length in order to filter very short tracks which usually represent clutter. The threshold for the minimum track length was varied from 0 to maximum (1022). The LACD results can initially be greatly improved without any new losses. However, the trade-off achieved with longer tracks is not as strong as the performance of the median image detector. Overall, the impact of a lower IoU value is reduced compared to the image differ-

¹Our trained models are available at s.fhg.de/uavdet-avss17

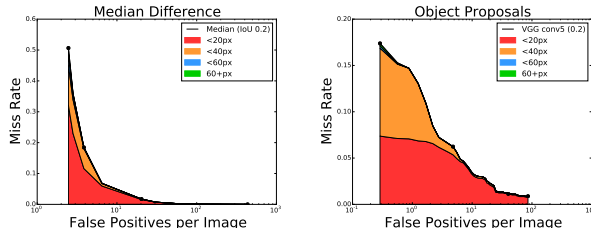


Figure 4. Both detectors can very reliably detect UAVs of more than 40 pixels squared. Far distant UAVs are the main source of failure. Note the different scales on the miss rate axes.

encing methods which indicates a better localization accuracy of this approach.

We evaluate three variations of object proposal networks: one based on CaffeNet and two based on different layers of the VGG net. Evaluation is performed by varying the number of proposals we accept in each image up to a maximum of 100 proposals. The trade-offs achieved by this approach are among the best of all detector types. In particular the proposals based on the *conv5* layer of the VGG network have good alignment with few misses and few false positives. However, the proposal networks are not able to achieve 0 misses due to the small objects in the static camera sequences. On the moving camera sequence which only contains larger objects a very good detection accuracy can be achieved, including 0 misses at less than 10 false positives per image.

Based on these observations we choose the median image based detector and the region proposal detector based on the VGG *conv5* layer as our principal flying object detector. The median detector is the faster approach, does not require any training, and is able to achieve 0 missed detections. The proposal detector gives a better trade-off for our metrics and is applicable to moving camera sequences as well. Using our unoptimized and single-threaded code the median image detector requires 0.4 seconds per image and the proposal detector 0.8 seconds.

Figure 5 shows qualitative detection results. Top ten ranked proposals generated by the RPN using VGG-16 are shown in the top row. The proposals highly overlap with small and large UAVs under varying illumination conditions. In case of multiple flying objects, proposals are not only generated for one object as illustrated in Figure 5 (top right). The top ten proposals cover both the UAV and the bird. In case of tiny objects, the RPN using CaffeNet fails as the top 100 proposals cover the more structured background (bottom left). The median detector is not confused by structured background (bottom middle) but generates more false positive in many images (bottom right).

5.2. Impact of Object Size

To get a better idea of the failure cases of the two chosen detectors, we investigate the number of missed detections in



Figure 5. Qualitative detection results.

relation to their image size. Results are depicted in Figure 4. We grouped the UAV annotations into groups of four different sizes, ranging from very small occurrences in the range of 10 to 20 pixels squared up to large detections with over 60 pixels squared. It can be seen that the majority of missed detections happens for very small sized (i.e. far distant) UAVs. Almost no UAVs of size 40 or higher are missed by any of the detectors. Particularly the proposal detector has problems finding the smallest UAV occurrences. However, in relation to false positives, the trade-off achieved by the proposal detector is again clearly better for very small objects. A requirement of less than 25 false positives per image, for example, would lead to a miss rate of almost 0.1 for small objects with the proposal detector while the median detectors miss rate is well above 0.5.

5.3. UAV Classification

We apply the CaffeNet UAV classifier in order to reduce false positives without creating many new misses. We thus choose a conservative threshold and only discard detections which are predicted to be no UAV with high confidence of more than 0.9. Based on the median image result at $t = 0.06$ and 0 misses we are able to reduce the number of false positives per image from 91 to 82 without an increase in misses. The proposal detectors false positives per image can be decreased from 25 to 23 at a fixed miss rate of 0.1. Thus, even with our limited training data the resulting classifier, when applied conservatively, has a positive impact on the detection accuracy.

6. Conclusion

We have evaluated multiple flying object detection approaches with the goal of UAV detection. We found that a simple median image based approach can perform very well for static cameras and generates the least amount of missed detections. In moving camera scenes, region proposal networks perform well and can even give a better trade-off, if a few missed detections can be accepted. Even with very little training data and a straightforward fine-tuning approach we were able to show the potential of a UAV classifier to further reduce the number of false positive detections for both detectors.

References

- [1] S. R. Ganti and Y. Kim. Implementation of detection and tracking mechanism for small UAS. In *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. IEEE, 2016. 2
- [2] S. Hu, G. H. Goldman, and C. C. Borel-Donohue. Detection of unmanned aerial vehicles using a visible camera system. *Applied Optics*, 56(3), 2017. 2
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 3
- [4] T. Müller. Robust drone detection for day/night counter-UAV with static VIS and SWIR cameras. In *Proceedings of the SPIE: Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XIV*. SPIE, 2017. 2, 3
- [5] 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. 3
- [6] A. Rozantsev, V. Lepetit, and P. Fua. Detecting flying objects using a single moving camera. Technical report, Institute of Electrical and Electronics Engineers, 2016. 2
- [7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014. 3
- [8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3