A Dataset of Infrared Images for Deep Learning based Drone Detection

Purbaditya Bhattacharya
Dept. of Signal Proc. and Comm.
Helmut Schmidt University
Hamburg, Germany
bhattacp@hsu-hh.de

Patrick Nowak
Dept. of Signal Proc. and Comm.
Helmut Schmidt University
Hamburg, Germany
patrick.nowak@hsu-hh.de

Daniel Ahlers
Dept. of Signal Proc. and Comm.
Helmut Schmidt University
Hamburg, Germany
ahlersd@hsu-hh.de

Udo Zölzer
Dept. of Signal Proc. and Comm.
Helmut Schmidt University
Hamburg, Germany
zoelzer@hsu-hh.de

***Abstract*—Recently, drone detection has become a topic of interest due to the widespread usage of drones in various applications, particularly for recreational purposes. Such detection tasks are usually performed by deep learning models which require different kinds of image datasets to be trained on. Hence, a dataset of infrared images for drone detection is introduced in this paper. In order to generate the dataset, videos of drones are captured initially with multiple cameras at two different locations. The video frames are then extracted and the drones are annotated with the help of an annotation tool and an automated script. A comprehensive analysis of the dataset is provided and multiple configurations of a selection of CNN models are trained on a fraction of the dataset. The trained models are employed on the test dataset and their performance is evaluated.**

***Index Terms*—Dataset, object detection, drone detection, deep learning, convolutional neural network, image processing**

I. Introduction

In recent years, the usage of drones has increased rapidly across several industries or fields of application. They were always used in military operations but are now widely used for industrial and commercial applications, too. Some of the use cases include security for law enforcement, monitoring for rescue operations, transportation, and aerial photography for various applications. Additionally, recreational drones are in abundance now and flying them have been a regular occurrence in cities. Given their gradual growth, public safety becomes an area of concern and must be addressed. Hence, reliable and robust methods to detect and monitor drones are developed. An automated drone detection system can generate an alert based on any anomalies and capture drones. Such systems can operate based on data processed by different sensors. Indeed, interceptor drones which are equipped with sensors and capturing devices are already developed which are able to detect, follow, and capture any unauthorized drones flying over secure areas like an airfield. However, a surveillance system including various sensors and cameras needs to be available in a secure zone. In this context, infrared cameras are used as one of the sensors and they might provide better visibility under certain conditions even if the background is noisy or cluttered and the environmental illumination is low. Hence, a dataset of infrared images containing drones is introduced in this work.

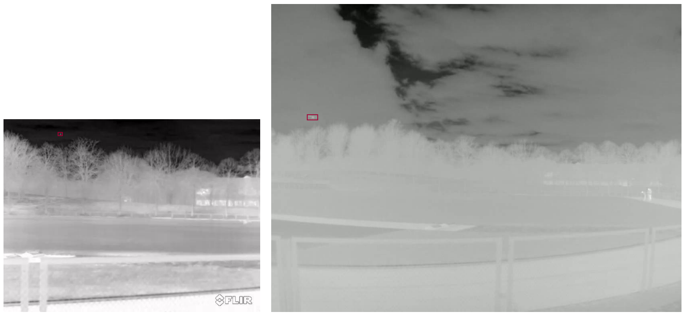
While many datasets of aerial images captured by drones are available across the internet, there is a dearth of open source databases containing images of drones, particularly in the infrared spectrum. Notable contributions in this area include the work in [1] where different scenarios consisting of drones are available as part of the Anti-UAV challenge, and in [2] which introduces a multi sensor dataset. The dataset introduced in this work is composed of annotated drone images captured by two different infrared cameras of different resolutions. Multiple drones are used during the recording and a fraction of the images contain multiple drones unlike some datasets. The drones are captured in front of multiple background objects and offer a varying difficulty in terms of visibility. The complete dataset will be publicly available in the near future.

The drone dataset is created with the objective of developing and training deep convolutional neural network (CNN) models in order to integrate such models in a drone detection system. Recently, deep learning based methods are widely used in multiple object detection tasks since they tend to produce excellent results [3]–[5]. Hence, the dataset is used to train a selection of deep learning models. The following sections provide the details of dataset generation and annotation along with a statistical analysis of the dataset. Finally, the performance of the models trained on the dataset is evaluated.

II. Dataset Generation

The dataset generation can be divided into two parts, namely video recording and annotation of drones in the video frames.

This work is part of the “Digital Sensor-2-Cloud Campus Platform” (DS2CCP) project, which is funded by the Federal Ministry of Defense under the dtec.bw program. Project website: https://dtecbw.de/home/forschung/hsu/projekt-ds2ccp



(a) FLIR

(b) InfraTec

Fig. 1: Comparison of the image resolution of both cameras: (a) FLIR Scion OTM366 (640×480 pixels) and (b) InfraTec VarioCAM HD Z (1024×768 pixels).

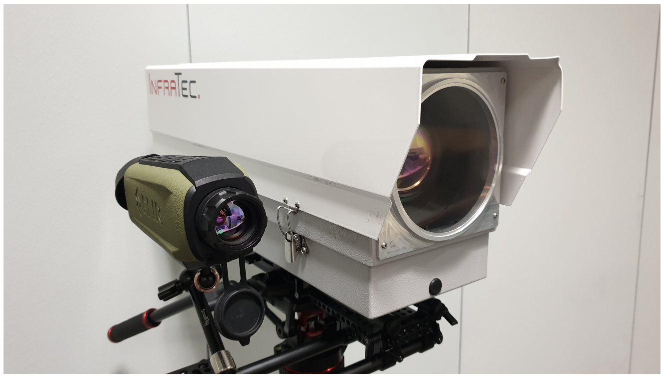


Fig. 2: Used infrared cameras: FLIR Scion OTM366 (left) and InfraTec VarioCAM HD Z (right).

A. *Video Recording*

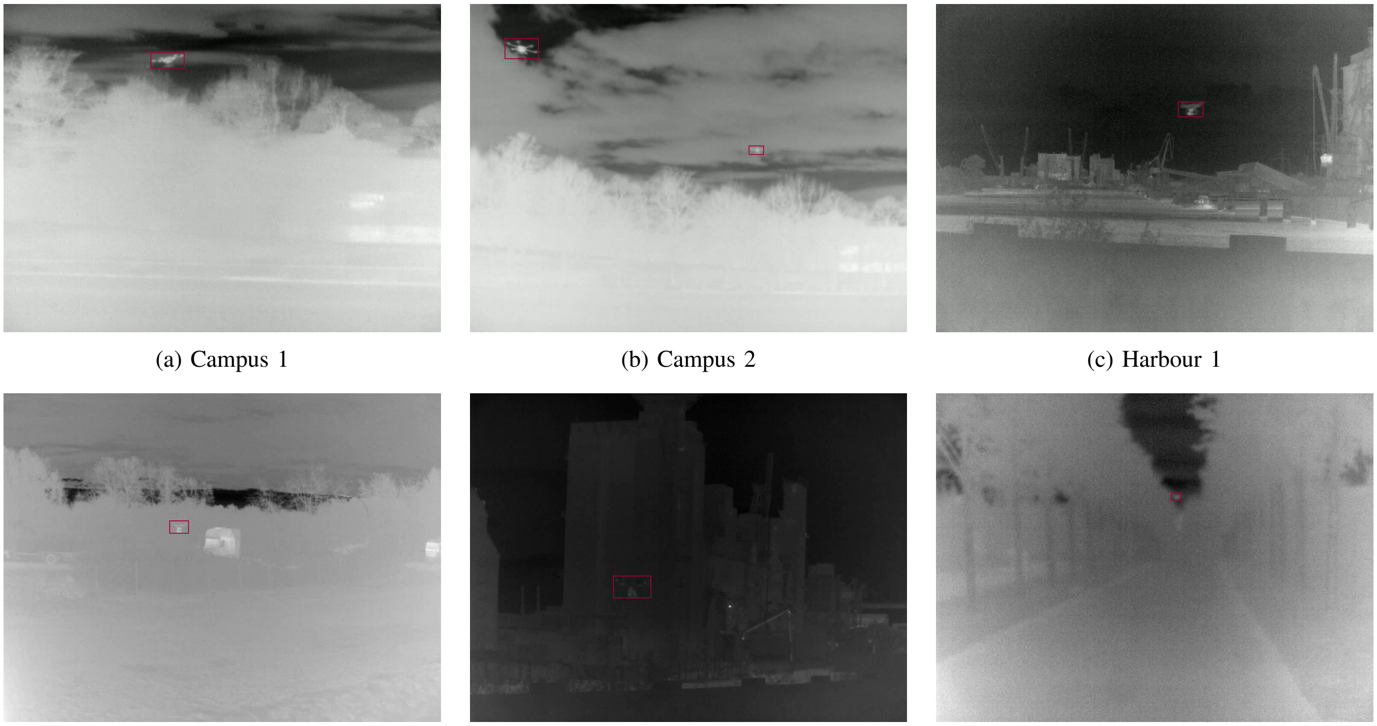
The dataset is constructed from drone videos captured using two infrared cameras with different image resolutions and an aspect ratio of 4:3 (see Fig. 1). The cameras can be seen in Fig. 2. Firstly, a FLIR Scion OTM366 is used to record drone videos in the infrared spectrum with a resolution of 640×480 pixels. Secondly, infrared videos with a higher resolution of 1024×768 pixels are recorded using an InfraTec VarioCAM HD Z equipped with a zoom lens (25-150 mm). The drone videos were recorded on different days at two different locations using both cameras simultaneously. At first, videos were recorded on the football field of the university campus from different perspectives and distances between 100 and 200 meters. In total, three different drones can be seen inside these videos. In most of the videos, an Artcopter Raptor drone is shown. The second drone in some videos is a Holybro X500. A third smaller drone is visible in some videos, which is a DJI Mavic Pro 2. Exemplary images recorded by the InfraTec VarioCAM HD Z with one or two drones can be seen in Figs. 3(a) and 3(b), respectively. As second recording location, a small harbour was chosen due to the variety of possible backgrounds like harbour, (industry) buildings, trees, or trucks (see Figs. 3(c)-(f)). Here, a DJI Phantom 2 drone is recorded. In post-processing, longer phases without a drone are removed and the associated videos divided into several parts. In total, seven videos are recorded by the FLIR Scion OTM366 and five by the InfraTec VarioCAM HD Z that are divided into nine and 14 partial videos, respectively. After the video recording, the individual frames are extracted and saved in JPG format.

B. *Annotation*

In order to use the recorded videos for training and evaluation purposes, drones inside the video frames have to be labelled. For this purpose, the LabelImg [6] graphical annotation tool is used. The annotations are done in the Pascal VOC [7] format. To speed up the labelling process, a MATLAB script is written that uses cross-correlation to determine the current position of the drone in the vicinity of the previous position. The resulting annotations are then manually checked and corrected if applicable. Additionally, annotations of drones that are barely distinguishable from the background in individual images or are mostly outside the image are marked as difficult. An exemplary annotation marked as difficult can be seen in Fig. 3(f). Here, the drone is barely distinguishable from the treetops.

III. Dataset Analysis

The images and annotations of the dataset are divided into 23 folders each, corresponding to the 23 partial videos described above. This means that the images can be used not only as single images for drone detection but also as continuous video frames to perform drone tracking. In total, the dataset consists of 66,438 images and 71,520 annotated drones. A more detailed composition of the dataset can be seen in Table I. The images taken with the FLIR Scion OTM366 represent about 61 % of the dataset (40,619 images), of which again about 65 % were taken at the harbour (26,368) and the remaining images on campus (14,251). The lower number of 25,819 images recorded by the InfraTec VarioCAM HD Z can be explained by the unbalanced number of images on campus (2,446) and at the harbour (23,373). This imbalance is mainly due to the fact that the images recorded by the InfraTec VarioCAM HD Z on the campus hardly differ due to the required power supply during the recording and the resulting static point of view with identical background, which is why it was decided to annotate only a part of the images and add them to the dataset. In addition to the number of images, Table I also lists the number of annotated drones for each scenario. The number of drones per image shows that, as described in Sec. II, there are sometimes two or even three drones in one image on the campus, so that a total of 47,193 drones are labeled on images recorded by FLIR Scion OTM366 and 24,327 drones on images recorded by InfraTec VarioCAM HD Z. The total number of 2,617 images without a drone can be explained by the fact that the videos should not be divided into parts that are too short, despite the drone briefly disappearing from the image or being unrecognizable. In 10,130 cases, the annotated drone is marked as difficult because it is barely distinguishable



(a) Campus 1

(b) Campus 2

(c) Harbour 1

(d) Harbour 2

(e) Harbour 3

(f) Difficult

Fig. 3: Exemplary images of the dataset recorded by the InfraTec VarioCAM HD Z on campus with (a) one or (b) two drones and at the harbour with different backgrounds like (c) harbour, (d) trucks, (e) buildings, and (f) trees. Additionally, (f) shows an annotated drone marked as difficult.

from the background or mostly outside the image. The dataset prepared with images from FLIR Scion OTM366 and InfraTec VarioCAM HD Z is initially divided into normal and difficult images depending on the visibility of drones due to clutter, occlusion, or heavy blurring. Difficult images would contain a drone which is indistinguishable from its background due to similar temperature or heavily blurred due to motion or during automatic camera calibration. In this work, the relatively easy images are selected where the drones are mostly visible by the naked eye. It is however noteworthy that the set of normal images do contain a substantial number of challenging images where the drones are partially occluded/ blurred or the contrast between the drones and background is not big.

In addition to the number of images and drones, also the position of the drones in the images is important for an object detector. Therefore, Figs. 4 and 5 show heatmaps of the frequency of occurrence of the position of the annotated drones on the individual pixels of the images separately for the two cameras. Black pixels in the heatmaps represent positions that are not reached by any drone. As can be seen in Fig. 4, the drone usually remains in the centre of the image, which can be explained by the fact that the FLIR Scion OTM366 is a handheld device with which the responsible person followed the drone. This means that the drone does not often appear at the edge of the image, but the background of the image is more variable.

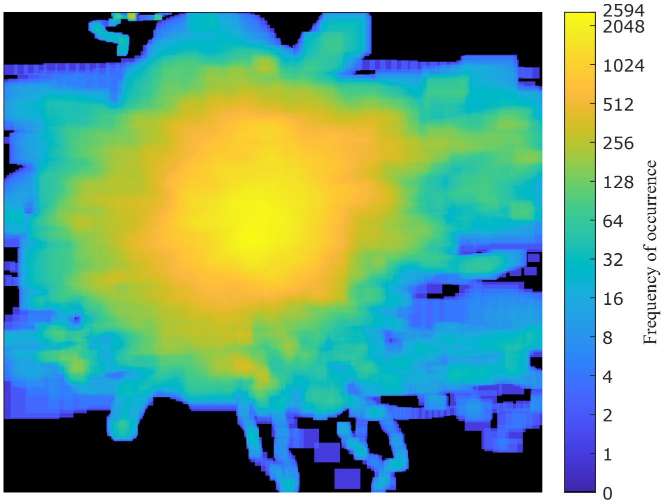


Fig. 4: Annotation heatmap of the 47,193 drones inside the videos recorded by the FLIR Scion OTM366.

In contrast to the FLIR Scion OTM366, the InfraTec VarioCAM HD Z is very heavy and requires a notebook to control it, which is why it is attached to a tripod that can be rotated but not panned. Thus, horizontal movements of the drone can be followed, but not the vertical movements, which means that the drone also appears at the top and bottom of the images

TABLE I: Total number of images and annotated drones contained in the dataset separated into different cameras and locations. Additionally, the number of drones marked as difficult is given.

Szenario	Number of images	Number of drones per image				Number of drones	Number of drones marked as difficult
		0	1	2	3		
FLIR (campus)	14,251	199	7,062	6,815	175	21,217	4,066
FLIR (harbour)	26,368	392	25,976	0	0	25,976	2,224
FLIR (total)	**40,619**	**591**	**33,038**	**6,815**	**175**	**47,193**	**6,290**
InfraTec (campus)	2,446	448	1,464	534	0	2,532	346
InfraTec (harbour)	23,373	1,578	21,795	0	0	21,795	3,494
InfraTec (total)	**25,819**	**2,026**	**23,259**	**534**	**0**	**24,327**	**3,840**

recorded by the InfraTec VarioCAM HD Z (see Fig. 5). Being a heavy camera, the horizontal movement can also be slow and very fast movement of the drone can sometimes be hard to follow. The most common positions can be found in the upper half of the picture.

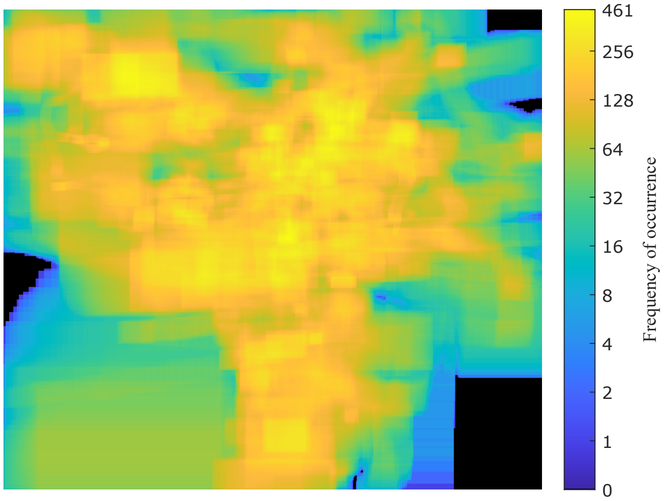


Fig. 5: Annotation heatmap of the 24,327 drones inside the videos recorded by the InfraTec VarioCAM HD Z.

Finally, the size of the annotated drones in the images is analysed, too. To do this, Table II lists four statistical metrics of drone size, namely minimum, mean, median, and maximum. The smallest drone sizes of down to 7 pixels are achieved by drones that are partially outside the image. For both cameras, the values of the drones on campus are notable lower than the corresponding values of the drones at the harbour. This is due to the considerably greater distance between the cameras and the drone during the recordings on campus. On average, the drones represent between 0.170 % and 0.605 % of the image. However, these average values are strongly influenced by the maximum drone size values, which range between 1.230 % and 14.092 %. Therefore, the median is also added to the table. This shows that, depending on the camera, half of the drones cover less than 0.151 % (FLIR) or 0.231 % (InfraTec) of the images. Overall, it can be said that the dataset reflects the reality of small drones at a further distance very well.

TABLE II: Size of the annotated drones contained in the dataset separated into different cameras and locations. The image size of the FLIR Scion OTM366 and InfraTec VarioCAM HD Z is 307,200 pixels and 786,432 pixels, respectively.

Szenario	Size of drones in pixels			
	Min	Median	Mean	Max
FLIR (campus)	7 (0.002 %)	325 (0.106 %)	590 (0.192 %)	3,780 (1.230 %)
FLIR (harbour)	140 (0.046 %)	532 (0.173 %)	1,858 (0.605 %)	33,460 (10.892 %)
FLIR (total)	**7** **(0.002 %)**	**464** **(0.151 %)**	**1,288** **(0.419 %)**	**33,460** **(10.892 %)**
InfraTec (campus)	88 (0.011 %)	578 (0.073 %)	1,337 (0.170 %)	10,731 (1.365 %)
InfraTec (harbour)	96 (0.012 %)	1,938 (0.246 %)	3,382 (0.430 %)	110,825 (14.092 %)
InfraTec (total)	**88** **(0.011 %)**	**1,820** **(0.231 %)**	**3,170** **(0.403 %)**	**110,825** **(14.092 %)**

IV. Deep Learning Based Drone Detection

In this section the FLIR and the InfraTec datasets are used to train and evaluate the performance of three deep learning models from the YOLO family - YOLOv5 [8], YOLOv6 [9], and YOLOv7 [10]. The models are pre-trained on the COCO dataset and the pre-trained weights are used to initialize the models during training. The primary commonality between these models is that their entire architectures can be divided into three primary structures, namely backbone, head, and neck.

TABLE III: Number of images split into training and test data for FLIR Scion OTM366 and InfraTec VarioCAM HD Z.

Number of images	FLIR	InfraTec
Training	23,816	15,213
Test	13,229	4,829

The backbone is a pyramidal structure primarily composed of convolution, batch normalization, and activation layers and it gradually reduces the spatial resolution of the feature maps while increasing the feature map depth. Hence, such a structure is used to generate multi-resolution feature maps to be processed by the next structure in the architecture. YOLOv5 [8] combines the Cross Stage Partial (CSP) Net [11] and the Darknet from earlier YOLO versions to get the CSPDarknet53 as its backbone. At the end of the YOLOv5 backbone there is an additional module called fast spatial pyramid pooling (SPPF) modified from spatial pyramid pooling (SPP) module introduced in [12]. This module aggregates feature maps processed by parallel branches of linear and non linear filters with activation functions. YOLOv6 [9] uses an efficient re-parameterizable backbone called EfficientRep with the help of module level re-parameterizable blocks (RepBlock) [13]. The behaviour of these blocks is different during training and inference, with the goal of computational load reduction during inference. Similar re-parameterizable modules are used in the backbone of YOLOv7 [10] as well, where they can be merged to more simpler modules during inference. Near the end of its backbone, YOLOv7 introduces a modified SPP block.

The neck of such architectures is responsible to collect and combine low and high level feature maps from selected layers in the backbone. The structure of the neck is usually a bottom up and a top down structure, where feature maps of same dimensions are either concatenated or summed. In order to increase the spatial dimensions bilinear interpolation is usually performed in the bottom up branch of the structure. Such a structure is called the Path Aggregation Network (PAN) in YOLO. YOLOv5 [8] uses the CSP-PAN structure while YOLOv6 [9] uses the Rep-PAN structure enhanced by the presence of RepBlocks or CSPStackRepBlocks in larger architectures. YOLOv7 [10] uses a PAN structure as well which includes the CSP-OSA modules and RepBlocks for efficient processing.

The head structure of such architectures contains the classification and the regression heads in order to predict the class scores and the relative coordinates of an object location. For classification, a form of focal loss [15] is used by the YOLO models. For box regression, these architectures define a set of initial anchor boxes and gradually learn the deviation required by the appropriate anchor boxes to locate an object. The loss function used by YOLO models is a variation of the IOU loss [16]–[18]. Additionally, YOLO models employ an objectness loss [8].

The models are trained separately with the FLIR and InfraTec datasets. Initially, the FLIR images and annotations are divided into training and test datasets. The images captured at the harbour are used for training while the images from the campus are used for testing. The InfraTec data is also divided into training and test datasets where most of the harbour images are used for training. Since the number of campus images is relatively low, a small section of the harbour images or images from a particular video are added to the campus images for testing. Table III shows the division of data into train and test set for each dataset. This division results in 64 % and 36 % of FLIR images for training and testing, respectively. In the case of InfraTec dataset, 67 % and 33 % of its data are used for training and testing, respectively. The network models are initialized with pretrained weights and the default hyperparameters are used, except for the learning rates and the weight decay values which are adjusted. To train with the YOLO models, the VOC annotations are converted to text annotations with the help of a conversion script. This script is however modified to produce correct transformations between coordinate systems defined in VOC and YOLO by avoid rounding and shift errors. Each model is trained for 80 epochs with a batch size of 8 on a Nvidia RTX8000 workstation GPU.

Table IV shows the performance of the models on the FLIR and InfraTec test datasets in terms of mean average precision ($\mathrm{MAP}_{\mathrm{IOU=0.5}}$, $\mathrm{MAP}_{\mathrm{IOU=0.5:0.95}}$) values at different intersection over union (IOU) thresholds and average processing time expressed in milliseconds. It can be seen in the table that all networks perform reasonably well on the test datasets. To train on the FLIR dataset, the variants YOLOv5m, YOLOv6m, and YOLOv7 are chosen where ’m’ denotes medium. Similarly, the medium variants of YOLOv5 and YOLOv6 and the YOLOv7W6 are selected for training with the InfraTec dataset. YOLOv5 has the least number of parameters and performs well on the FLIR test images. YOLOv7 performs relatively good as well and requires the least average processing time per image. With the InfraTec test dataset, YOLOv6 produces the best results in terms of MAP scores while YOLOv7 produces the fastest inference. The capacity of reparameterization in YOLOv6 and YOLOv7 makes the models usually faster at inference time in spite of having more parameters compared to YOLOv5. It can also be observed that the results obtained with the InfraTec dataset is better than the results with the FLIR dataset. This might be attributed to the fact that the InfraTec dataset has some images from the harbour in its training and test dataset, while the FLIR test set has no images from the harbour.

A selection of detection examples is shown in Fig. 6. The red boxes in the images denote ground truth labels and the green boxes along with the confidence scores denote detections. An example frame from the campus captured by the InfraTec VarioCAM HD Z is shown in the first row. As shown in Fig. 6(a), YOLOv5 is able to detect the bigger drone but it is unable to detect the smaller drone. It also makes a false detection with a relatively high confidence score. YOLOv7

TABLE IV: MAP results of different object detection models for FLIR and InfraTec dataset.

Dataset	Model	Number of parameters	$\text{MAP}_{0.5:0.95}$	$\text{MAP}_{0.5}$	Avg. Time / Image in ms
FLIR (640×640)	YOLO-v5m	20.85 M	**0.46**	0.78	19.40
	YOLO-v6m	34.80 M	0.42	0.64	20.43
	YOLO-v7	36.50 M	0.43	**0.82**	**18.70**
InfraTec (1024×1024)	YOLO-v5m6	35.25 M	0.51	0.81	46.00
	YOLO-v6m6	79.53 M	**0.60**	**0.89**	33.51
	YOLO-v7W6	80.90 M	0.55	0.85	**24.30**

is able to detect the bigger drone but it misses the smaller drone as well (see Fig. 6(c)). Only YOLOv6 is able to detect both drones in this example, as shown in Fig. 6(b). A similar behaviour can be noticed in the examples from the second row of Fig. 6 which is captured by the InfraTec VarioCAM HD Z at the harbour. As shown in Figs. 6(d) and 6(f) respectively, YOLOv5 and YOLOv7 are able to detect the actual drone while wrongly detecting another drone and producing a false positive each. YOLOv6 on the other hand performs better in this example (see Fig. 6(e)). Indeed, the performance of YOLOv6 is generally better compared to the other models and this behaviour is reflected in the results shown in Table IV.

The third and fourth rows of Fig. 6 show example images from the campus captured by the FLIR Scion OTM366. It can be seen in Figs. 6(g) and 6(i), that YOLOv5 and YOLOv7 are able to detect the drones in a relatively easy example. However, YOLOv6 is unable to detect the smaller drone, as shown in Fig. 6(h). In the last example, YOLOv5 and YOLOv6 are unable to detect the actual drone while making additional wrong detections (see Figs. 6(j) and 6(k)). YOLOv7 is able to detect the actual drone but also makes a false detection (see Fig. 6(l)). Notably, the drone in the middle which is detected by all the models is actually a drone, but is excluded from the datasets as part of the difficult examples. The general performance of the models as shown in Table IV is reflected in many such examples, with YOLOv6 being underwhelming. In general, the models behave well when the drone is well visible and the background is less noisy. However, there are occasional drops in detection due to sudden drop in confidence scores which leads to discontinuous tracking by all of the models. Such behaviour leads to a lack of robustness and will be addressed in the future.

V. Conclusion

In this work, a dataset of infrared drone images is introduced and described. The images are captured by two cameras of different resolutions at two different locations. The images are annotated in a hybrid approach and the ground truth labels are created in order to train deep learning models. Three such pre-trained models from the YOLO family are trained on the dataset and their performance is evaluated. In the future, the dataset will be expanded and more deep learning models should be trained. A dataset should also be created which contains both infrared and RGB images of the same scenery and the baseline models should be retrained on it to yield better results. Images of diverse nature and from different locations can also help extend the number of classes for object detection apart from drone. Additionally, the baseline models can be improved by introducing additional modules and data pre-processing methods. Model architectures can be changed so that they can process images of rectangular shapes instead of a square image, thus requiring no padding and possibly reducing the number of operations. Introduction of recursion, memory or transformers in such models can improve the robustness and address the problem of discontinuity in detection, making them suitable for tracking. Finally, an autonomous drone detection system will be built to detect and track drones.

References

[1] N.Jiang et al., "Anti-UAV: A Large-Scale Benchmark for Vision-Based UAV Tracking," in *IEEE Transactions on Multimedia*, vol. 25, pp. 486–500, 2023.

[2] F. Svanström, F. Alonso-Fernandez, and C. Englund, "A dataset for multi-sensor drone detection,", in *Advances in Neural Information Processing Systems (NIPS)*, vol. 28, 2015.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,", in *Data in Brief*, vol. 39, 107521, 2021.

[4] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.

[5] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, 2020.

[6] Tzutalin, "LabelImg," Free Software: MIT License, 2015, [Online], Available: https://github.com/HumanSignal/labelImg [Visited on 08/30/2023].

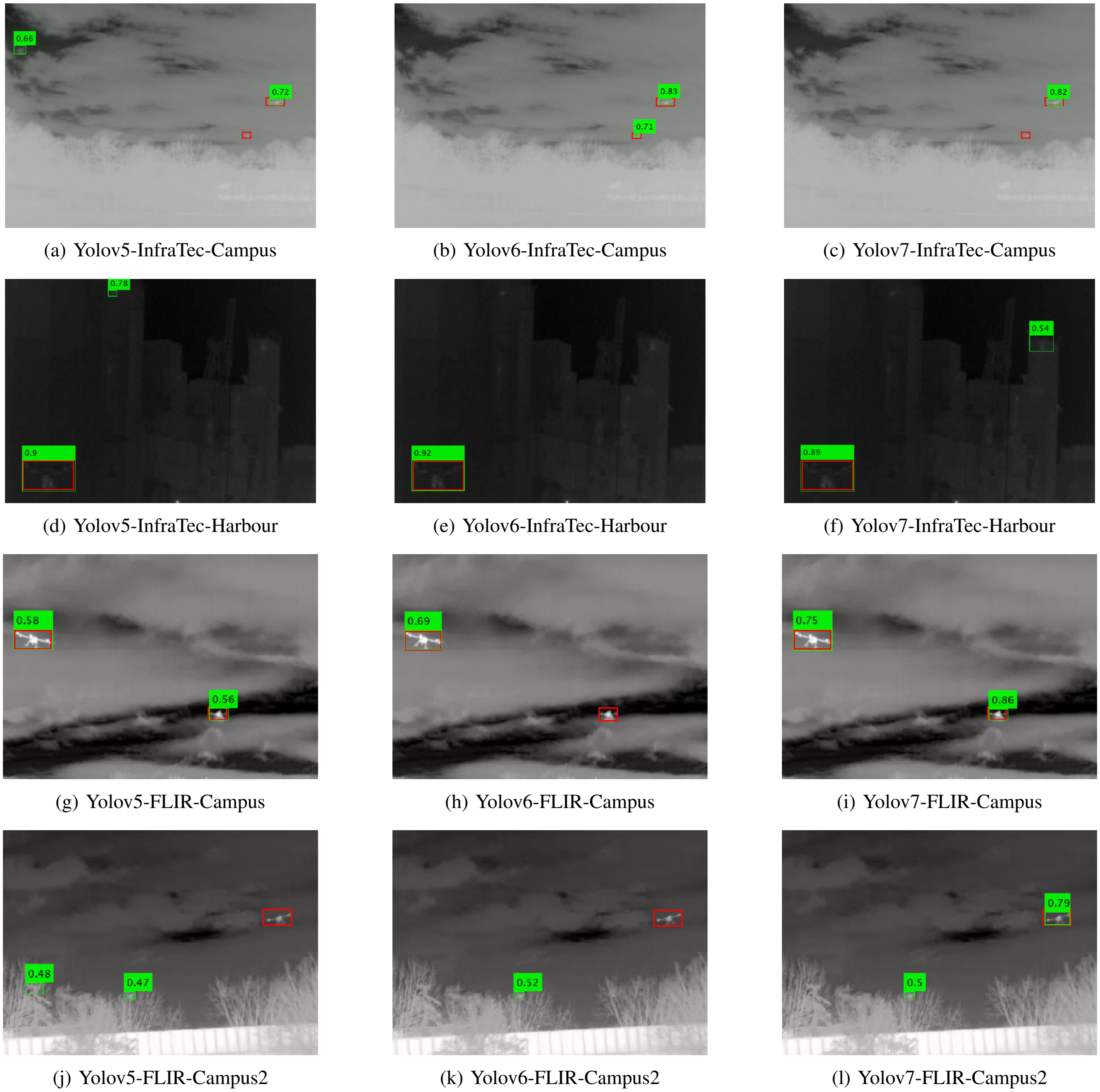
[7] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," in *International Journal of Computer Vision*, vol. 88, pp. 303—338, 2010.

[8] G. Jocker et al., "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Zenodo, October 2020, https://doi.org/10.5281/zenodo.4154370.

[9] C. Li et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," in *ArXiv*, 2209.02976, 2022.

[10] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *ArXiv*, 2207.02696, 2022.

[11] C.-Y. Wang et. al., "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, 2020.



(a) Yolov5-InfraTec-Campus

(b) Yolov6-InfraTec-Campus

(c) Yolov7-InfraTec-Campus

(d) Yolov5-InfraTec-Harbour

(e) Yolov6-InfraTec-Harbour

(f) Yolov7-InfraTec-Harbour

(g) Yolov5-FLIR-Campus

(h) Yolov6-FLIR-Campus

(i) Yolov7-FLIR-Campus

(j) Yolov5-FLIR-Campus2

(k) Yolov6-FLIR-Campus2

(l) Yolov7-FLIR-Campus2

Fig. 6: Example of drone detection on selected images from the test dataset by the YOLO models. Cropped sections of the entire image are shown for better visibility.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 9, pp. 1904–1916, 2015.

[13] K. Weng, X. Chu, X. Xu, J. Huang, and X. Wei, “EfficientRep:An Efficient Repvgg-style ConvNets with Hardware-aware Neural Network Design,” in *ArXiv*, 2302.00386, 2023.

[14] T. Kim, “Yolo-to-COCO format converter,” [Online], Available: https://github.com/Taeyoung96/Yolo-to-COCO-format-converter [08/30/2023].

[15] 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 (ICCV)*, pp. 2980–2988, 2017.

[16] H. Rezatofighi et al., “Generalized intersection over union: A metric and a loss for bounding box regression,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 658–666, 2019.

[17] Z. Zheng et al., “Distance-iou loss: Faster and better learning for bounding box regression,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 12993–13000, 2020.

[18] Z. Gevorgyan, “Siou loss: More powerful learning for bounding box regression,” in *ArXiv*, 2205.12740, 2022.