ISSN: 1983 7402 ITA, 29 a 30 SET 2020

Convolutional Neural Networks for target detection in thermal images

Henrique Eduardo de Macedo, José Maria Parente de Oliveira e Marcos Ricardo Omena de Albuquerque Máximo
Technological Institute of Aeronautics, ITA, São José dos Campos – SP, Brazil

***Abstract*— The use of thermal sensors embedded in remotely piloted aircraft constituted a remarkable advance in the operational capacity of military forces. However, the amount of information available alongside the regular workload overwhelms sensor operators. This study analyzes the performance of the YOLOv3 algorithm regarding target detection in a dataset of thermal images generated using the DcOMPASS sensor. The training method sought the configuration with the best performance conducting a hyperparameters search. Initially, training was carried out with the entire dataset, and then, separately with data only in *blackhot* or only *whitehot*. A software approach for the generation of images with inverted grayscale palette was also an option. The achieved results revealed training with images of opposite polarity (*whitehot and blackhot*) affects the final result negatively. The evaluation metrics were frame rate per second (FPS) and Mean Average Precision (mAP), and the finals scores demonstrate that YoloV3 can be successfully applied in the detection of targets in infrared images.**

***Keywords*— Target detection, Convolutional Neural Network, thermal images.**

I. Introduction

The use of Remote Piloted Aircraft Systems (RPAS) has grown exponentially in the world in recent years. Their versatility and potential for employment radically changed the most diverse sectors.

At the end of 2018, the United States of America had 277,000 drones, and the conservative projection for the year 2022 is a fleet of 400,000 [1]. Specifically in Brazil, considering only civilian drones that are required by law to be registered with National Civil Aviation Agency of Brazil (ANAC), their number jumped from 13,256 in 2017 to 76,865 in 2020, an increase of almost six times in three years [2].

In the military field, a RPAS main purpose is to serve as a platform for weapons and sensors on missions where employing a crew would be unnecessary or risky. Due to their versatility, thermal sensors are instruments equipped as a payload in vast majority of these aerial systems.

One of its functions is to help deal with object detection. Object detection in Unmanned Aerial Vehicle, as a kind of burgeoning technique, has numerous applications, such as aerial image analysis, intelligent surveillance, and routing inspection [3].

However, even today, the infrared image interpretation to discover targets always depends on human interpretation, which is inefficient and requires a lot of manpower [4]. RPAS offer digital images but theirs interpretation is manual.

MACEDO, H. E., eduardohem@fab.mil.br; PARENTE DE OLIVEIRA, J. M.,parente@ita.br; MAXIMO, M. R. O. A.mmmaximo@ita.br
Thanks to the National Council for Scientific and Technological Development - CNPq, for supporting the Process: 313282/2019-6.

Sensor operators on intelligence missions are overwhelmed with a vast amount of images and data made available by the sensor. In addition, due to the large flight autonomy of these aircraft, tasks such as target detection become more complex over time with increased fatigue.

In this context, automated tools for targets detection over specific classes are essential to achieve the expected result. In the field of artificial intelligence, convolutional neural networks (CNN) are state-of-art in detecting objects in images.

They are able to automatically and adaptively learn spatial hierarchies of features through backpropagation and are based on multiple building blocks (convolution layers, pooling layers, fully connected layers, etc [5].

Frequently new CNN-based algorithms are surpassing previous state-of-art results in competitions where they aim for the best performance in specific datasets such as COCO (Common Objects in Context) with more than 80 classes and 330,000 images [6].

The contribution of this paper rests on the analysis of the performance of the YOLOv3 [7] object detection algorithm over a single class of objects of interest (land vehicles). Infrared images from the Digital Compact Multi-Purpose Advanced Stabilized System-UAV (DCoMPASS) composes the dataset, and to the best of our knowledge, this is the first work to analyze images from that sensor.

Section II briefly describes related works. The was replaced by remainder of this paper is organized as follows, Section III gives theoretical background about CNNs and the structure of the YOLOv3 algorithm. After, Section IV highlights the attributes of the sensor and the dataset. Section V discusses the developed experiments and the obtained results. Lastly, Section VI concludes and shares our ideas for future work.

II. Related works

Other works sought to research about the detection of objects in infrared images. With a focus on detection in the marine environment, Scholler et al. [8] compared the performance of three convolutional neural networks using a large dataset composed of images of vessels and buoys in Long Wave Infrared (LWIR).

Some others as Akula, and Sardana [9], just tried to explore the extraction of attributes. Having as main objectives to evaluate the performance of a CNN as a feature extractor for target recognition in infrared imagery and additionally conduct experiments to optimize the classification performance using features extracted from different layers.

In the defense area, d'Acremont et al. [5] propose synthetic infrared images to produce large groundtruthed datasets for training, obtaining good results even with no data augmentation and fine-tuning steps.

Target recognition in thermal infrared images is challenging due to the high variability of target IR signatures and competing background IR signature due to a number of environmental and target parameters [9]. Factors such as internal heat source, exposure to sunlight, and time of day cawas replaced by n change its visualization in the image as its temperature increase or decrease.

The absence of publicly available data on the performance of convolutional neural networks in aerial infrared images, specifically to images obtained through the DCoMPASS sensor, motivated this research.

III. Convolutional Neural Networks

A. *Functioning of convolutional neural networks*

In the 1958, Frank Rosenblatt developed the Perceptron Model, inspired by previous works by Warren McCulloch and Walter Pitts.

The perceptron is the simplest form of a neural network used for the classification of patterns said to be linearly separable [10]. Each entry (x) is multiplied by its respective weight (w), then, added to a $bias$ and finally applied to an activation function lambda (λ) as shown in Figure 1.

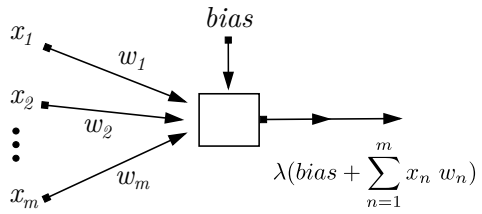


Fig. 1. Rosenblat's perceptron

Several neurons together form what is called artificial neural network (Figure 2). The first layer (input) sends the result of the activation function to the units of the intermediate (hidden) layers, in the same way, twas replaced by his layer sends the response to the final layer (output).

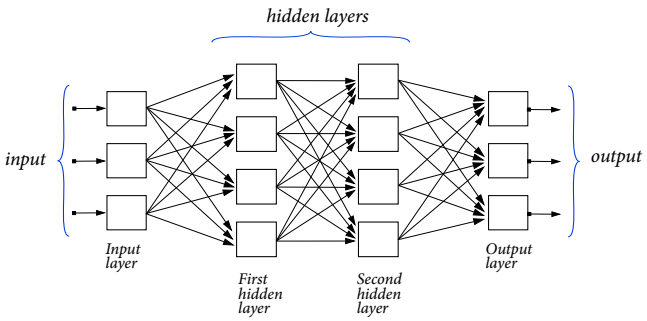


Fig. 2. Artificial neural network

Neural networks arranged in several layers allow the learning of increasingly abstract elements, a process known as deep learning. Within deep learning, convolutional neural networks are a specialized kind of neural network for processing data that has a known grid-like topology like images [11].

The main component of the convolutional neural network is the convolution layer. The main difference between a regular and a convolution layer is that a regular layers learn global patterns in their input feature space (for example, patterns involving all pixels), whereas convolution layers learn local patterns[12].

Roughly, the convolution is mathematical operation that, from two functions, generates a third one expressing how the shape of one is modified by the other. In the context of images, we can understand this process as a kernel that transforms an input image.

In convolutional network terminology, the first argument to the convolution is often referred to as the input, and the second argument is the kernel. The output is sometimes referred to as the feature map [11].was replaced by

This operation is applied several times in different regions of the image. At each application, the region is changed by a parameter called stride, usually equal to 1. Figure 3 describes the process.

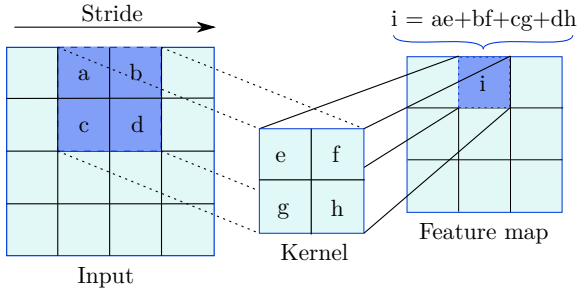


Fig. 3. Convolutional operation.

Another typical layer of a convolutional neural network is the pooling layer. A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the max pooling operation (Figure 4) reports the maximum output within a rectangular neighborhood [11].

Pooling helps to make the representation approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change [11].

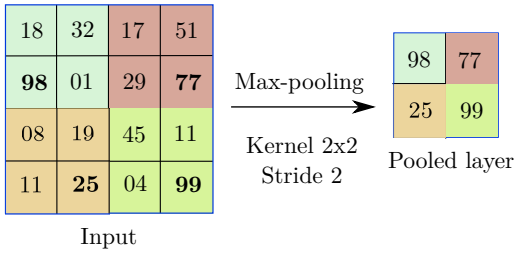


Fig. 4. Max pooling operation.

B. *YOLOv3 network*

YOLOv3 is an algorithm that employs a CNN for real time detection task. For conciseness, we provide a brief overview of the YOLO technique. For a more comprehensive explanation, we refer the interested reader to the original YOLO papers [13][14][7].

Darknet-53 feature extractor is the backbone of YOLOv3. This backbone provide 53 convolutional layers, each followed by a batch normalization layer and Leaky ReLU activation

function. During training the network use sum of squared error and binary cross-entropy as functions losses for bounding box and class predictions, respectively [7].

Besides detection, this kind of algorithm also predict class labels, just applying a single neural network to the whole image. YOLOv3 divides the image into $S \times S$ grid cells (for example 13×13). For each of these cells the algorithm predicts several anchor boxes.

Anchor boxes are boxes with predefined aspect ratios. These aspect ratios are determined before training by running K-means on the whole dataset.

The K-means algorithm aims to separate all samples to k clusters, which are usually chosen to be as far as possible from each other spatially, in Euclidean distance, to produce effective data mining results [3].

Once defined these anchors, the algorithm chooses the boxes that best overlapped the groundtruth using the Intersection over Union (IoU) concept.

IoU is given by the ratio of the area of intersection and area of union of the predicted bounding box and the ground truth bounding box (Figure 5).

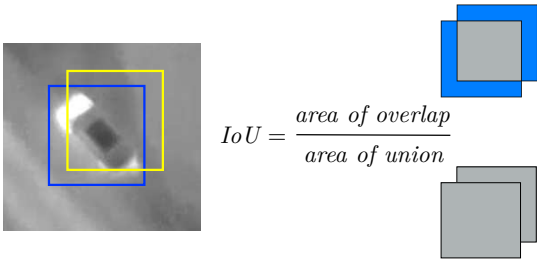


Fig. 5. Intersection over Union

For each grid cell, YOLOv3 provides one prediction for each anchor box in a vector format as depicted in (1):

$$y = \begin{bmatrix} t_x \\ t_y \\ t_h \\ t_w \\ t_o \\ c_1 \\ \vdots \\ c_n \end{bmatrix}. \quad (1)$$

t_o is an objectness score to indicate if this box contains an object and t_x, t_y, t_h, and t_w are offsets relative to a particular anchor box. The values $c_1, \ldots, c_n$ are class probabilities that permit us to associate a class with a box.

Using a sigmoid function (σ), is possible to obtain the bounding box center coordinates as depicted in (2) and (3). For bounding box width and height we need to apply the transformations (4) and (5) using width (p_w) and height (p_h) of the respective anchor box. All these transformations considers the grid cell has an offset from the top left corner of the image given by (c_x, c_y):

$$b_x = \sigma(t_x) + c_x, \quad (2)$$
$$b_y = \sigma(t_y) + c_y, \quad (3)$$
$$b_w = p_w e^{t_w}, \quad (4)$$
$$b_h = p_h e^{t_h}, \quad (5)$$
$$\text{confidence} = \sigma(t_o). \quad (6)$$

Where:

- b_x, is the x coordinate of the center of the bounding box.
- b_y, is the y coordinate of the center of the bounding box.
- b_h, is the bounding box height.
- b_w, is the bounding box width.
- confidence, is the probability an object is contained inside the grid cell $\times$ the IoU, $Pr(obj) \times IoU(b, obj)$.
- $c_1 \ldots c_n$, presence or not of each class using one-hot encoding.

YOLOv3 provides predictions at 3 different scales. At each scale, the prediction is a 3D tensor encoding bounding box (1).

IV. Sensor and Dataset

A. DcOMPASS

The DCoMPASS sensor, shown in Figure 6, is a surveillance payload for Intelligence, Surveillance, Target Acquisition, and Reconnaissance (ISTAR) with the following features: a high definition color TV camera, a large format thermal imager, a laser target illuminator, an eyesafe laser rangefinder, and a laser target designator [15]. The focus of this paper, the thermal imager, uses an indium antimonide detector that covers the medium-wavelength from 3.4 to 4.8 μm with 512×640 pixels definition and 25 frames per second.

Fig. 6. DcOMPASS payload for Unmanned Aerial Vehicles.

Despite all images being in grayscale, the sensor also offers the operator the possibility to switch between two palette options for thermal images. In the first, called *blackhot*, the warmest regions in the sensor's field of view are the darkest. In the second option, *whitehot*, the light regions are the warmest, as shown in Figure 7.

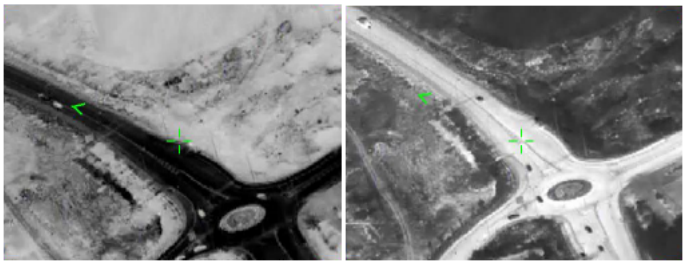


Fig. 7. Images in *blackhot* (left) and *whitehot*.

TABLE I
DATASET SUMMARY

Attribute	Value
whitehot images	2440
blackhot images	975
total samples	3415
width x height	576 x 720
image size	≈ 30 kb, 96 dpi
average targets/sample	13,48

B. Dataset

In this research, we collected the image data to compose the dataset from videos of the DcOMPASS sensor trying to make it more diverse as possible. The selection of videos sought to include samples of all hours of the day, at a flight altitude ranging from 6000 to 10000 ft, with tilt angles between 30 and 80 degrees, while searching for varied target states for example vehicles with engine off, and engine on, several temperatures, and meteorological conditions. This diversity of conditions added to the characteristics inherent to thermal images that may vary over the course of the day are in themselves an augmentation procedure.

After this first round of selection, we extracted the frames at an interval of 1 frame/second from scenes related to the class of interest. To perform the extraction, the opensource software FFmpeg was used. The final data are 576×720 pixels images, around 30 kB each in RGB, although, all of them are in gray tones.

At the last step, the resulting dataset was annotated with bounding boxes indicating the groundtruth using a custom software which exported the annotations to the format required by the YOLO framework. This paper focuses on a single class (*vehicle*) which includes only cars, trucks and buses (Figure 8), although the network only treats it as a single class. The images collected were the most diverse possible, full and partial images were used, with isolated and grouped elements. Table I shows a brief summary of the dataset.

Fig. 8. Example of scene with several vehicles.

V. EXPERIMENTS AND RESULTS

Initially, 10% of the samples were randomly separated from the dataset for testing purposes. The evaluation used as metric the mean average precision (mAP) computed from the average precision (AP) of each class, as defined in (10) and (9), respectively. The mAP is the standard measure used in computer vision to evaluate the performance of algorithms in the detection task. In this research, as there is only one class the two terms are equivalent. The calculation of AP is done by integrating the area under the precision (7) and recall (8) function curve, Figure 9.

The metrics Precision (P) and Recall (R) are defined as:

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

Where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively.

We determine whether a bounding box (BB) is a true or a false positive if the IoU value is beyond a certain threshold, as follows:

- True Positive: $IoU > 0.5$.
- False Positive: $IoU < 0.5$ or a duplicated bounding box.
- False Negative: no detection at all or has an $IoU > 0.5$ but has a wrong classification. In our case, as we have only one class that is not an issue.

Having computed all values, we draw a plot of $precision \times recall$ curve Figure 9. So, given a class $c \in \{c_1, c_2 .., c_m\}$, the average precision for this class (AP_c) is the integral over the function $precision = p(recall)$, in other words, the graphics area, according to (9).

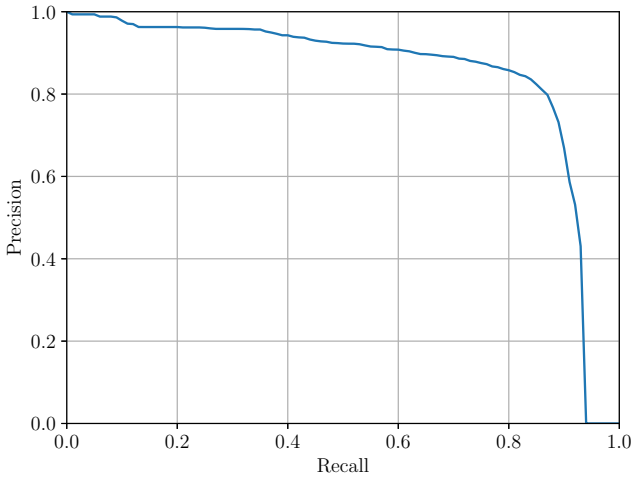


Fig. 9. Precision-Recall curve.

Then, to obtain the mAP, we simply calculate the average of the AP of all classes, as in (10).

$$AP_c = \int_0^1 p(r)dr \tag{9}$$

$$mAP = \frac{1}{m}\sum_{n=1}^{m} AP_c \tag{10}$$

In a defense context, no large-scale real datasets can generally be collected [5]. Due to the small size of ours, we tried to circumvent this issue with a software solution. So, for each image in *whitehot* a script produced its corresponding *blackhot* image or vice versa as a photographic negative, thus, doubling our dataset. However, the network performance slightly decreased, a 0.21% mAP drop, and a homogeneous set in this aspect showed itself more promising.

To confirm this fact, the network trained with the whole dataset, containing the originals *whitehot* and *blackhot* images. After, we carry out training with only one category at a time. Despite having more data available, the first network achieved a lower result than the networks trained with uniform data. Hence, all subsequent training sessions used a dataset consisting only of *whitehot* images with 2440 samples.

Besides, most of the images have flight data annotations in green overlapping the images. The sensor system does that automatically. At first, we chose not to provide labels for targets covered by these notes. However, the network training performed better including these targets.

In many application scenarios, the dimensions of boundary boxes exhibit strong patterns. Detection algorithms take advantage of this fact to improve their performance and use preselected bounding boxes customized to the dataset. YOLOv3 uses 9 preselected bounding box priors (anchors) [16]. For performance improvement, we calculated custom anchors boxes using K-means algorithm for our dataset.

After processing, our custom anchors have width and height as follows: (22×19), (19×33), (32×27), (26×43), (44×37), (34×61), (62×51), (43×93), (84×100).

For the definition of the best hyperparameters set, this paper employs a random search. This process is more efficient for hyper-parameter optimization than trials on a grid. When granting random search the same computational budget, it finds better models by effectively searching a larger, less promising configuration space [17].

A python code carried out a total of 60 training sessions randomly changing the hyperparameters before each round through trial and error. The modifications included the following hyperparameters: batch size, learning rate, momentum, and burn-in. Additionally, the script also acted on the saturation, angle, and exposure used for data augmentation. The range and best set of hyperparameters are described in Table II.

Our setup uses Nesterov Accelerated Gradient as the optimizer. We also use a fixed decay value of 0.0005 to penalize the weights.

All the training took place in an Ubuntu 18.04 environment with a Nvidia GeForce GTX 1080 Ti GPU using the Darknet framework by Alexey Bochkovskiy [18].

TABLE II
HYPERPARAMETERS RANGE

Attribute	From	To	Best
Batch Size	16	112	48
Learning rate	0.001	0.0001	0.0004
Momentum	0.65	0.99	0.93
Burn-in time	200	600	400
Scale	0.005	0.15	0.12
Saturation	0.5	5	2.5
Exposure in time	0.5	5	0.5
Angle rotation	0	180	80

The search obtained the $mAP@0.5$ results shown in Figure 10. As expected, there was a lot of variation in the outcomes. Rounds numbers 11, 23, and 48 did not even converge. Training round number 50 was the best set. Table III shows complete mAP results through all IoU.

We can observe YOLOv3 has difficulties to indicate precisely object localization (the mAP is only 9.8% at $IoU = 0.75$). That is a known characteristic of this algorithm.

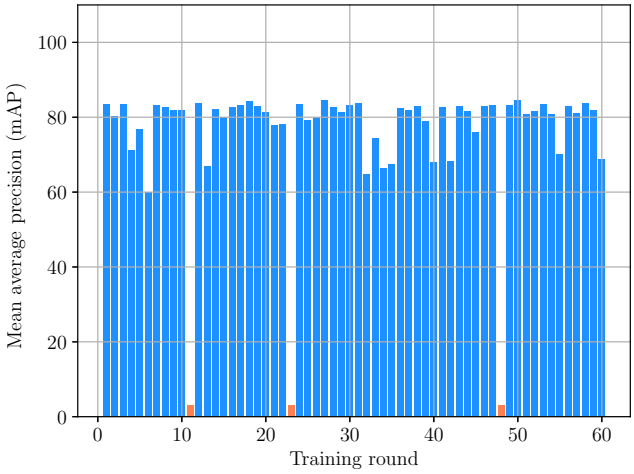


Fig. 10. Random search results at $IoU = 0.5$.

TABLE III
mAP RESULTS

mAP@IoU=0.50	mAP@IoU=0.75	mAP@IoU=0.50:0.95
0.8463	0.098	0.292

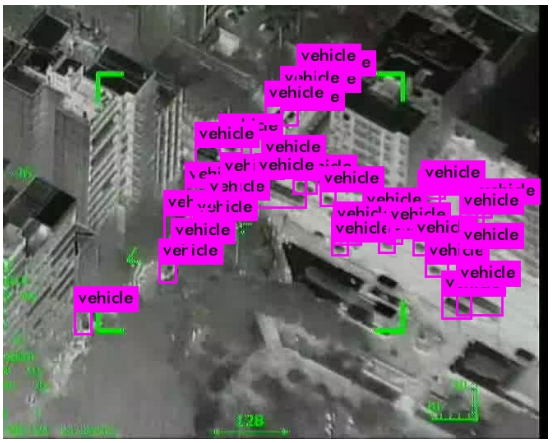
The computational performance test was performed on a Tesla T4 16G from Google Colab and obtained the average value of 29.8 frames per second (fps) considering an image resolution of 608×608 pixels. This value allows us to use it in real-time detection.

Figures 11a and 11b show an example with the same image before and after detection performed by the algorithm.

VI. CONCLUSIONS

This article assessed the YOLOv3 algorithm for object detection of targets in middle wave infrared images from the

(a)



(b)

Fig. 11. Detection examples.

DcOMPASS sensor.

The numbers reached by the algorithm in terms of precision and computational time demonstrates its potential to be applied in real operational activities. The use of this type of technique can collaborate to improve the performance of military missions with unmanned aerial vehicles.

It could be used in the processing of videos after the flight to mark scenes, or, due to its speed, it can also be successfully used to detect targets in real time. That is a great support to reduce the workload of the sensor operator.

A disadvantage was its inability to generalize together the two categories of images, *whitehot* and *blackhot*, with the same performance. The use of two networks each one trained in a type of image could address the problem.

The specifics of military sensors and the difficulty in obtaining large sets of data due to issues such as the number of sensors in operation and the secrecy of missions imposes the need to be well aware of the performance of these algorithms, specifically when applied to image types other than conventional RGB.

As a suggestion for future works, the comparison with other algorithms such as YoloV4 or EfficientDet over the same dataset would allow the evaluation of the best network for operational use.

REFERENCES

[1] "Unmanned aircraft systems report," Federal Aviation Administration, 2020. [Online]. Available: https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/Unmanned_Aircraft_Systems.pdf

[2] "Drones - quantidade de cadastros," Agência Nacional de Aviação Civil, 2020. [Online]. Available: https://www.anac.gov.br/assuntos/paginas-tematicas/drones

[3] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "Uav-yolo: Small object detection on unmanned aerial vehicle perspective," *Sensors (Switzerland)*, vol. 20, no. 8, pp. 1–12, 2020.

[4] G. Zheng, X. Wu, Y. Hu, and X. Liu, "Object detection for low-resolution infrared image in land battlefield based on deep learning," in *Chinese Control Conference, CCC*, vol. 2019-July. IEEE Computer Society, jul 2019, pp. 8649–8652.

[5] A. D'Acremont, R. Fablet, A. Baussard, and G. Quin, "CNN-based target recognition and identification for infrared imaging in defense systems," *Sensors (Switzerland)*, vol. 19, no. 9, may 2019.

[6] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dolí, "Microsoft COCO: Common Objects in Context," 2015.

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[8] F. E. Schöller, M. K. Plenge-Feidenhans'L, J. D. Stets, and M. Blanke, "Assessing Deep-learning Methods for Object Detection at Sea from LWIR Images," in *IFAC-PapersOnLine*, vol. 52, no. 21. Elsevier B.V., jan 2019, pp. 64–71. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S240589631932169X

[9] A. Akula and H. K. Sardana, "Deep CNN-based Feature Extractor for Target Recognition in Thermal Images," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, vol. 2019-Octob. Institute of Electrical and Electronics Engineers Inc., oct 2019, pp. 2370–2375.

[10] S. S. Haykin *et al.*, *Neural Networks and Learning Machines Third Edition*. New York: Pearson Prentice Hall, 2009.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[12] F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 779–788. [Online]. Available: http://pjreddie.com/yolo/

[14] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 6517–6525. [Online]. Available: http://pjreddie.com/yolo9000/

[15] ELBIT, "ELOP DCoMPASS Digital Compact Multi-Purpose Advanced Stabilized System-UAV," Elbit Systems, Tech. Rep., 2016. [Online]. Available: www.elbitsystems.com

[16] S. Chanda, P. K. Prasad, A. Hast, A. Brun, L. Mårtensson, and U. Pal, "Finding logo and seal in historical document images - an object detection based approach," in *Pattern Recognition - 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26-29, 2019, Revised Selected Papers, Part I*, ser. Lecture Notes in Computer Science, vol. 12046. Springer, 2019, pp. 821–834. [Online]. Available: https://doi.org/10.1007/978-3-030-41404-7_58

[17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.

[18] A. Bochkovskiy, "Darknet," https://github.com/AlexeyAB/darknet, 2020.