INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 4, PP. 825–831
Manuscript received July 16, 2023; revised October, 2023. DOI: 10.24425/ijet.2023.147707

Optimization of Animal Detection in Thermal Images Using YOLO Architecture

Łukasz Popek, Rafał Perz, Grzegorz Galiński, and Artur Abratański

Abstract—**The article presents research on animal detection in thermal images using the YOLOv5 architecture. The goal of the study was to obtain a model with high performance in detecting animals in this type of images, and to see how changes in hyperparameters affect learning curves and final results. This manifested itself in testing different values of learning rate, momentum and optimizer types in relation to the model's learning performance. Two methods of tuning hyperparameters were used in the study: grid search and evolutionary algorithms. The model was trained and tested on an in-house dataset containing images with deer and wild boars. After the experiments, the trained architecture achieved the highest score for Mean Average Precision (mAP) of 83%. These results are promising and indicate that the YOLO model can be used for automatic animal detection in various applications, such as wildlife monitoring, environmental protection or security systems.**

Keywords—**artificial neural networks; YOLOv5; transfer learning; genetic algorithms; thermal imaging**

I. Introduction

ANIMAL detection in thermal images is one of the significant challenges in the field of image analysis and object recognition. Thermal images, which record thermal radiation emitted by bodies, can provide valuable information on the presence and location of animals in various scenarios, such as wildlife monitoring, environmental protection and security systems in sparse visibility. However, due to their specific nature, animal detection in thermal images is more difficult than in traditional video images. The purpose of this paper is to apply the YOLO architecture to animal detection in thermal images. Through the use of machine learning, deep neural networks and object detection techniques, an attempt is made to create an efficient and accurate system that can automatically identify animals.

In recent years, many advanced object detection techniques have been developed, that use machine learning and deep neural networks. Moreover, the development of of thermal imaging techniques has increased the availability of such devices in the civilian market at affordable prices, making it possible to conduct research and create datasets. Currently, the vast majority of work on the acquisition of images containing animals is centered around the issue of object detection on materials obtained during unmanned aircraft vehicle (UAV) raids. The main goal of this approach is the macroscopic determination of the population size of a specific species. Good illustration of an article about testing the usefulness of animal detection for forested areas is [1]. It should be mentioned that this approach to the problem is different from the assumed research. The distance between the lens and the animal must be large enough to not frighten the object of study. This makes it impossible to take accurate shots. The result is change the problem from detecting specific animal species to distinguishing, whether a brighter spot is an artifact or a living object.

In this context, the closest to the stated problem are articles on animal observation using photo traps. This allows a non-invasive method of observing fauna, from a distance that makes it convenient to take pictures. The paper [2] summarizes the issue well from the technical side and the process aspect, and also reduces the problem into two classes devision - cervids and porcupines. However, the authors used only RGB scale photos to teach the solution which means a completely different morphology of the materials from the one adopted in this paper. According to [3] ,it was proved, that despite large number of wild animal species on images, still there is possibility to achieve good numeric results. With dataset containing 11 classes the mAP achieved for detection task scenario was on level of 87%.

Quite similar problem was considered by the authors of papers [4] and [5] on the prevention of terrorism and illegal immigration using autonomous detection of objects in harsh weather conditions with a special focus on people. The dataset contained more than 20,000 thermal images extracted from the videos, although presented in RGB scale. While the article [4] was more exploratory for the problem studied, in [5] authors tried to made a comparison of the performance, state-of-the-art object detectors, such as Faster R-CNN, SSD, Cascade R-CNN. The results obtained by the authors, confirmed the effectiveness of the methods chosen.

On the other hand, meeting the needs of the autonomous vehicle industry, the authors of the paper [6] tried to test the SSDMobileNet architecture model for object detection among the following classes: car, bicycle, human. The publication is very valuable because of its tests on a publicly available

Łukasz Popek is with Warsaw University of Technology, Faculty of Electronics and Information Technology, and Sieć Badawcza Rafał Perz, Poland (e-mail: popek.luka@gmail.com).

Rafał Perz and Artur Abratański are with Warsaw University of Technology, Faculty of Power and Aeronautical Engineering, and Sieć Badawcza Rafał Perz, Poland (e-mail: rafal.perz@pw.edu.pl, artur.abratanski.dokt@pw.edu.pl).

Grzegorz Galiński is with Warsaw University of Technology, Faculty of Electronics and Information Technology, Poland (e-mail: grzegorz.galinski@pw.edu.pl).

dataset. Unfortunately, the sheer quantitative results obtained by the authors - an mAP of 35% means that the model was not well enough matched to the problem. In the article [7], instead of taking pictures from the ground level, they were taken using UAV-s. However, the authors make an effective attempt to distinguish between species of Australian fauna - not only by getting as close to the object as possible, but also by modifying the YOLO architecture itself. As a result, the results obtained are satisfactory although still the issue itself is not identical. This research is a natural continuation of the work [8] containing a preliminary analysis and selection of the optimal method for further development. Using a dataset consisting of deer and hog classes, traditional image segmentation methods were compared along with the implementation of popular transfer learning-trained neural networks. The quantitative results for the most thematically related scientific articles are collected and summarized in Table I.

TABLE I
RESULTS FROM SIMILAR STUDIES

Reference	Model	Classes	mAP	Recall
[4]	YOLOv2	human	97%	75%
[5]	YOLOv3/FasterRCNN 4	human, dog, other	98%	79%
[6]	SSDMobileNet V1/V2	human, car, bike	35%	24%
[7]	YOLOv3/D-YOLO	wild boar, rabbit, kangaroo	97%	96%
[8]	YOLOv3/FasterRCNN	wild boar, cervidae	96%	70%

II. Theory

A. Thermal imaging

Thermal imaging technology is based on the use of special sensors called thermal imaging cameras, which are capable of recording infrared radiation. This radiation is invisible to the human eye, but can be detected by thermal imaging sensors. Thermal imaging cameras consist of a matrix of pixels that measure the intensity of sensed radiation emitted by objects. Based on this data, thermal cameras generate a thermal image in which different colors or shades represent different temperatures. The principle of thermal imaging technology is based on the detection of temperature differences between objects and the environment. In the context of wildlife monitoring, two main bands are crucial: mid wavelength infrared (MWIR) and long wavelength infrared (LWIR). A long wave acquisition camera was used during the study. They do not require an additional source of light or heat, as thermal radiation sensors in these ranges capture the thermal energy of the objects being observed. The camera range of acquisition depends on the quality of device. In this study images were obtained from Pulsar Helion XP50, which enable observation with high resolution up to 100m.

Thermal imaging technology, which performs very well in conditions of limited visibility, completely loses this advantage during the day. Considering work of [9], it can be mentioned several limitations of thermal imaging technology. Sensors provide much less detail than visible light cameras, because instead of the information provided by color in the visible spectrum, they only provide detected temperature ranges in thermograms, usually with much lower resolution. The outcomes of measurements can be notably influenced by various weather conditions, including solar radiation, precipitation, wind, and air humidity. Moreover the temperature on the outer surface of the body is significantly impacted by the physical characteristics of the animal's coat, such as its thickness and quality, it can cause sometimes major problems with recognising right species of animal on the image.

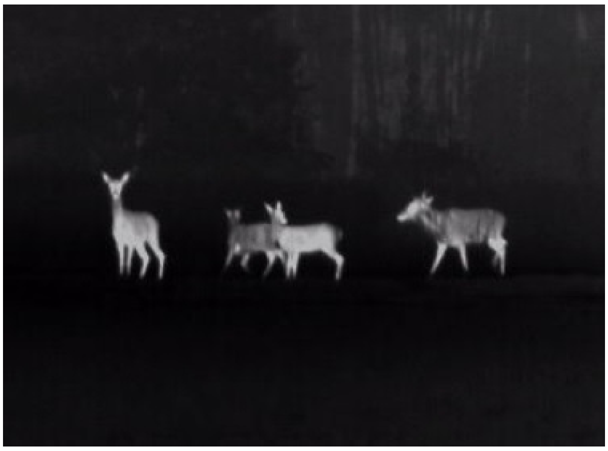

Fig. 1. Example image from thermal camera

B. Autonomous detection using YOLOv5

In recent years, object detection systems have become an indispensable tool in image analysis and visual processing. However, most existing object detection solutions rely on performing classification in multiple areas of the image, leading to slower speeds. To solve this problem, the YOLO architecture was developed, which is an acronym for ”You Only Look Once.”

The YOLO architecture is one of the most popular and efficient methods of real-time object detection. The basic idea of this architecture is to simultaneously detect and classify objects in a single image processing operation. Unlike traditional approaches, YOLO treats object detection as a regression problem rather than a classification problem. It divides an image into a grid of cells, and then for each cell it predicts the object's membership in different classes and calculates regression of the object's position and size. This method is based on deep neural networks, which consist of several convolutional layers and linking layers that extract object features from the image. The basic element of the YOLO architecture is the final layer, which generates predictions. Within this layer, it uses a convolution technique of feature maps to predict bounding boxes containing objects and the probability of belonging to different classes for each envelope. Confidence thresholds are then applied to filter out unreliable predictions, leaving only the most certain objects. What sets the YOLO architecture apart is its speed [10]. Because it performs object detection in a single processing operation, it achieves much higher speed than traditional approaches. Moreover, by calculating regression the position and size of objects, YOLO is able to accurately determine the location of objects in an image.

In subsequent years, improved versions of YOLO have been developed, improving the speed of operation and detection efficiency. This paper considers version 5 made available by [11]. Unlike traditional YOLO, version 5 uses a more complex architecture called EfficientDet, based on the EfficientNet architecture. Using the more complex architecture in YOLOv5 increases accuracy and better generalization to a wider range of object categories. Another difference between YOLO and YOLOv5 is the training data used to train the object detection model. YOLO was trained on the PASCAL VOC dataset, which consists of 20 object categories. Moreover, YOLOv5 was trained on a larger and more diverse dataset called D5, which contains a total of 600 object categories.

YOLOv5 uses a new method for generating Anchor Boxes, called "dynamic anchor boxes." It involves using a clustering algorithm to group real bounding envelopes into clusters, and then using the centroids of the clusters as anchor boxes. This allows the anchor boxes to match the size and shape of the detected objects more precisely. YOLOv5 also introduces the concept of Spatial Pyramid Pooling, (SPP), a type of pooling layer used to reduce the spatial resolution of feature maps. SPP is used to improve the performance of small object detection because it allows the model to see objects at multiple scales.

C. Hyperparameters of neural networks

Hyperparameters in neural networks are important for their optimal performance. During transfer learning, they allow for obtaining better results and faster convergence. Following the work of [12], the following were selected for optimization.

1) Learning rate: The learning rate is called often the most important hyperparameter [13]. It is a scalar value typically set between 0 and 1. It determines the proportion by which the model's parameters are adjusted in response to the gradient of the loss function with respect to those parameters. A higher learning rate allows for larger updates, potentially leading to faster convergence, but it also increases the risk of overshooting the optimal solution or oscillating around it. Moreover the optimization algorithm may fail to converge or exhibit unstable behavior, resulting in poor generalization performance. Conversely, a lower learning rate ensures more cautious updates, which can help to achieve convergence, but at the cost of slower training progress. This requires a significantly higher number of iterations to reach the desired performance. The choice of an appropriate learning rate is crucial for a successful training of machine learning models. Proper tuning of the learning rate is often considered a critical component of model optimization, as it can significantly impact the overall performance and effectiveness of the trained models.

2) Momentum: Momentum, as a key factor, introduces a dynamic component to the optimization process, influencing the behavior of parameter updates [14]. It is a hyperparameter that affects the process of updating the model weights. It allows gathering momentum during learning and helps speed up convergence. It determines the influence of previous updates on the current update step, allowing the optimization process to traverse challenging optimization landscapes more efficiently. By adding momentum to the update equation, the algorithm gains inertia, enabling it to overcome small-scale fluctuations and accelerate convergence. The proper selection and adjustment of the momentum parameter is critical for achieving optimal convergence and avoiding convergence stagnation or overshooting. Momentum values, that are too high can cause oscillations, and those that are too low can slow down the learning process.

3) Optimizer: Is an algorithm used to update weights during the learning process. Two optimazers were consider in this work: Stochastic Gradient Descent (SGD) and ADAM. SGD is a classic optimization algorithm that updates the parameters using the gradient of the loss function computed on a mini-batch of training samples [15]. It follows a fixed learning rate throughout the training process and does not include any adaptive learning rate mechanism. On the other hand ADAM is an adaptive learning rate optimization algorithm that combines the benefits of both AdaGrad and RMSProp [16]. It uses adaptive learning rates for each parameter by estimating first and second moments of the gradients. It incorporates momentum to improve convergence speed and can handle sparse gradients effectively.

In terms of learning rate, SGD uses a fixed value that remains constant throughout the training process. This can make it more challenging to find an optimal value for the given problem, as manually tuning the learning rate may be required. ADAM automatically adapts the learning rate for each parameter based on the estimated first and second moments of the gradients. It scales the learning rate based on the magnitudes of the gradients, effectively reducing the learning rate for parameters with large gradients and increasing it for parameters with small gradients.

For determining Momentum value, ADAM incorporates it by keeping track of the exponentially decaying average of past gradients. This helps accelerate convergence by adding a persistent direction to the parameter updates. However, the momentum term can also introduce bias towards previous updates, potentially impacting convergence in certain cases. SGD can also include momentum by adding a fraction of the previous gradient to the current update step. However, it requires manual tuning of the momentum parameter, and without careful adjustment, it may hinder convergence or cause oscillations.

The exact choice between values and methods of updating hyperparameters depends on the specific characteristics of the problem at hand, the available computational resources.

D. Hyperparameter tuning and metrics used

Hyperparameter tuning is a key step in neural network optimization, which aims to find a set of values that will achieve the best model performance. During the work, two strategies were used to select them: the grid method (grid search) and genetic algorithms.

The grid method involves testing different combinations of hyperparameters in a defined parameter space. First, a space of hyperparameters is defined with a selected range of values to be tested. Then a grid of hyperparameter combinations is created, where each combination represents one set. For each set, a model is trained on the training data using an appropriate optimizer, such as SGD or ADAM. The end result is the monitoring of performance measures, such as mean Avarage Precision or loss function, performed on the validation set. Table II collects the defined parameter values for which the experiments were performed.

TABLE II
CHECKED HYPERPARAMETERS VALUES

SGD	0.01	0.937
SGD	0.001	0.937
SGD	0.001	0.9
SGD	0.001	0.99
ADAM	0.001	0.999
ADAM	0.0001	0.9
ADAM	0.0001	0.99
ADAM	0.0001	0.999

Genetic algorithms are an effective method for tuning hyperparameters in neural network optimization. Through a combination of selection, crossover and mutation, we can search the space of hyperparameters to find optimal combinations and prevent the algorithm from stopping at a local extreme. In the present work, the entire process was performed using predefined functions for the model provided by [11]. The first step was to initialize the initial population and pretrain it for 10 epochs. Next, the quality of the model was evaluated based on the predefined adjacency function. All hyperparameters of each generation were loaded, along with its fitness. The top 5 generations evaluated on the basis of the fitness function are listed. Its value was calculated based on a weighted average of the metrics obtained. Mean Average Precision (mAP) is a widely adopted metric for evaluating object detection algorithms. It measures the average precision across multiple detection categories, taking into account precision-recall curves for each category and provides a comprehensive assessment of detection performance by considering both precision (the ratio of correctly detected objects to all detected objects). The fitness function consisted weighted valuesL 0.1 mAP precision for Intersection over Union (IoU) greater than 50%, 0.9 value of mAP for average coverage between 50%-90%. The 5 generations thus defined are now sorted based on a weighted random order, with the weight defined by the reduced value of the adaptation function from the previous step. The best adapted vector from the list is selected for possible mutation. For each hyperparameter, there is an 80% chance with a variance of 0.04 of mutation occurring to create new offspring based on a combination of the best parents from all previous generations. It is assumed for this model, in order for genetic algorithms to succeed in their intended optimization goal, that the minimum number of iterations is 300.

E. Dataset and hardware resources used

The private training set used in the work [8] was used to teach the model. The dataset consists of 460 images, each stored in a lossless format, ensuring preservation of visual information. The images have an average resolution of 512x512 pixels, offering sufficient detail for fine-grained analysis. A comprehensive set of annotations accompanies each image, providing ground truth labels for object presence, object categories, and relevant attributes. The annotations were carefully annotated by domain experts to ensure accuracy and reliability. It contains two classes of objects: cervids (*Cervidae*), included species such as red deer (*Cervus elaphus*), european roe deer (*Capreolus capreolus*) and fallow deer (*Dama dama*), and swine, mainly included photos of European wild boar (*Sus scrofa*). The collection was expanded compared to the previous survey counting 280 of the ”deer” class and 180 photos of the ”wild_boar” class. The dataset was split into three groups of images: training set (70%), validation set (20%) and test set (10%).

Preprocessing plays a vital role in preparing gray scale images for subsequent analysis. It involves applying a series of techniques to enhance image quality, reduce noise, and highlight relevant features. Several steps was performed for improving quality of images. To improve the visual quality and distinguishability of image features it was utilized contrast enhancement. It includes histogram stretching and adaptive histogram equalization. On the other hand it was preformed image normalization. This technique aims to standardize the pixel intensity values across images, ensuring consistent and comparable data.

Due to the high computational costs associated with training neural networks, it was decided to use the Google Colab application [17]. This platform allows free use of virtual machines that have graphics processing units (GPUs), which enable rapid training of the implemented solutions. The model was trained using an Nvidia Tesla T4 graphics card with 16 GB of memory. The YOLO5s model, which is slightly inferior in terms of achieved final results and has a considerably smaller architecture, was selected for testing purposes. However, it offers the advantage of significantly lower computational cost during training and shorter inference time.

III. Experiment

In the case of the evolutionary algorithm, the adaptation function was tested after 10 iterations performance. Then based on values from fit functions, crossover and mutation operations were performed. The results of the best fitted vector was stored in memory. The process of training the evolutionary set itself was performed on 300 cycles involving crossover and mutation. As a result, the obtained set was used in transfer learning with parameters assumed for grid search approach. The final results from evolution process was graphs depicted in Figure 2.

For the Grid Search approach each set of predefined hyper-parameters was tested by training the network model for 100 epochs. It was predefined value, checking performance of each set of parameters on the same number of iteration. The values

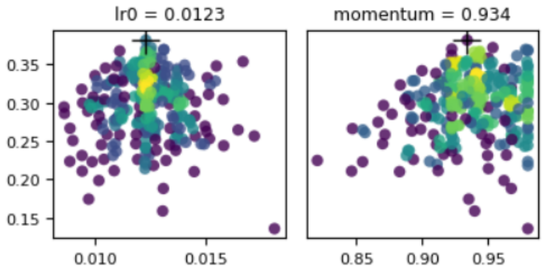

Fig. 2. Population of obtained results for learning rate and momentum - showing fitness (y axis) vs hyperparameter values (x axis). Yellow color indicates higher concentrations.

of the summary error and a metric describing the accuracy of the solution - the mAP - were measured on the test set, as shown in the graphs in Figure 3. The number of iterations was adopted experimentally on the basis of observed training runs. The use of the genetic algorithm's space search alone yielded results close to the original (actually default) values of the hyperparameters used for training. The mAP value did not get significantly better values.

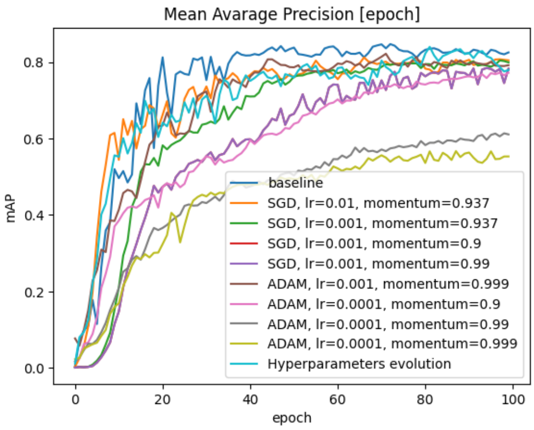

Fig. 3. Changes in the value of the mAP metric over the learning curve for different values of the post-primary hyperparameters.

Another aspect, was compared was the training of the network from the perspective of minimizing the value of the loss function, which is depicted in Figure 4. The Ultralytics library allows these values to be broken down into the following types of errors: bounding box regression loss function, classification loss function, and confidence loss function. The authors of [18] treat the total loss in an aggregate way, which is also used in this paper. Discussing the nature of the training series in detail, it should be noted that the reality did not differ from initial expectations. First, as the initial value of the learning rate decreased, the convergence achieved was slower, and the oscillations during the process itself were smaller. Moreover, it happened, the solution remained in a local optimum from which it was impossible to get out. It should be noted that the learning waveforms on the basis of the analyzed data series, do not exhibit significant quantitative differences except for a single case, which does not translate at all into better qualitative values measured by detection efficiency.

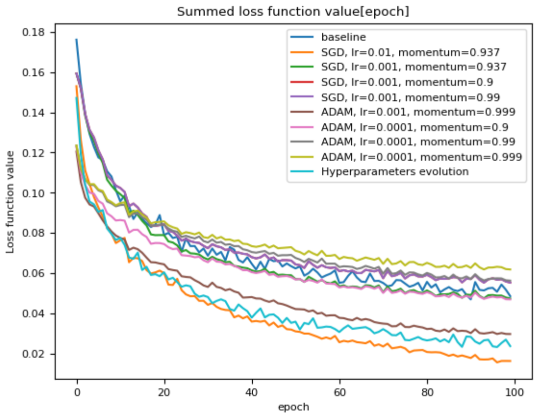

Fig. 4. Change in the value of the summed loss function over the learning.

The mAP precision values obtained by the solution were lower than those obtained on the YOLOv3 architecture by an order of magnitude of about 10% relative to, for example, the [5] and [8] papers. To check whether the under performance is due to the architecture, the YOLOv5l architecture was also trained, which, according to the authors, obtained on the benchmark dataset results about 10% better. The experiment prefaced in the figure 5 showed that in this case the architecture has no particular effect on the results - the sheer value obtained for the smaller architecture is also a barrier to the possibility for the larger one.

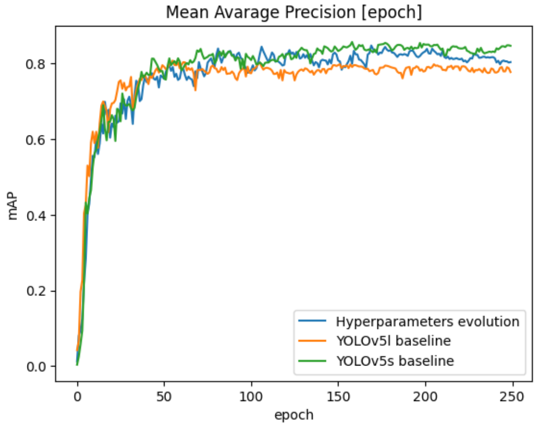

Fig. 5. Summary of the results obtained during training for more epochs. It can be observed the flattening, appearing around epoch 80, regardless of the model adopted.

The experimental results demonstrate the YOLOv5 model's effectiveness in detecting and localizing wild animals in various natural environments. The visualizations reveal accurate bounding box placements around the animal instances, with high confidence scores indicating reliable detections. The results depicted in figure 6. The main errors that occurred during the inspection of the final visualizations were: false detection of objects heated by the sun (stones, trunks), failure to completely cover the silhouette of the animal by the bounding-box due to inhomogenity of animal coats brightness.

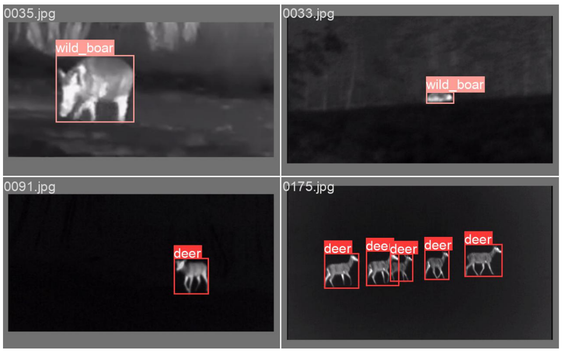

Fig. 6. Example of visualisation on test split of dataset.

IV. Discussion

The research investigated the impact of hyperparameter variations on learning curves and final results. Notably, the study explored different values of the learning rate, momentum, and optimizer types to assess their influence on the model's learning performance. Hyperparameter tuning was carried out using two distinct methods: grid search and evolutionary algorithms. The model's training and evaluation were conducted on an in-house dataset comprising thermal images featuring deer and pig subjects. The trained architecture achieved an impressive Mean Average Precision (mAP) score of 83%, indicating its efficacy in automatic animal detection. The investigation into the impact of hyperparameters on the learning performance of the YOLOv5 architecture revealed significant findings. By systematically testing various combinations of hyperparameter values, we gained insights into their effects on the model's accuracy and detection capabilities. Notably, the learning rate, momentum, and optimizer types played pivotal roles in influencing the model's performance. The selection of appropriate hyperparameters significantly contributed to the achieved high mAP score. The use of both grid search and evolutionary algorithms allowed for comprehensive exploration of the hyperparameter space and facilitated the identification of optimal configurations.

However results gained after training are below expectation, comparing to gathered in table I. The values checked using grid search did not significantly improve the results obtained by the trained models. For the model trained on the default values, performance was better or the same while convergence was achieved with virtually the same number of epochs. This means that the tuning of hyperparameters did not yield results raising the disambiguation of the performative curve. For the genetic model, the generated populations also oscillated around the default values. This may mean, in the space of optimization of hyperparameters, at the very beginning the values were well chosen, and the search of the space was too short to move away from the local optimum. On the other hand, this is contradicted by the fact that after using a larger architecture, the results measured by mAP did not improve. This may mean that for a given dataset we have reached the limits of the model's fitting ability. Only after expanding the dataset will it be possible to obtain better results. As it was already mentioned the primary issues encountered during the inspection of the ultimate visual representations included the erroneous identification of objects heated by sunlight (such as stones and tree trunks) and the inability to fully encompass the animal's silhouette within the bounding box due to variations in brightness.

The results of this research contribute to the field of animal detection in thermal images by showcasing the efficacy of the YOLOv5 architecture. The high mAP score achieved demonstrates the model's potential in accurately detecting animals, highlighting its viability in real-world applications such as wildlife monitoring, environmental protection, and security systems. The success of the YOLO model in this context can be attributed to its object detection capabilities and efficient processing, which are critical for handling thermal image data.

V. Conclusion

This article, was focused on animal detection in thermal images using the YOLOv5 architecture. The primary objective was to develop a high-performance model capable of accurately detecting animals in this specific image modality. Additionally, it was aimed to investigate the influence of different hyperparameters on the learning curves and final results of the model.

To achieve our goals, it was conducted experiments that involved testing various combinations of hyperparameters, including learning rate, momentum, and optimizer types. Two different approaches, namely grid search and evolutionary algorithms, were employed for hyperparameter tuning.

The results of our experiments are partially promising. The trained YOLOv5 architecture achieved a decent Mean Average Precision (mAP) score of 83%. This indicates the model's robustness and effectiveness in accurately detecting animals in thermal images. These findings offer valuable insights into the application of the YOLO model for automatic animal detection in diverse fields, such as wildlife monitoring, environmental protection, and security systems. However despite number of attempts, it was impossible to achieve better results, even with more complex and larger architure.

Our research highlights the significance of hyperparameter selection in optimizing the performance of the YOLOv5 architecture for animal detection. Through systematic exploration of different hyperparameter values, enabled to identify the configurations that yielded the best results. The use of both grid search and evolutionary algorithms provided comprehensive insights into the interplay between hyperparameters and learning performance. Also the comparison of both approaches gave significant insight about optimizing parameters for machine learning models.

Future research in this domain could focus on expanding the dataset to include a broader range of animal species and environmental conditions. Additionally, exploring other state-of-the-art object detection architectures and comparing their performance with YOLOv5 would be beneficial. Despite the promising results, it is essential to acknowledge the limitations and potential areas for future research. Firstly, the in-house dataset used for training and evaluation contained images with deer and pig subjects, which may limit the generalizability of the model to other animal species. Expanding the dataset to include a broader range of animal classes would enhance the model's versatility and applicability in real-world scenarios. Additionally, further investigations into fine-tuning hyperparameters specific to thermal images could potentially improve the model's performance and generalization capabilities. Comparisons with other state-of-the-art object detection architectures would also be valuable to assess their relative performance in animal detection tasks. Moreover, further investigations into fine-tuning the hyperparameters specifically for thermal images may contribute to even higher detection accuracy and generalization capabilities. Also it is considered to create more efficient fitness function, which provides better understanding of tuning hyper parameters using evolution algorithms.

In conclusion, our study demonstrates the successful application of the YOLOv5 architecture for automatic animal detection in thermal images. The achieved high mAP score and the observed impact of hyperparameter tuning techniques affirm the model's potential for various practical applications, including wildlife monitoring, environmental protection, and security systems. This research contributes to the advancement of animal detection technologies and lays a foundation for future studies in this field.

Acknowledgment

The authors would like to thank experts for their appropriate and constructive suggestions to improve this article.

References

[1] J. Witczuk, S. Pagacz, A. Zmarza, and M. Cypel, "Exploring the feasibility of unmanned aerial vehicles and thermal imaging for ungulate surveys in forests - preliminary results. international journal of remote sensing," *International Journal of Remote Sensing*, vol. 39, no. 15-16, pp. 5504–5521, 2018. [Online]. Available: https://doi.org/10.1080/01431161.2017.1390621

[2] A. Vecvanags, K. Aktas, I. Pavlovs, E. Avots, J. Filipovs, B. A., G. Done, D. Jakovels, and G. Anbarjafari, "Ungulate detection and species classification from camera trap images using reti-nanet and faster r-cnn," *Entropy*, vol. 24, no. 3, p. 353, 2022. [Online]. Available: https://doi.org/10.3390/e24030353

[3] M. Choiński, M. Rogowski, P. Tynecki, D. P. J. Kuijper, M. Churski, and J. W. Bubnicki, "A first step towards automated species recognition from camera trap images of mammals using ai in a european temperate forest," pp. 299–310, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-84340-3_24

[4] M. Ivašić-Kos, M. Krišto, and M. Pobar, "Human detection in thermal imaging using yolo," 2019, 5th International Conference on Computer and Technology Applications, pp. 19-24. [Online]. Available: https://doi.org/10.1145/3323933.3324076

[5] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE Access*, vol. PP, no. 3, pp. 125 459–125 476, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3007481

[6] I. R., S. H. Mudumba, and H. R. Adkay, M. Nandi Vardhan, "Human detection in thermal imaging using yolo," 2020, object Detection Using Thermal Imaging, 17th India Council International Conference (INDICON), pp. 19-24, New Delhi, India. [Online]. Available: https://doi.org/10.1145/3323933.3324076

[7] A. Ulhaq, P. Adams, T. Cox, L. T. Khan, A., and M. Paul, "Automated detection of animals in low-resolution airborne thermal imagery," *Remote Sensing*, vol. PP, no. 3, pp. 125 459–125 476, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.3007481

[8] Popek, Ł., Perz, R., and Galiński, G., "Comparison of different methods of animal detection and recognition on thermal camera images," *Electronics*, vol. 12, no. 270, pp. 125 459–125 476, 2023. [Online]. Available: https://doi.org/10.3390/electronics12020270

[9] J. Cilulko, P. Janiszewski, and M. e. a. Bogdaszewski, "Infrared thermal imaging in studies of wild animals," *European Journal of Wildlife Researche*, vol. 59, no. 270, pp. 17–23, 2013. [Online]. Available: https://doi.org/10.1007/s10344-012-0688-1

[10] L. Tan, T. Huangfu, and L. e. a. Wu, "Comparison of retinanet, ssd, and yolo v3 for real-time pill identification," *BMC Med Inform Decis Mak*, 2021. [Online]. Available: https://doi.org/10.1186/s12911-021-01691-8

[11] J. Glen, "Yolov5 by ultralytics (version 7.0) [computer software]," 2014, access: 13.06.2023. [Online]. Available: https://doi.org/10.5281/zenodo.3908559

[12] Isa, I. S., Rosli, M. S. A, Yusof, U. K., Maruzuki, M. I. F., and Sulaiman, S. N., "Optimizing the hyperparameter tuning of yolov5 for underwater detection," *IEEE Access*, vol. 10, pp. 52 818–52 831, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3174583

[13] K. You, M. Long, J. Wang, and M. I. Jordan, "How does learning rate decay help modern neural networks?" *arXiv preprint arXiv:1908.01878*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1908.01878

[14] B. Lim, S. Zohren, and S. Roberts, "Enhancing time-series momentum strategies using deep neural networks," *The Journal of Financial Data Science*, 2019. [Online]. Available: https://doi.org/10.3905/jfds.2019.1.015

[15] T. M. Breuel, "The effects of hyperparameters on sgd training of neural networks," *arXiv preprint arXiv:1508.02788*, 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1508.02788

[16] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," *Knowledge Engineering and Data Science*, vol. 2, no. 1, pp. 41–46, 2019.

[17] E. Bisong, "Google colaboratory. in building machine learning and deep learning models on google cloud platform," 2019.

[18] Q. Xu, Z. Zhu, H. Ge, Z. Zhang, and X. Zang, "Effective face detector based on yolov5 and superresolution reconstruction." *Computational and mathematical methods in medicine*, 2021. [Online]. Available: https://doi.org/10.1155/2021/7748350