This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/UPEC57427.2023.10294447, 2023 58th International Universities Power Engineering Conference (UPEC)

Real-time object detection on high-voltage powerlines using an Unmanned Aerial Vehicle (UAV)

Elisavet Bellou
Electronic and Electrical Engineering Department
Brunel University London, Uxbridge, UK
elizabeth.bellou@brunel.ac.uk

Ioana Pisica
Electronic and Electrical Engineering Department
Brunel University London, Uxbridge, UK
ioana.pisica@brunel.ac.uk

Konstantinos Banitsas
Electronic and Electrical Engineering Department
Brunel University London, Uxbridge, UK
konstantinos.banitsas@brunel.ac.uk

Abstract*— Unmanned Aerial Vehicles (UAVs) are gaining significant scientific interest in critical infrastructure inspection due to their flexibility, cost-effectiveness and advanced computer vision capabilities. This research focuses on high-voltage powerline surveillance, where automatic inspection is a priority for grid companies to prevent power failures. To address the need for real-time detection with limited computational power, we evaluate the recently developed object detection algorithm, YOLOv5. We propose a fine-tuned model trained on a custom dataset to detect key components, i.e. towers, insulators and conductors. The proposed method achieves an overall accuracy rate of 82.3% (mAP@0.5) and enables real-time detection, demonstrating its suitability for inspection tasks and visual-based navigation. Our model was also tested on a custom-built quadcopter with an Nvidia Jetson Nano (4GB) on board, achieving a frame rate of 33fps on live video under real environmental conditions.

Keywords—Unmanned Aerial Vehicles (UAVs); high-voltage powerlines; computer vision; object detection; custom dataset.

I. INTRODUCTION

The energy sector, as defined by the U.S. Presidential Policy Directive (PPD-21) and the European Commission directives, is considered uniquely critical as it provides an "enabling function" across all critical infrastructure sectors [1], [2]. Within this sector, electricity facilities play a vital role in supplying energy to entire countries. As such, safety and security through regular inspection and surveillance is a priority for grid companies. Safety includes the prevention of unexpected faults or energy loss, which could be caused by external factors such as extreme weather or system failure. Currently, the most commonly used methods for powerline inspection are walking patrols, helicopter-assisted inspections, or ground vehicle inspections. However, aerial inspection using unmanned aerial vehicles (UAVs) technology is rapidly gaining popularity as an alternative to these conventional methods, which are costly, time-consuming, and prone to human error. UAVs offer a unique combination of aircraft mobility and quasistatic positioning by hovering for close-up inspections, potentially making them versatile and very efficient means of inspection, especially for locations that are difficult to reach otherwise [3]. To achieve improved detection speed and accuracy, UAV applications have been combined with significant advancements in image classification and real-time object detection using deep learning methods, such as Convolutional Neural Networks (CNNs) [4]. While traditional visual-based methods of image classification have been used in powerline inspection, there has been a growing trend towards the use of deep learning techniques, with the majority of recent work being published after 2018. However, despite the significant progress in the field of computer vision and deep learning, there are still many challenges when it comes to flying robots. The complexity of an aerial platform, with limited computational capacity and restrictions in terms of size, weight and power source, demands a combination of different research fields in order to achieve a realistic result. This includes aerodynamics, sensors, power supply, autopilot logics, computational and payload capabilities, as well as communication links. This work is part of a project that aims at visual-based navigation over transmission line facilities in order to conduct automated inspection of high –voltage powerlines (>500kv). In order to reach this goal, the first crucial step is detecting the main components of such a powerline network. High-performance results can serve as a solid foundation for the development of a complete powerline inspection system that could track towers and powerlines to navigate autonomously. This paper examines and presents the performance of the object detection algorithm YOLOv5, developed by Ultralytics, chosen for its high accuracy and speed compared to other computer vision techniques. YOLO [5] stands for You Only Look Once, because unlike previous regional neural networks (RCNNs), where the network made predictions after dividing the image into several regions of interest (ROIs) to extract features, in YOLO the input image is passed through the network only once in order to make predictions. This offers fast predictions achieving real –time object detection with a video stream with less than 25ms latency. YOLO had been under several improvements, YOLOv2 [6], YOLOv3 [7], YOLOv4 [8], reaching YOLOv5 [9]. The main difference of the fifth version is the depth and width of the layers and the parameters of the model. YOLOv5 is smaller and faster than previous versions, utilizing five different-sized models that vary based on modifications in layer width and depth. These models include the "nano," "small," "medium," "large," and "xlarge" variants. Using these configurations, we train and evaluate fine-tuned models, incorporating internal data augmentations, to predict three object classes of high-voltage powerline facilities: towers, insulators, and conductors. Results are evaluated on GPU environment, as well as in real flight with a custom-built

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/UPEC57427.2023.10294447, 2023 58th International Universities Power Engineering Conference (UPEC)

quadcopter equipped with a Pixhawk4 flight controller and Nvidia Jetson Nano (4GB) as companion computer with a web camera.

II. Related work

Deep learning methods to detect electrical components and perform fault diagnosis attracted researchers' interest during the last five years, due to computer vision technology advantages, such as multiple object detection with high accuracy and speed. Reference [10] was a pioneer work in the field, as CNN was used in order to classify the status of insulators. CNN model with multi-patch feature extraction method is applied to represent the status of insulators and a Support Vector Machine (SVM) is trained based on these features. Dataset included images of normal insulators and three types of damages. A RCNN architecture after image preprocessing (correction, cropping and enhancement), to detect cracked insulators was proposed in [11], based on Region-CNN approach. The same problem of identifying broken insulators was studied previously in [12]. The author combined Faster-RCNN architecture with U-Net to detect self –blast glass insulators. High accuracy results (>90% mAP) proved that deep learning has the potential to achieve a fully automated powerline inspection. Seeking to increase the speed of the models, single-stage algorithms gained interest. YOLOv3 for detecting and classifying distribution line poles and YOLOv2 for detecting tower components, were proposed in [13] and [14], respectively, comparing these models to Faster-RCNN. YOLOv2 and YOLOv3 outperformed Faster-RCNN in terms of mAP and detection speed in both cases. An interesting work for deploying YOLO algorithm on a UAV was presented in [15]. YOLOv4-tiny was tested with different Single Board Computers (SBCs), Nvidia Jetson Nano, Nvidia Jetson TX2, Nvidia AGX Xavier and Raspberry Pi 4, resulting at real-time detection speed with AGX Xavier. The latest version, YOLOv5, was fine-tuned on custom dataset and evaluated for detecting normal and defective insulators in [16]. Regarding other small transmission line objects, such as dampers, spacers and adjusting plates, an optimized YOLOv5 algorithm was proposed in [17]. Our work contributes to the existing literature by extending detection to high-voltage powerline elements, i.e. conductors and towers on a new dataset created from scratch.

III. Data description and methodology

A. Dataset

For this research an original dataset of 2056 images capturing components of transmission lines, was developed for the neural networks' training and testing (data split$\approx$ 70%-20%-10%). The images consist of both aerial and ground videos/images, taken from a DJI Mavic 2 Zoom quadcopter (CMOS 1080p, 30fps) and from a 64MP conventional camera, accordingly. To capture real flight parameters with diverse backgrounds and different points of view, angles and sun position of the shots differ (Figure 1). To further ensure diversity in terms of backgrounds and lightening conditions, the shooting took place in different locations and times during the day (before and after noon).

a) Downside up b) Top-down c) Side

Fig. 1: Sample images of the dataset

Images have been cropped, rotated and resized to 512x640 (original size 1080x1920) in order to simplify the computational procedure. No other preprocessing was conducted (eg. contrast adjustments and Gaussian filters, which enhance features and reduce noise), as according to literature it is not proved that preprocessing techniques have any added value to CNN models. In fact, Dodge & Karam (2018) in [18] showed that in some cases they affect the predictions negatively. However, contrast adjustments, as a data augmentation technique, are applied internally as part of the training code, along with other techniques, such as hue, saturation, brightness, adding noise, mosaic etc., increasing this way, the model's performance and ability to generalize; which was our main goal.
All objects of the above dataset where manually annotated using polygon shape with the open source annotation tool CVAT. Totally, 1756 towers, 4150 insulators, 3734 conductors were annotated. We extracted segmentation pixel coordinates and bounding box coordinates of each object which were used for the model training procedure.

B. Schematic representation of methodology

In order to train, evaluate and finally select a model to be deployed on the UAV for in-flight testing, we followed certain steps. The main concept of this work is presented in Figure 2.

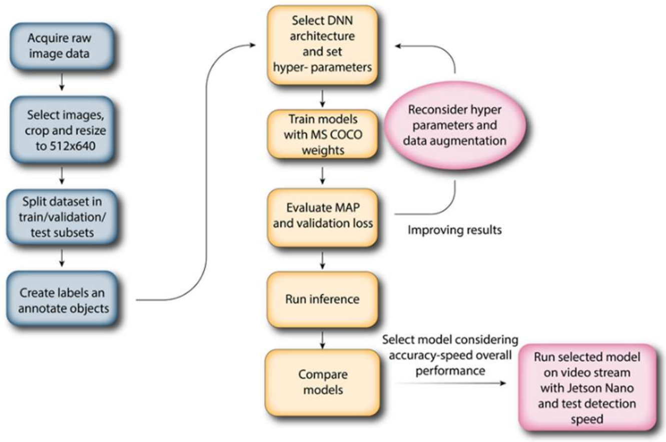


Fig. 2: Schematic representation of the overall methodology

C. Algorithm architecture

The architecture of single-stage object detectors, such as YOLO, consists of three parts: a Backbone, a Neck and a Head, as Figure 3 suggests. YOLOv5 architecture includes a Cross Stage Partial

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/UPEC57427.2023.10294447, 2023 58th International Universities Power Engineering Conference (UPEC)

Network of Darknet as backbone, where the input of the image data takes place for feature extraction, the Path Aggregation Network (PANet) as neck of the network to create feature pyramids, where feature fusion takes place and the head of the network or YOLO layer, which provides three different sizes of feature maps for multi-scale prediction [19]. Anchor boxes are used to output classification predictions, objectness score and bounding boxes (location). In our methodology we use YOLOv5 pre-trained models as a starting point to build up our training. Transfer learning as this training methodology is called, takes advantage of other models, trained on large open-source datasets, such as ImageNet, GoogleNet, MS COCO dataset etc. following different architectures. This method is more effective compared to training from scratch, in terms of computational efficiency. It also demonstrates better results when a small dataset is available. In our case we use MS COCO pre-trained weights; a large –scale object detection dataset consisted of 330K images and 80 object categories. The pre-trained models used for our YOLOv5 models are listed in Table I and can be downloaded using the developers' public repository[1].

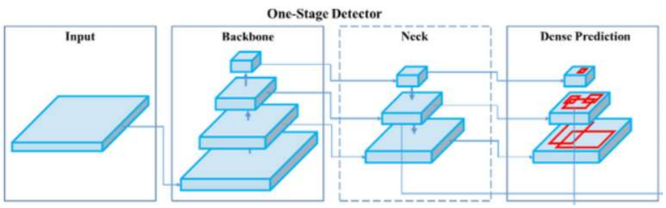


Fig. 3: Single-stage object detectors' architecture [8]

TABLE I. PRE-TRAINED MODELS USED FOR TRANSFER LEARNING

Model	Model architecture	Pre-trained model	Size
	yolov5s	yolov5s.pt	14.1MB
YOLOv5	yolov5m	yolov5m.pt	40.8MB
	yolov5l	yolov5l.pt	89.3MB
	yolov5x	yolov5x.pt	166MB

D. Hyperparameters' tuning and training

Initially, we conducted training for all YOLOv5 models using the default hyperparameters and internal data augmentations provided in the original code. Based on the mean Average Precision (mAP) and the behavior of training and validation loss over the training epochs, we deemed it necessary to fine-tune some of the main hyperparameters of each model using a sequence of grid searches. To accomplish this, we first defined a set of hyperparameters to be optimized, i.e. initial learning rate, weight decay and momentum, then we created a grid of possible values for each hyperparameter. This grid represented all the possible combinations of hyperparameters that we evaluated. For example, for three potential values for each of the three hyperparameters, we had to evaluate a total of 27 combinations. We trained and evaluated the model for each combination, finding that the main hyperparameter affecting results up to approximately 2% in mAP, was the initial learning rate (lr0), while other hyperparameters could remain at default values. Amongst different values of the initial learning rate, within the range of [0.0001, 0.01], the value of 0.001 for medium, large and xlarge sizes and 0.009 for small size, produced the best accuracy results. A tendency of the objectness branch to overfit too early (<100 iterations) was also noticed in all models. This was prevented by setting a lower initial learning rate for medium large and xlarge models and by raising some augmentation values in the original script. Particularly, hyper-parameters for high augmentations script were used for all models, adding to default settings, image rotation degrees and up-down image-flip. Any additional external augmentations applied had a negative effect on mAP. The final powerline (PL) models of yolov5x and yolov5l were trained for 150 epochs, batch size 16 and 32, respectively and 200 epochs for yolov5s and yolov5m with a batch size of 64 and 32, respectively. Information regarding the training environment is listed in Table II.

TABLE II. TRAINING ENVIRONMENT

Model	Dependencies/hardware	version
	python	3.7.13
YOLOv5	Torch	1.11.0
	CUDA	11.3
	OpenCV	4.1.3
	GPU	P100 16GB (google colab)
	CPU	AMD Ryzen 5 2500U 8GB RAM

E. UAV platform

The project included the development of a UAV platform, based on flexibility and simplicity, thus a custom built quadcopter UAV was constructed for on-board object detection processing. A quadcopter or quadrotor UAV, as its name suggests, has four rotors mounted on the frame with fixed-angled propellers. A typical quadcopter has either "x" configuration or "+" configuration (Figure 4) and rotors move clockwise (CW) and counter-clockwise (CCW) in pairs of two [20]. By certain acceleration on rotors it conducts motion on a 3-axis scheme, each of them representing Pitch, Yaw, Roll.

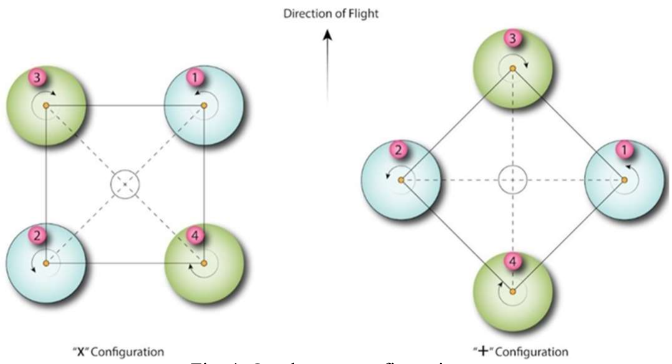


Fig. 4: Quadcopter configuration

[1] Jocher, G. (2020). YOLOv5 by Ultralytics [Computer software]. https://github.com/ultralytics/yolov5

The initial design phase was based on determining the motor's thrust/power and estimating the aircrafts maximum take-off weight, as [21] suggests.

In Table III the final selected parts of our quadcopter are listed and photos of the platform provided in Figure 5. The Autopilot system consists of Pixhawk 4, as flight controller and a Jetson Nano (4GB) Developer kit, as companion computer with on-board communication via MAVlink protocol. Communications where established via Radio Frequency (RF) for telemetry and command link between the aircraft and the Remote Control (RC) transmitter (433MHz), while video streaming was enabled via Wifi connection (2.4GHz). Mission Planner software was used as Ground Control Station (GCS).

TABLE III. COMPONENTS OF THE QUADCOPTER MODEL WITH WEIGHTS

component	type	weight
Payload	Webcamera (HD)	40gr
Autopilot	Pixhawk 4 (flight controller) and Jetson Nano (companion computer)	185,8 gr
Battery1	Li-ion 10000mAh for Jetson Nano	165gr
Battery2	LiPo 4S 3700mAh	372gr
Frame	S500 (composite material) and carbon fiber landing gear	590gr
MotorX4	2212 930KV	208gr
PropellersX4	8045" carbon fiber	72gr
Other	Power Management board, ESCs 30A, RC receiver, connectors.	200 gr
Total		≈1850gr

Fig. 5: Quadcopter aerial platform with the components listed in Table III.

IV. RESULTS

A. Metrics

The first evaluation is conducted based on the Loss Function of the model. YOLOv5 calculates loss based on classification scores, objectness scores and regression of bounding box scores. Binary Cross -Entropy with Logits loss function from Pytorch [22] is used for classification and objectness branch, while Intersection over Unit (IoU) calculates regression branch (bbox). IoU is defined by the area where the predicted box and the ground truth overlaps, divided by the total area of both bounding boxes (predicted and ground truth):

$$IoU(b_{pred}, b_{gt}) = \frac{Area(b_{pred} \cap b_{gt})}{Area(b_{pred} \cup b_{gt})} \quad (1)$$

During training, loss on training set and loss on validation set should be reducing, otherwise the model indicates overfitting. In our methodology we selected training iterations based on validation loss behavior. The final evaluation stage of the trained model is defined by the mean Average Precision, Recall and Precision. Precision is calculated using false positives and true positives of the predictions, while recall relies on true positives in relation to false negatives:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Mean Average Precision is calculated by the sum of the average precision for each query:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} APi \quad (4)$$

An IoU of 50% is the minimum acceptable to evaluate the accuracy for the majority of the models, indicated as mAP@.5. We also evaluate the average mAP of all IoUs (mAP@0.5:0.95).

B. Evaluation of accuracy and speed

Table IV shows mAP score of each PL-model variation, i.e. "s", "m", "l" and "x", as well as speed in validation dataset. Figure 6 also provides graphs of mAP, Precision and Recall for all models. YOLOv5s proved the fastest and also achieved the highest mAP@0.5 score. Therefore, it was further optimized with Tensor-RT library (Pytorch), reaching 94.3% mAP score for towers' detection. Inference of the optimized model on test images with predictions is demonstrated in Figure 7.

TABLE IV. YOLOV5 PERFORMANCE WITH TESLA P100 GPU (16GB) AND YOLOV5S MODEL OPTIMIZED WITH TENSOR-RT ACHIEVING 94.3% ACCURACY (MAP@0.5) FOR TOWERS DETECTION. INPUT IMAGE SIZE 640X640. NMS TIME PER IMAGE ≈ 1-1.5MS (NOT INCLUDED).

PL-Models	Precision	Recall	mAP@ 0.5	mAP @.5:95	FPS	ms
YOLOv5s	0.807	0.78	**0.82**	0.607	303	3.3
YOLOv5m	0.8	0.78	0.819	0.597	119	8.4
YOLOv5l	0.787	**0.788**	0.817	0.613	81.9	12.2
YOLOv5x	**0.825**	0.763	0.818	**0.635**	45.4	22.0
YOLOv5s-TRT	0.817	0.786	**0.823**	0.616	**303**	**3.3**
Tower	0.94	0.903	0.943	0.832		
Insulator	0.787	0.909	0.889	0.634		
Conductors	0.723	0.547	0.636	0.382		

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/UPEC57427.2023.10294447, 2023 58th International Universities Power Engineering Conference (UPEC)

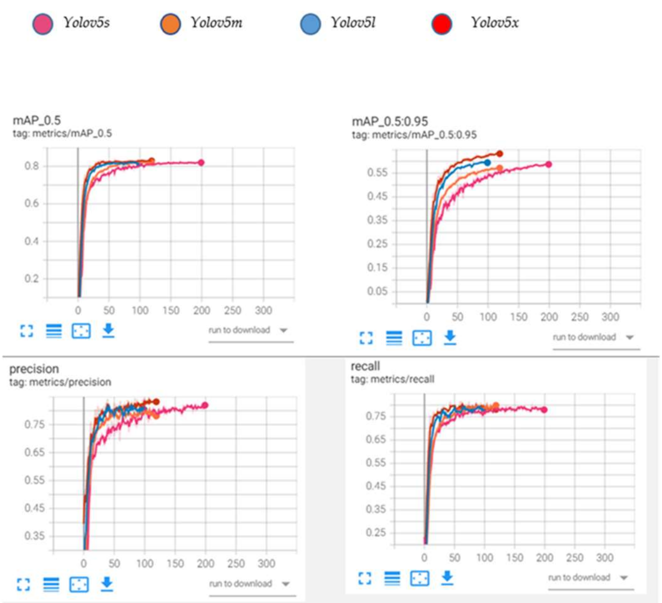
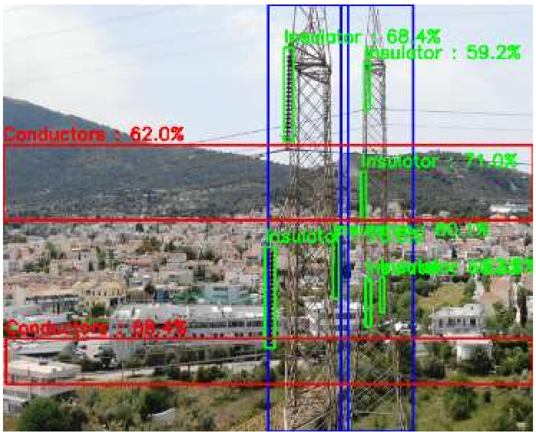
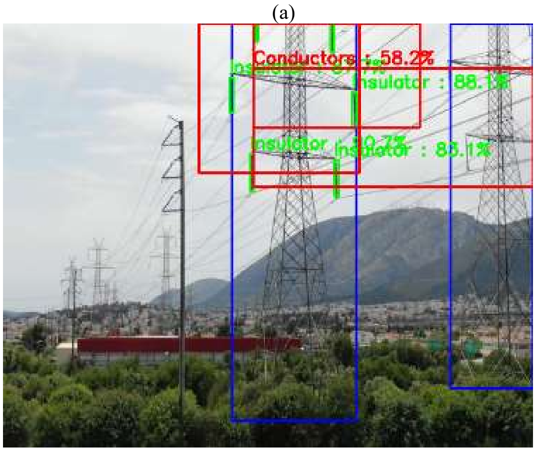


Fig. 6: Graphical depiction of Precision, Recall and mAP results



(a)



(b)

(c)

Fig. 7: Yolov5s-RT inference on test images, indicate high confidence level predictions with complex background. Towers and conductors shown further away in (b) are not detected, however it is encouraging that the model did not "confuse" the wooden pole's components with the model objects. In (c) the model does not detect all insulators, due to the shot angle combined with the complex background, however it successfully detects conductors even on the lower right side of the image.

C. *Flight test findings*

Speed and size are crucial parameters to consider when choosing a model for testing in flight, taking into account hardware capabilities. Since we observed only small differences in accuracy among our PL-models, we selected the YOLOv5s-TRT model for the experiment. Recall was also taken into consideration since it indicates false negatives which are significant for powerline inspection. The flight took place in Greece, in an urban environment of eastern suburbs of Attica, in accordance with EASA regulations (945/2019 and 947/2019) and Hellenic Civil Aviation Authority guidelines. With a temperature of 28^{o} C and steady north wind of 7 m/s with gusts up to 14 m/s, the quadcopter faced no issues experimenting with different powerline approaches, however stability loss and vibrations on throttle changes were noticed, when flying at >30m altitude, affecting the video stream on the ground control station. This means that several hardware improvements need to be made in order for the UAV to be suitable for a typical inspection scenario, where flexibility is crucial, in terms of different altitudes and fast direction changes. Results on the live video and default input image size at 640x640 showed a significant drop in speed at approximately 10-12fps which makes it hard for the UAV to adapt in real-time applications. Reducing the input image size of the optimized model at 288x288 we achieved real-time detection (30-33fps) with approximately 5% drop in accuracy (mAP@0.5). The experiment also demonstrated the successful real-time detection of towers, even from distances of up to 200m. This achievement is significant as it represents the initial step towards autonomous navigation, wherein the UAV can detect the tower from the takeoff point and subsequently approach for further inspection of insulators and powerlines.

V. Conclusions and Future Work

In this study, we fine-tuned and trained different models of YOLOv5 architecture for real-time object detection on high-voltage powerline facilities, achieving an overall accuracy rate of 82.3% mAP@0.5, with the highest score noted in towers' detection (94.3%). Our model was tested both on a GPU environment and in a real flight experiment, using a custom-built quadcopter with the Jetson Nano developer kit for on board processing. Our work proves that the fined-tuned YOLOv5s model optimized with Tensor-RT and trained on an original custom –dataset, is capable of deployment on a mini quadcopter with only basic configuration, achieving real-time detection up to 33fps. These findings suggest that YOLOv5 models implemented on UAVs equipped with advanced sensors, (i.e. FullHD optical sensors, LiDAR and thermal sensors) are suited for semi-autonomous navigation and inspection tasks over transmission line facilities, which is subject of future research. In addition to local inspection tasks, it is highly feasible to conduct large-scale inspections by using a number of such UAVs to cover the transmission line network. Furthermore, this work can provide a basis for intelligent powerline surveillance/ inspection tasks and paves the way for future research on fault detection of powerlines by combining computer vision and data generated from different sensors, such as electromagnetic detectors or thermal/UV sensors, aiming at minimizing human intervention during flight.

Acknowledgment

The authors would like to thank "UcanDrone" CEO and his technical team for supporting the flight experiments, as well as drone operator, Mr. Konstantinos Vlamidis and Altus Lsa for both assisting in the dataset collection flights.

References

[1] Cybersecurity and Infrastructure Security Agency (CISA). (n.d.). Energy Sector. Retrieved June 29, 2022, from https://www.cisa.gov/energy-sector.

[2] European Parliament, Council of the European Union. (2006). Directive 2005/89/EC of the European Parliament and of the Council of 18 January 2006 concerning measures to safeguard security of electricity supply and infrastructure investment (Text with EEA relevance) and Regulation (EU) 2019/941 repealing Directive 2005/89/EC.

[3] Amoiralis, E. I., Tsili, M. A., Spathopoulos, V., & Hatziefremidis, A. (2014). Energy efficiency optimization in uavs: a review. In Materials Science Forum, 792, 281-286. Trans Tech Publications Ltd.

[4] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. Neurocomputing, 187, 27-48.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 779-788.

[6] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7263-7271.

[7] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[8] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[9] Jocher, G. et al. (2020). YOLOv5: v3.0. Zenodo. https://doi.org/10.5281/zenodo.3908559.

[10] Zhao, Z., Xu, G., Qi, Y., Liu, N., & Zhang, T. (2016). Multi-patch deep features for power line insulator status classi-fication from aerial images. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 3187-3194). IEEE.

[11] Li, S., Zhou, H., Wang, G., Zhu, X., Kong, L., & Hu, Z. (2018). Cracked insulator detection based on R-FCN. In Journal of Physics: Conference Series, 1069, No. 1, 012147. IOP Publishing.

[12] Ling, Z., Qiu, R. C., Jin, Z., Zhang, Y., He, X., Liu, H., & Chu, L. (2018). An accurate and real-time self-blast glass insulator location method based on faster R-CNN and U-net with aerial images. arXiv preprint arXiv:1801.05143.

[13] Chen, B., & Miao, X. (2020). Distribution Line Pole Detection and Counting Based on YOLO Using UAV Inspection Line Video. Journal of Electrical Engineering & Technology, 15(1), 441-448.

[14] Chen, Q., Gao, Y., Peng, Y., Zhang, J., & Sun, K. (2019). Accurate Object Recognition for Unmanned Aerial Vehicle Electric Power Inspection using an Improved YOLOv2 Algorithm. In 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), 610-617. IEEE.

[15] Ayoub, N., & Schneider-Kamp, P. (2021). Real-time on-board deep learning fault detection for autonomous uav inspections. Electronics, 10(9), 1091-1106.

[16] Feng, Z., Guo, L., Huang, D., and Li, R. (2021). Electrical Insulator Defects Detection Method Based on YOLOv5. IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), 979-984, doi: 10.1109/DDCLS52934.2021.9455519.

[17] Gu, J.; Hu, J.; Jiang, L.; Wang, Z.; Zhang, X.; Xu, Y.; Zhu, J.; Fang, L. (2023). Research on Object Detection of Overhead Transmission Lines Based on Optimized YOLOv5s. Energies, 16, 2706. https://doi.org/10.3390/en16062706.

[18] Dodge, S. & Karam, L. (2016). Understanding how image quality affects deep neural networks. Eighth Interna-tional Conference on Quality of Multimedia Experience (QoMEX), pp. 1-6, doi: 10.1109/QoMEX.2016.7498955.

[19] Xu, R., Lin, H., Lu, K., Cao, L., & Liu, Y. (2021). A forest fire detection system based on ensemble learning. Forests, 12(2), 217.

[20] Kuantama, Endrowednes & Vesselenyi, Tiberiu & Dzitac, S. & Tarca, Radu. (2017). PID and Fuzzy-PID control model for quadcopter attitude with disturbance parameter. International Journal of Computers, Communications and Control.

[21] Sadraey, M. H. (2020). Design of Unmanned Aerial Systems, Design of Unmanned Aerial Systems. Wiley, 35-63.

[22] Pytorch org. Contributors. (2022). BCE with Logits Loss documents. Pytorch website. Accessed 29 June 2022. https://pytorch.org/docs/master/generated/torch.nn.BCEWithLogitsLoss.html.