Deep Learning Models for UAV-Assisted Bridge Inspection: A YOLO Benchmark Analysis

Trong-Nhan Phan 1,2 , Hoang-Hai Nguyen 1,2 , Thi-Thu-Hien Ha 1,2 , Huy-Tan Thai 1,2 , and Kim-Hung Le 1,2

University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
{21522407, 21522034, 21522056}@gm.uit.edu.vn, {tanth, hunglk}@uit.edu.vn

Abstract. Visual inspections of bridges are critical to ensure their safety and identify potential failures early. This inspection process can be rapidly and accurately automated by using unmanned aerial vehicles (UAVs) integrated with deep learning models. However, choosing an appropriate model that is lightweight enough to integrate into the UAV and fulfills the strict requirements for inference time and accuracy is challenging. Therefore, our work contributes to the advancement of this model selection process by conducting a benchmark of 23 models belonging to the four newest YOLO variants (YOLOv5, YOLOv6, YOLOv7, YOLOv8) on COCO-Bridge-2021+, a dataset for bridge details detection. Through comprehensive benchmarking, we identify YOLOv8n, YOLOv7tiny, YOLOv6m, and YOLOv6m6 as the models offering an optimal balance between accuracy and processing speed, with mAP@50 scores of 0.803, 0.837, 0.853, and 0.872, and inference times of 5.3ms, 7.5ms, 14.06ms, and 39.33ms, respectively. Our findings accelerate the model selection process for UAVs, enabling more efficient and reliable bridge inspections.

1 Introduction

Visual inspection, a method using one's naked eye to assess the condition of structures, has played an essential role in maintaining and ensuring the quality of urban public works [1]. This method involves detecting structural issues such as corrosion, cracks, and other problems to ensure the works' safety and durability. For instance, visual inspection is routinely used to inspect the condition of high-rise buildings' envelopes. For bridges constantly subjected to harsh weather conditions and traffic loads, visual inspection is an excellent method for inspectors to detect early signs of deterioration, corrosion, and damage. As a result, experts can suggest timely solutions to maintain and repair these structures. Technology devices, like cameras, digital image analyzers, and other devices, have contributed to improving visual inspection. These tools allow experts to remotely examine components for corrosion and damage, which reduces cost and safety risks for inspectors in visual inspection [2]. Among them, UAVs are commonly used to inspect structures that are difficult to access or have complex geometries. However, the monitoring process by UAVs encounters particular

challenges. First, during the visual inspection process, a large amount of data requires specific expertise to analyze and examine, which is labor intensive [3]. Secondly, inspectors might face low-quality images due to the lack of proper light conditions and wind turbulent disturbance, leading to false detection [4]. Therefore, it is necessary to automate the process with AI-supported to increase accuracy while reducing the inspection time.

The mentioned automatic process can be achieved using computer vision models. Computer vision (CV) is a branch of artificial intelligence that enables computers and systems to extract meaningful insights from digital images, videos, and other visual data sources. Subsequently, they can provide recommendations based on the information they gather. In visual inspection, automation can be done by combining a computer vision model with a camera-attached UAV. This could be a game-changer in the field of visual construction inspection as UAVs have adaptability, easy installation, low maintenance costs, versatility, and relatively small operating cost [5], while computer vision models are robust, easy to train and deploy. In this process, UAVs take pictures of structures. After that, the computer vision model processes the images and gives predictions about what components appear in the images and their possible problems. Then, this information would be sent to the inspectors, which would assist them in assessing the structure's condition and accordingly decide whether the structures need to be repaired or renovated. Currently, deep learning models are also the main topic of this paper.

Although the combination of computer vision models and camera-attached UAV(s) is the future of automated visual inspections, specific problems need to be examined. Despite the rapid advancement of UAVs in recent years, their computational power and storage space are still limited. Meanwhile, in visual inspection, the number of images that need to be processed could grow enormously, and data collection is often repeated since external factors, such as wind-induced vibrations in captured footage or low illumination leading to dark images, can significantly affect the quality of data [6]. The size of images is also a significant problem since processing, storing, or transmitting such a large amount of data is time-consuming and storage-intensive. In the traditional centralized CV-UAV method, these problems are not properly resolved because CV models are placed on a local computer or in the cloud, and UAVs have an additional task to transmit these images to where CV models are placed and wait to receive the results. Having to transmit images between UAVs and CV models is a bottleneck of the process, as it is slow and vulnerable to poor connections and hackers' attacks. Edge AI is a rising technology that can deal with that. It overcomes this problem by embedding CV models into UAVs, so images are processed inside the UAVs, and only the results are sent to inspectors. The results are often text and damaged-predicted images, just a tiny part of the captured images. By doing so, the amount of data needed to be transmitted is reduced by a lot, thus speeding up the process and minimizing the risk caused by poor connections and hackers.

We found that the task of discovering the potential of Edge AI in the field of visual inspection, especially of bridges, is essential [7]. The principal reason

is that bridges are sensitive infrastructure transportation components, so they must be routinely inspected. It is necessary to develop an automatic bridge inspection method combining Edge AI and visual inspections to quickly and correctly assess the condition of bridges while requiring low effort. This method would help the bridge preservation process go smoothly and hence ensure the safety of traffic participants as well as social connectivity. However, incorporating CV models into UAVs is a challenge. This is because some modern CV models achieve high accuracy but require enormous computational and storage capacity, which UAVs lack. On the other hand, some older CV models fit UAVs' computational and storage capacity, but their accuracy is inadequate. Through this work, we aim to find suitable CV models for bridge inspection with UAVs and provide a benchmark for future works. Therefore, in this paper, we focus on experimenting with CV models, comparing their performance through accuracy and inference time on COCO-Bridge 2021+, a dataset for bridge inspection. In detail, we benchmark four YOLO variants, including YOLOv5 [8], YOLOv6 [9], YOLOv7 [10], and YOLOv8 [11]. The experimental results show that YOLOv8n, YOLOv7tiny, YOLOv6m, and YOLOv6m6 are the models that achieve the best trade-off between accuracy and inference time among 23 total models of these four YOLO variants.

2 Related work

The application of machine learning has been gaining significant attention in various fields, including agriculture, visual inspection within the petroleum industry, urban building maintenance, and bridge inspection. When it comes to the first field, agriculture, visual inspection plays an important role in identifying leaf diseases. In an article from 2021 [12], authors used a Vision Transformer model (ViT) integrated with a Drone Pi to categorize and group diseased leaves. The Cassava Leaf Disease Dataset used in this study includes 21,397 photos of cassava leaves in five distinct groups: CBB, CBSD, CGM, CMD (four types of leaf diseases), and healthy leaves. The ViT model produced remarkable results and demonstrated its potential for early identification of leaf disease [13]. In detail, this model observed the highest F1 score of 96% in the detection of CMD. Another work [14] (2019) utilized a deep convolutional neural network (DCNN) to analyze hyperspectral images, taken by UAVs, with high spatial resolution. Specifically, in a well-controlled field experiment, drones gathered pictures on five different days on wheat fields that were either healthy or infected with rust across a whole crop cycle. Then, the model used these photos for calibration and got a high performance throughout the lifespan of plants, especially at the late stages of disease spreading. The proposed model obtained an accuracy, at 85%, better than the conventional random forest classification's performance, at 77%. The two aforementioned works contribute significantly to the advancement of the automation of crop disease detection.

In the area of building inspection, it is clear that using UAVs (Unmanned Aerial Vehicles) would be both cost-effective and safe. A 2021 study [15] pre-

sented an automated inspection system for exterior crack detection on high-rise buildings using UAVs. A deep learning image classification algorithm based on Google's Xception CNN yielded an excellent performance in detecting wall cracks, with a Success Percentage at 80%, where this measure is calculated by the ratio between the number of cracks detected and the total number of cracks. The dataset has 20,001 non-crack images and 20,001 crack images, where each image is 227 × 227 pixels. Another study [16], in 2023, investigated using deep learning, with a small amount of data and transfer learning, to detect cracks in masonry facades. This approach achieved remarkable accuracy and F1 scores, reaching 100% using CNN models, specifically MobileNetV2 and InceptionResNetV2, on the Brickwork Cracks Dataset containing 5,926 images. In 2023, the authors in [17] employed three different deep learning models, ResNet-50, VGG-16, and AlexNet, to detect surface defects and reconstruct the 3D scene. These mentioned studies show the potential to enhance the inspection efficiency in buildings using deep learning models.

Finally, in bridge inspection, visual inspection is a pivotal element. Notably, in a 2018 study by the American Mechanical Engineers [18], a method employing deep learning was suggested for comprehensive maintenance and inspection of bridges based on a dataset containing 3,832 images collected from the internet, including arch, suspension, and cable-staved bridges. This method utilized a fine-tuned AlexNet to classify bridge types, achieving validation and testing accuracies of 96% and 96.6%, respectively. In addition, a Faster R-CNN model based on ZF-net was utilized to identify and locate bridge towers and decks and obtained an mAP of 90.45% on the validation dataset. Detecting concrete cracks in bridges was achieved through a fine-tuned GoogLeNet, attaining high validation and testing accuracies of 99.39% and 99.36%, successively. In 2017, another noteworthy study [19] presented a region-based deep learning strategy for autonomous structural visual inspection. Authors used a Faster R-CNN model to detect different types of damage in real-time, including bolt corrosion, concrete cracks, steel corrosion, and steel delamination on a dataset containing 2, 366 images labeled. The result showed impressive average accuracy (AP) ratings for each category, with an overall mean AP of 87.8%. Additionally, a 2019 article [20] introduced a single-stage detector for concrete bridge surface damage detection. The 2,206 inspection photos of Hong Kong's highway concrete bridges were included in the image dataset used for this method. The original YOLOv3 model was enhanced through a novel transfer learning technique, significantly improving detection accuracy to 80% and 47% at the IoU threshold of 0.5 and 0.75, correspondingly. Using deep learning and texture analysis, ASR (Abstract Alkali-Silica Reaction) crack diagnosis is a new advancement in bridge inspection [21]. A dataset containing 35 ASR defect photos from different Queenslandian bridges impacted by ASR was produced for this investigation. The integration of texture morphology and the InceptionV3 model led to a remarkable validation accuracy of 94.07%. These results demonstrate the feasibility of machine learning for bridge inspection, especially damage detection.

In conclusion, our comprehensive review of research underscores the impactful role of machine learning and deep learning methods in advancing visual inspection processes across various industries. These state-of-the-art technologies not only excel in early identifying potential issues but also significantly enhance safety, productivity, and overall infrastructure quality.

3 Models and Evaluation methodology

This section serves as the cornerstone of our study, outlining the models and evaluation methodology. Regarding models, we introduce the four variant models of YOLO from version 5 to version 8 in Section 3.1. For evaluation methodology, we present the experimental dataset, followed by implementation tools and model selection criteria, and end up with evaluation metrics in Section 3.2.

3.1 Deep learning models

Nowadays, two types of object detectors may be distinguished: one-stage detectors and two-stage detectors. In comparison to one-stage detectors, two-stage detectors frequently reach higher accuracy (both recognition and localization accuracy) and have slower inference times. Our work focuses entirely on one-stage detectors since they are suitable for real-time object detection. Models tested in our work include the four newest: YOLOv5, YOLOv6, YOLOv7 and YOLOv8. Apart from the fact that they are state-of-the-art models for Object Detection, all of them have controversial performances, with opposite results in different scenarios. Thus, those models' efficiency in the visual inspection of bridges is questionable and needs to be studied thoroughly.

- YOLOv5: YOLOv5 was released in June 2020 by Glenn Jocher [8]. YOLOv5 built upon many of the improvements made in YOLOv4, including flexible control of model size, application of Hardswish activation function, and data enhancement [22], but the major difference is that it was built based on PyTorch framework instead of DarkNet. From the literature, we found that YOLOv5 is better than most SOTA on various datasets [23–25]. However, some works show contradictory results. For instance, YOLOv4 yields better AP50 than YOLOv5 on a custom faulty components detection of electric poles dataset [26].
- YOLOv6: YOLOv6 was released in 2022 by Meituan Vision AI Department [9]. It has many improvements in backbone, neck, head, and training strategies, with the most noticeable being its anchor-free paradigm. Previous works show that, compared to YOLOv5, YOLOv6 lacks stability and flexibility but makes up for impressive capabilities in detecting small, densely packed objects. Official YOLOv6 papers show that YOLOv6 outperforms YOLOv5 and all previous SOTA in almost all categories [9,27]. Meanwhile, on a sampled COCO train2017 dataset, YOLOv6 has a higher recall and

mAP@50:95 while having worse precision, mAP@50, and inference speed compared to YOLOv5 [28].

- YOLOv7: YOLOv7 was released in 2022 by Alexey Bochkovskiy et al. [10]. YOLOv7's architecture has three main features: E-ELAN for efficient learning, model scaling for varying sizes, and the "bag-of-freebies" approach for accuracy and efficiency. In fact, YOLOv7 is promising as many works show that it is better than all previous SOTA, especially in terms of accuracy [10, 29–33].
- YOLOv8: YOLOv8 was released in 2023 by Ultralytics, the company behind YOLOv5 [11]. YOLOv8 is anchor-free, predicts fewer boxes, and has a faster non-maximum suppression (NMS) process. As YOLOv8 was introduced recently, there are not many works studying it in the literature. Overall, YOLOv8 seems better than YOLOv5 (except for training speed) and has similar performance compared to YOLOv7 [34].

3.2 Evaluation methodology

Evaluation dataset The evaluation dataset used in this paper is COCO-Bridge 2021+ Dataset authored by Eric Bianchi and Matt Hebdon. The Virginia Department of Transportation (VDOT) provided real structural bridge inspection records, which served as the origin of the dataset used for bounding box detection. In specifics, this dataset is made up of 1470 annotated structural bridge photos, of which about 10% are from the test set (136 images) and 90% are from the training set (1321 images). The dataset contains information on four structural bridge details: out-of-plane stiffeners, cover plate terminations, gusset plate connections, and bearings. The dataset's sample photos for each category are displayed in Figure 1.

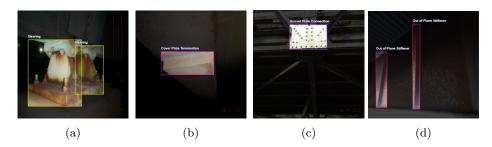


Fig. 1: Sample images of each structural bridge detail: (a) bearing; (b) cover plate terminations; (c) gusset plate connections; (d) out-of-plane stiffeners.

Implementations tools All experiments were carried out using an identical computer with an x86_64 CPU architecture, CUDNN version 9.2.1, and two CUDA devices. The GPU used was an NVIDIA GeForce RTX 3060 Ti with approximately 8.36 GB of memory. The software versions used were Torch version 2.0.1+cu117, Torchvision version 0.15.2+cu117, and Python version 3.9.18.

Evaluation Metrics In this section, we introduce the purpose of evaluating the performance of our model in detecting the details of the bridge. It is important to note that accurate evaluation is crucial to ensure the model works effectively in real-world applications.

- Intersection over Union (IoU): This measures the overlap between the predicted bounding box and the ground truth bounding box. IoU can be formulated as follows:

$$IoU = \frac{Area_of_Overlap}{Area_of_Union} \tag{1}$$

- Average Precision (AP): This summary of the shape of the Precision-Recall curve to a single number simplifies the model evaluation task. All models in our work use Object-Detection-Metrics's Every Points Interpolation AP.
- Mean Average Precision (mAP): This is a popular metric to measure the performance of object detection models. It calculates the average of average precision (AP) over all classes. There are two mAP metrics used in our work: mAP@50 and mAP@50:95. mAP@50 calculates the mAP at an IoU threshold of 0.5, while mAP@50:95 consider different IoU thresholds from 0.5 to 0.9.
- Inference Time: Refers to the duration it takes a model to process an input and generate a corresponding output. This encompasses the entire process, from when the model receives the input to when it delivers the final result. This metric is crucial for real-time applications and is influenced by both the model's structure and the available computational resources.
- Number of parameters: The number of parameters can affect both the computational efficiency and the capacity of the model. A model with too many parameters may overfit the training data, while a model with too few parameters may not be able to learn the underlying patterns in the data.
- GFLOPs: This stands for Giga Floating Point Operations, and it measures the computational complexity of the model. It is the total number of floating point operations (additions, multiplications, etc.) required to perform a forward pass through the model. A model with fewer GFLOPs is generally faster and consumes less energy.

4 Experimental Results

In this section, we describe our experimental results on the COCO-bridge-2021+ dataset. Table 1 illustrates the input size, the number of parameters, and GFLOPs

C	1. The sum.	nary or models s mp	ut size, paran	i, and Gri
	Model	Input size (pixels)	Params (M)	GFLOPs
	YOLOv5n	640 x 640	1.76	4.1
	YOLOv5s	640 x 640	7.02	15.8
	YOLOv5m	640×640	20.87	47.9
	YOLOv5l	640 x 640	46.12	107.7
	YOLOv5x	640×640	86.19	203.8
	YOLOv6n	640 x 640	4.63	11.3
	YOLOv6s	640×640	18.50	45.2
	YOLOv6m	640 x 640	34.80	85.6
	YOLOv6l	640 x 640	59.54	150.5
	YOLOv6n6	1280×1280	10.34	12.4
	YOLOv6s6	1280×1280	41.32	197.5
	YOLOv6m6	1280×1280	79.53	378.8
	YOLOv6l6	1280×1280	140.21	672.5
	YOLOv7tiny	640 x 640	6.02	13.0
	YOLOv7	640×640	36.50	103.2
	YOLOv7x	640 x 640	70.80	188.0
	YOLOv7w6	1280×1280	69.81	88.6
	YOLOv7e6	1280×1280	110.39	143.6
	YOLOv7d6	1280×1280	152.90	197.4
	YOLOv8n	640 x 640	3.00	8.1
	YOLOv8s	640 x 640	11.12	28.4
	YOLOv8m	640 x 640	25.84	79.3
	YOLOv8l	640×640	43.60	164.8

Table 1: The summary of models's input size, param, and GFLOPs.

of YOLO variants, indicating that models are available in a broad range of input sizes and complexity. In detail, models take 640x640 pixels or 1280x1280 pixels input sizes and have parameter numbers in the approximate range of 1.76M - 152.90M and GFLOPs in the range of 4.1 - 672.5. The input size of 640x640 pixels is enough for the models to detect most objects in an image, but the larger input size of 1280x1280 pixels may help in the case of blurred images or far-away objects at the cost of higher model complexity.

As shown in Table 2, mAP@50:95 and mAP@50 fluctuated from 0.401 to 0.475 and from 0.790 to 0.872, respectively, while the inference time ranged from 5.3 ms to 57.46 ms. Based on mAP@50 and mAP@50:95, YOLOv6m6 obtained the highest accuracy, at 0.872, 0.475, while YOLOv8n observed the fastest inference time with 0.803. According to Figure 2, which shows the trade-off curve between mAP@50 and inference time, the four best models were chosen, including: YOLOv8n, YOLOv7tiny, YOLOv6m, YOLOv6m6.

We then deployed these four best models on edge devices, represented by Jetson Nano, which can attached to UAVs for automatic visual inspection. Figure 3 illustrated that only YOLOv7tiny and YOLOv8n can achieve real-time processing rates of 36.308 and 58.272 images per second, respectively. Meanwhile, YOLOv6m6 and YOLOv6m have better mAP but reach limited processing rates at 1.246 and 5.512, in that order.

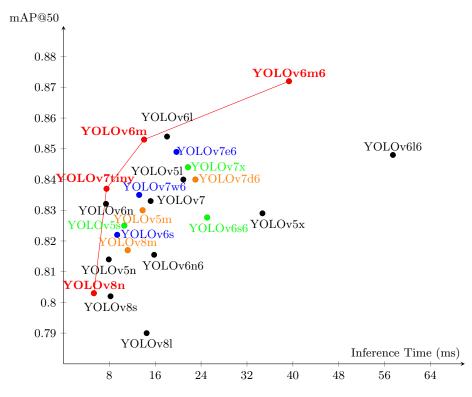


Fig. 2: Precision-inference time curve of experimental models.

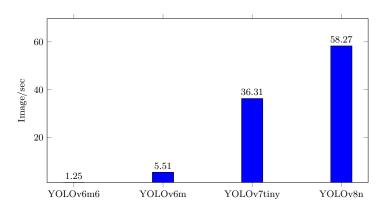


Fig. 3: The processing rates (images per second) comparison of the four best trade-off models on Jetson Nano.

Table 2: Comparing models' performance (inference time (ms), mAP@50 and mAP@50:95).

Model	Inference Time (ms)	mAP@50	mAP@50:95
YOLOv5n	7.9	0.814	0.401
YOLOv5s	10.6	0.825	0.431
YOLOv5m	13.8	0.830	0.449
YOLOv5l	20.9	0.840	0.457
YOLOv5x	34.7	0.829	0.466
YOLOv6n	7.41	0.832	0.458
YOLOv6s	9.36	0.822	0.467
YOLOv6m	14.06	0.853	0.469
YOLOv6l	18.06	0.854	0.473
YOLOv6n6	15.81	0.816	0.457
YOLOv6s6	25.06	0.828	0.449
YOLOv6m6	39.33	0.872	0.475
YOLOv6l6	57.46	0.848	0.474
YOLOv7tiny	7.5	0.837	0.433
YOLOv7	15.2	0.833	0.453
YOLOv7x	21.7	0.844	0.458
YOLOv7w6	13.2	0.835	0.415
YOLOv7e6	19.7	0.849	0.456
YOLOv7d6	23.0	0.840	0.458
YOLOv8n	5.3	0.803	0.442
YOLOv8s	8.2	0.802	0.442
YOLOv8m	11.2	0.817	0.464
YOLOv8l	14.5	0.790	0.438

5 Conclusion

Our research benchmarked the latest models in the YOLO family on the COCO-Bridge-2021+ dataset, partly assisting inspectors in choosing the most suitable model for the detail detection problem. The results point out that YOLOv5n, YOLOv7tiny, YOLOv6m, and YOLOv6m6 are the best models with high accuracy and inference speed. Our work paves the way for a new field: bridge details detection. There is considerable potential for refining and advancing our work. The number of benchmarked models is still limited, and the models' architectures are not optimized for the problem. Future works might explore the potential of various new architectures and modifications on known architectures. A performance improvement can also be made through better preprocessing and feature engineering using domain knowledge. More data can also lead to better real-life performance as it helps improve models' generalization and reduce overfitting. Better bridge detail detection helps contextualize the environment and defects, leading to better visual inspection of bridges.

Acknowledgment

This research is funded by the University of Information Technology-Vietnam National University HoChiMinh City under grant number D1-2024-08.

References

- "Overview of visual inspection." Available at https://inspectioneering.com/ tag/visual+inspection#tabSummary.
- H.-T. Thai, K.-H. Le, and N. L.-T. Nguyen, "Towards sustainable agriculture: A lightweight hybrid model and cloud-based collection of datasets for efficient leaf disease detection," Future Generation Computer Systems, vol. 148, pp. 488–500, 2023.
- 3. "Drones in inspection: Current scenario and challenges." Available at https://www.linkedin.com/pulse/drones-inspection-current-scenario-challenges-clearspot-ai/.
- "Inspection using drone technology." Available at https://forcetechnology.com/en/articles/inspection-using-drone-technology.
- 5. A. I. Hentati and L. C. Fourati, "Comprehensive survey of uavs communication networks," *Computer Standards & Interfaces*, vol. 72, p. 103451, Oct. 2020.
- M. Zakaria, E. Karaaslan, and F. N. Catbas, "Advanced bridge visual inspection using real-time machine learning in edge devices," Advances in Bridge Engineering, vol. 3, Dec. 2022.
- 7. K.-H. Le, K.-H. Le-Minh, and H.-T. Thai, "Brainyedge: An ai-enabled framework for iot edge computing," *ICT Express*, vol. 9, no. 2, pp. 211–221, 2023.
- 8. G. Jocher, "YOLOv5 by Ultralytics," May 2020.
- 9. C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "Yolov6: A single-stage object detection framework for industrial applications," 2022.
- 10. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.
- 11. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023.
- 12. H.-T. Thai, N.-Y. Tran-Van, and K.-H. Le, "Artificial cognition for early leaf disease detection using vision transformers," in 2021 International Conference on Advanced Technologies for Communications (ATC), IEEE, Oct. 2021.
- 13. H.-T. Thai, K.-H. Le, and N. L.-T. Nguyen, "Formerleaf: An efficient vision transformer for cassava leaf disease detection," *Computers and Electronics in Agriculture*, vol. 204, p. 107518, 2023.
- X. Zhang, L. Han, Y. Dong, Y. Shi, W. Huang, L. Han, P. González-Moreno, H. Ma, H. Ye, and T. Sobeih, "A deep learning-based approach for automated yellow rust disease detection from high-resolution hyperspectral uav images," *Remote Sensing*, vol. 11, p. 1554, June 2019.
- 15. P. Ko, S. A. Prieto, and B. García de Soto, "Abecis: an automated building exterior crack inspection system using uavs, open-source deep learning and photogrammetry," in *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*, ISARC2021, International Association for Automation and Robotics in Construction (IAARC), Nov. 2021.
- S. Katsigiannis, S. Seyedzadeh, A. Agapiou, and N. Ramzan, "Deep learning for crack detection on masonry façades using limited data and transfer learning," *Journal of Building Engineering*, vol. 76, p. 107105, Oct. 2023.

- T. Wang and V. J. Gan, "Automated joint 3d reconstruction and visual inspection for buildings using computer vision and transfer learning," *Automation in Construction*, vol. 149, p. 104810, May 2023.
- 18. X. Zhao, S. Li, H. Su, L. Zhou, and K. J. Loh, "Image-based comprehensive maintenance and inspection method for bridges using deep learning," in *Volume 2: Mechanics and Behavior of Active Materials; Structural Health Monitoring; Bioinspired Smart Materials and Systems; Energy Harvesting; Emerging Technologies*, SMASIS2018, American Society of Mechanical Engineers, Sept. 2018.
- Y. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, p. 731–747, Nov. 2017.
- 20. C. Zhang, C. Chang, and M. Jamshidi, "Concrete bridge surface damage detection using a single-stage detector," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, p. 389–409, Oct. 2019.
- A. Nguyen, V. Gharehbaghi, N. T. Le, L. Sterling, U. I. Chaudhry, and S. Crawford, "Asr crack identification in bridges using deep learning and texture analysis," Structures, vol. 50, p. 494–507, Apr. 2023.
- P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," Procedia Computer Science, vol. 199, p. 1066–1073, 2022.
- Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021.
- 24. U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, p. 464, Jan. 2022.
- K. Liu, H. Tang, S. He, Q. Yu, Y. Xiong, and N. Wang, "Performance validation of yolo variants for object detection," in *Proceedings of the 2021 International* Conference on Bioinformatics and Intelligent Computing, BIC 2021, ACM, Jan. 2021.
- 26. E. U. Rahman, Y. Zhang, S. Ahmad, H. I. Ahmad, and S. Jobaer, "Autonomous vision-based primary distribution systems porcelain insulators inspection using uavs," *Sensors*, vol. 21, p. 974, Feb. 2021.
- 27. C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, and X. Chu, "Yolov6 v3.0: A full-scale reloading," 2023.
- 28. M. Horvat, L. Jelečević, and G. Gledec, "Comparative analysis of yolov5 and yolov6 models performance for object classification on open infrastructure: Insights and recommendations," 09 2023.
- I. Gillani, M. Munawar, M. Talha, S. Azhar, Y. Mashkoor, M. uddin, and U. Zafar, "Yolov5, yolo-x, yolo-r, yolov7 performance comparison: A survey," pp. 17–28, 09 2022.
- 30. D. Shokri, C. Larouche, and S. Homayouni, "A comparative analysis of multi-label deep learning classifiers for real-time vehicle detection to support intelligent transportation systems," *Smart Cities*, vol. 6, p. 2982–3004, Oct. 2023.
- 31. W. Xinming, T. S. Hong, $et\ al.$, "Comparative study on leaf disease identification using yolo v4 and yolo v7 algorithm," AgBioForum, vol. 25, no. 1, 2023.
- 32. O. E. Olorunshola, M. E. Irhebhude, and A. E. Evwiekpaefe, "A comparative study of yolov5 and yolov7 object detection algorithms," *Journal of Computing and Social Informatics*, vol. 2, p. 1–12, Feb. 2023.
- H.-V. Nguyen, J.-H. Bae, Y.-E. Lee, H.-S. Lee, and K.-R. Kwon, "Comparison of pre-trained yolo models on steel surface defects detector based on transfer learning with gpu-based embedded devices," Sensors, vol. 22, p. 9926, Dec. 2022.

34. "Performance benchmark of yolo v5, v7 and v8." Available at https://www.stereolabs.com/blog/performance-of-yolo-v5-v7-and-v8/.