Multimedia Tools and Applications (2024) 83:86955–86982
https://doi.org/10.1007/s11042-024-19723-6

Pothole detection in adverse weather: leveraging synthetic images and attention-based object detection methods

Maros Jakubec[1] · Eva Lieskovska[1] · Boris Bucko[1] · Katarina Zabovska[1]

Received: 11 October 2023 / Revised: 12 June 2024 / Accepted: 19 June 2024 /
Published online: 5 July 2024

Abstract

Potholes are a pervasive road hazard with the potential to cause accidents and vehicle damage. Detecting potholes accurately is essential for timely repairs and ensuring road safety. However, existing detection methods often struggle to perform in adverse weather conditions, including rain, snow, and low visibility. This work aims to improve pothole detection across diverse weather and lighting scenarios, employing a two-phase strategy that integrates data augmentation with images generated by Generative Adversarial Networks (GANs) and the deployment of visual attention techniques. For this purpose, advanced models such as YOLOv8, RT-DETR, and our modified version of YOLOv8 were employed. In the first phase, multiple image-to-image translation models were trained and applied to a real-world dataset to generate synthetic images of potholes under different weather conditions, including rain, fog, overcast, dawn, and night. The detection accuracy results show improvements in all monitored metrics across most tested conditions following the incorporation of augmentation. The most significant improvement resulting from augmentation was observed in low-visibility conditions, captured during evening and night, with an increase of up to 11% and 19% in mean Average Precision (mAP@.5) across all models. The second phase employed different modifications of YOLOv8 with modules such as Attention-Based Dense Atrous Spatial Pyramid Pooling, Vision Transformer and Global Attention Mechanism to enhance the detection of potholes in challenging visual conditions. The compensation for increased model complexity, such as the utilization of depthwise convolutions, was also employed. To evaluate the effectiveness of this approach, a publicly available pothole dataset with images captured in diverse weather conditions is used. The results indicate that the proposed method achieved an 8.4% improvement pre-augmentation and a 5.3% improvement post-augmentation compared to the original YOLOv8, surpassing existing approaches in terms of accuracy and enhancing pothole detection in adverse weather conditions.

Keywords Pothole detection · GAN · YOLO · Object detection · Weather conditions · Road safety

✉ Maros Jakubec
maros.jakubec@uniza.sk

[1] University Science Park UNIZA, University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia

1 Introduction

Road infrastructure plays a significant role in the transportation sector, and its maintenance is essential for safe and efficient transportation. Potholes are one of the most common issues that road authorities face, as they can lead to accidents and cause significant damage to vehicles. Potholes typically result from the natural wear and tear of roads. However, weather conditions such as rain, snow, and freeze–thaw cycles can exacerbate the problem by causing cracks to form and expand. Ensuring the timely detection and repair of potholes is crucial to preventing such issues [1–4].

Current methods for pothole detection typically rely on a visual inspection conducted by either human operators or systems employing cameras or LiDAR technology [5–8]. However, these approaches can be costly, time-consuming, and frequently produce inaccurate results. Nevertheless, due to advancements in computer vision and machine learning techniques, there has been a significant shift towards automating this process. The development of automatic pothole detection methods has the potential to greatly enhance road safety and reduce the costs associated with manual inspections. One promising approach is to use deep learning algorithms, such as You Only Look Once (YOLO) [9], to detect potholes and road bumps in images captured by cameras mounted on vehicles. However, training these algorithms requires a large amount of labelled data, which can be challenging to obtain, especially in various weather conditions.

While automatic pothole detection often achieves high accuracy, current methods do not always reflect the real problem. Figure 1. shows a close-up image taken while standing over a pothole (left). This is typically how data is acquired in most pothole datasets. However, this approach does not accurately reflect the reality of potholes on roads. The image on the right shows a more realistic scenario of pothole instances captured from a moving vehicle, representing how pothole detection tasks should be perceived. When the methods are presented in a manner that accurately reflects the problem, the detection performance suffers. This is because noise in images or videos, often exacerbated by low resolution, causes small potholes to appear as insignificant objects that blend into the background.

One of the main challenges in pothole detection is the variability in weather conditions. Rain, snow, fog, and low light can affect road surface visibility, making pothole detection more difficult. For instance, rain or snow can obscure potholes or create shadows, affecting detection accuracy. Collecting annotated data for potholes in various weather conditions can be time-consuming and impractical due to the high costs and safety concerns involved. To address this issue, we propose using generative AI

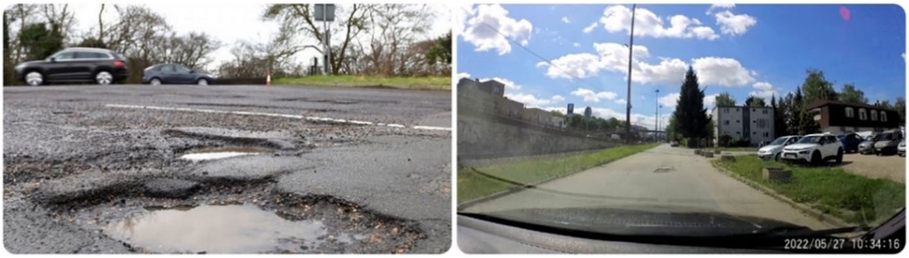


Fig. 1 Instances of potholes: The image on the right presents a more realistic example of a pothole detection task compared to the one on the left

to create synthetic images in different versions of weather and lighting conditions. By using Generative Adversarial Networks (GANs) to generate synthetic data, we can create a more diverse and extensive dataset, which can improve the performance of pothole detection algorithms in adverse weather conditions.

In this study, we developed an automated pothole detection system using advanced computer vision algorithms such as YOLOv8, Real-Time Detection Transformer (RT-DETR) [10] and our modification of YOLOv8. YOLOv8 continues to build on the achievements of its predecessors, introducing innovative features and enhancements. RT-DETR represents an advanced end-to-end object detection system, delivering real-time performance without compromising precision. In addition to these state-of-the-art (SOTA) architectures, our study has modified YOLOv8 by integrating self-attention through vision transformer blocks with hierarchical feature maps and a Global Attention Mechanism (GAM) into the neck scheme. The main objective is to assess the performance of selected architectures in handling unfavourable weather conditions.

The work contribution can be highlighted as follows:

We proposed using GANs to translate pothole images into different weather conditions to generate new synthetic data. The number of training and validation images was increased by a factor of 7. Augmented data plays a pivotal role in enhancing the model’s robustness against various challenges, including alterations in lighting and weather conditions.

We introduced a vision transformer along with GAM modules into the YOLOv8 neck to enhance the effectiveness of capturing valuable latent features and the global context prior to reaching the detection head. To improve the receptive field and capture more valuable image context, the Spatial Pyramid Pooling-Fast (SPPF) has been replaced by Attention-Based Dense Atrous Spatial Pyramid Pooling (ADASPP), which employs multi-scale features with a parameter-free attention module. Because potholes are relatively small compared to the overall scene, we incorporated the additional detection head for very small objects. The use of depthwise convolutions and the reduction in the number of convolution filters in detection layers helped to compensate for increased model complexity.

We evaluated the effectiveness of the proposed system in detecting potholes captured under real-world adverse weather conditions, including rain, evening, and nighttime. The performance of the proposed system was compared with existing methods for pothole detection.

The rest of this paper is organized as follows: Sect. 2 provides an overview of the existing research on pothole detection and image-to-image translation, highlighting the key approaches and techniques used in these areas. Section 3 describes the GAN architecture used for image-to-image translation and the proposed modification of YOLOv8 algorithm used for pothole detection. Section 4 outlines the dataset employed in this study, defines evaluation metrics, and discusses the process of model selection and training to optimize network architecture. Section 5 presents the quantitative results obtained from the experiments, along with a detailed analysis and discussion of the findings. Section 6 summarises the key contributions, draws conclusions based on the results and discussion, and suggests avenues for future research.

2 Related works

2.1 Pothole detection

With the rapid advancement of computer vision technology, there has been a concerted effort among researchers to explore diverse methods for detecting potholes in roads, aiming to facilitate timely repair and maintenance. Notably, deep learning-based object detectors have gained popularity as an effective approach for this task [11]. These detectors can be broadly classified into two subcategories: Region-based Convolutional Neural Networks (R-CNN) models, which employ a two-stage detection process, and single-stage models, which utilize uniform detection methods. Two-stage detectors, such as R-CNN [12], Fast R-CNN [13], and Faster R-CNN [14], typically employ a region proposal network or selective search in the initial stage to identify regions of interest (ROIs). In the subsequent stage, the selected ROIs are mapped for specific objects, and minimal bounding boxes are predicted for the detected objects. On the other hand, single-stage detectors like SSD [15], YOLO [9], and RetinaNet [16] eliminate the need for a region proposal network and directly perform detection on a dense sampling of all possible locations. Providing a balance between speed and accuracy, one-stage object detectors are a widely utilized tool for real-time applications.

Pena-Caballero et al. [17] evaluated object identification using YOLOv2 and YOLOv3, as well as semantic segmentation algorithms. They found that while segmentation achieved high accuracy, it also came with increased computational complexity. Ye et al. [2] proposed a CNN-based pothole detection method with pre-pooling before the first convolutional layer, achieving higher accuracy than conventional CNNs. Park et al. [18] proposed an automated pothole detection method using YOLOv4, YOLOv4-tiny, and YOLOv5 models. Their evaluation showed that YOLOv4-tiny performed the best, achieving a mean Average Precision at IoU 0.5 (mAP@0.5) of 0.787. However, limitations were noted, including reduced accuracy in detecting small, distant potholes. Salcedo et al. [19] introduced a series of deep learning models for developing a road maintenance prioritization system in India. Their system includes UNet for road segmentation to help determine fake or duplicate records. Object detection was conducted using EfficientDet and YOLOv5 models, resulting in mAP scores of 0.60 and 0.63, respectively, across three categories: single crack, crocodile crack, and pothole. The YOLOX algorithm was utilized in [20]. Experimental results indicate that the YOLOX-Nano achieved high accuracy in pothole detection while maintaining low computational costs and a compact model size of only 7.22 MB. Deepa and Sivasangari [21] introduced a hybrid deep learning framework that incorporates various stages, including histogram equalization-based image pre-processing, fuzzy c-means clustering-based segmentation, feature extraction, and classification using the Hybrid Deep Capsule autoencoder. The proposed model achieved an accuracy of 98.81% on the RDD2020 dataset, surpassing existing methods.

Despite promising results, the task of real-world pothole detection still presents challenges, primarily due to the small size of potholes relative to road images. This size discrepancy imposes limitations on training Convolutional Neural Networks (CNNs) with high-resolution images, primarily because of memory constraints. To overcome these challenges, Chen et al. [1] proposed resizing input images to fit the network and utilizing image patches from high-resolution images during network training. This approach involves a two-stage system: initially employing a localization network to locate the pothole instances in the low-resolution image, and then using a classification network based on candidate patches to determine the classes. Salaudeen and Celebi [3] used an Enhanced Super-Resolution GAN to improve the quality of road surface images and address the challenges associated with detecting small objects. Several studies have proposed deep CNN-based object detectors for detecting small objects, including potholes, in remote sensing imagery. For instance, Tayara et al. [22] introduced a convolutional regression neural network to detect vehicles from satellite imagery. Tang et al. [23] also proposed a modified Faster R-CNN detector that utilized a hyperregion proposal network to improve recall and employed a cascade-boosted classifier to reduce false detections. A YOLOv4-tiny model was used by Silva et al. [24] to detect potholes from aerial views captured by a flying drone with 95% accuracy. Despite the existing research, the literature lacks a thorough evaluation of how visual attention influences pothole detection.

Additionally, exploring novel state-of-the-art models could enhance pothole detection performance further. Xie et al. [25] proposed MADet, a one-stage detector that utilizes a feature-interaction alignment operation to enhance consistency between feature-prediction pairs through mutual-assistance learning. They also utilize joint optimization for predicting target bounding boxes, incorporating both anchor-based and anchor-free approaches. This enhances the detection of objects with diverse aspect ratios and addresses issue with object occlusion. In [26] Separate Feature Refinement (SFRNet) is proposed, featuring transformer-based branches for specific functions such as fine-grained classification and oriented localization. The end-to-end transformer RT-DETR, as proposed in [10], utilizes an efficient hybrid encoder for multi-scale feature processing and IoU-aware query selection to enhance object initialization. Furthermore, it enables flexible adjustment of inference speed without requiring retraining. RT-DETR-L achieved an impressive performance of 53.0% Average Precision (AP) on COCO val2017, with a processing speed of 114 frames per second (FPS) on a T4 GPU. YOLOv8 is the latest iteration in the YOLO series of object detection algorithms from the Ultralytics group. It introduces significant improvements over its predecessors, including enhanced model architecture and optimization techniques, which contribute to its superior performance on various benchmarks. YOLOv8 also offers better adaptability to different scales of objects, making it highly versatile for the detection of small objects. YOLOv8L achieved AP of 52.9% with an image size of 640 pixels on the MS COCO dataset test-dev 2017. Furthermore, YOLOv8L exhibits a speed of 418 FPS on an NVIDIA A100 TensorRT, highlighting its efficiency and computational prowess for object detection tasks.

In pothole detection research, there is a lack of testing conducted under adverse weather and lighting conditions. Some studies use thermal images to detect potholes in challenging weather conditions, such as fog and at night [8, 27]. Although, thermal images can provide valuable additional image features, further testing should be conducted using images captured in a moving vehicle scenario. While our previous work [28] primarily provided an overview and comparison of the performance of one- and two-stage detection architectures for pothole detection under adverse conditions, it did not incorporate targeted improvements into these architectures. This gap has motivated us to explore the potential of generative networks. Our aim is to increase the diversity of available data and enhance the robustness of selected detection models: RT-DETR and YOLOv8. Additionally, we aim to incorporate effective multiscale processing and visual attention modules into the YOLOv8 model to evaluate their suitability for pothole detection in challenging visual conditions.

2.2 Generative adversarial networks

In 2014, Goodfellow et al. introduced the concept of GANs, which has since become a significant development in the field of unsupervised deep generative models [29]. The architecture of a typical GAN consists of two competing neural networks inspired by a two-player minimax game: a generator network and a discriminator network. The objective of the generator is to produce realistic samples that can fool the discriminator, while the discriminator aims to distinguish between real and fake samples (as depicted in Fig. 2).

The field of image-generation techniques has witnessed notable breakthroughs in other GAN architectures. In 2015, Radford et al. [30] proposed a Deep Convolutional Generative Adversarial Network (DCGAN), which utilizes transposed convolutional layers to upsample the input noise vector, resulting in the generation of high-resolution images. Progressive Growing of GANs (PGGAN), introduced by Karras et al. in 2018 [31], adopts a progressive training strategy, beginning with low-resolution image generation and progressively increasing the resolution. This approach yielded remarkable results in generating high-resolution images, including 1024×1024 face images. StyleGAN, proposed by Karras et al. in 2020 [32], presents a variation of GANs that introduces a style-based generator architecture. Unlike traditional GANs, StyleGAN employs two inputs: a learned constant vector and a style vector that controls the image's style. It also introduces a synthesis network for generating images at different resolutions. StyleGAN achieved impressive outcomes in generating high-quality images. In 2023, GigaGAN was introduced [33] as a one-billion-parameter model that excels in text-to-image generation in data-rich scenarios while offering rapid inference times. It only takes 0.13 s to generate a 512px image and 3.66 s to generate a 4 K image.

Notably, GANs have demonstrated promising results in challenging generative tasks such as text-to-photo translation, image generation, image composition, and image-to-image translation [29, 34, 35]. Image-to-image translation (I2I) has emerged as a fundamental problem in computer vision and computer graphics, encompassing a wide range of applications. The goal is to learn the mapping from an input image (X) to a specific target image (Y), such as mapping grayscale images to RGB images. The concept of image translation traces its roots back to Hertzmann et al.'s image analogies [36]. This approach proposes a non-parametric model that utilises pairs of images to achieve image transformations. The seminal work by Mirza and Osindero [37] introduced conditional GANs, exemplified by Pix2Pix, enabling the learning of mappings between a source and target domain. This approach achieved impressive results in tasks such as scene translation, season

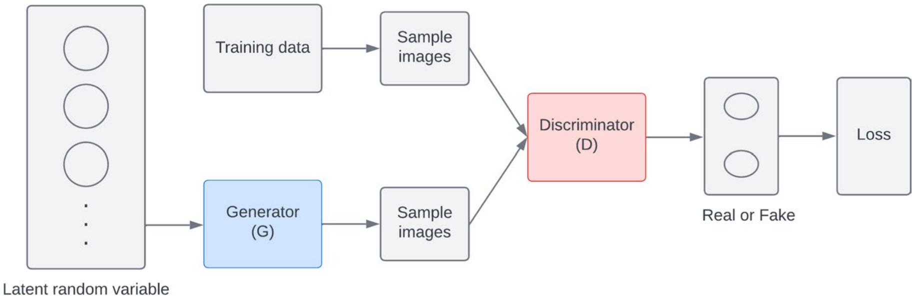


Fig. 2 The architecture of Generative Adversarial Network

transfer, and sketch-to-photo translation. Nevertheless, relying on paired images for training poses challenges and high costs in various scenarios.

To address the limitations associated with paired training data, Zhu et al. proposed CycleGAN, a framework that introduces a cycle consistency loss to enforce the translation of an image from the source domain to the target domain and back to the source domain. CycleGAN demonstrated consistent image generation and garnered substantial attention within the scientific community. Building upon CycleGAN, the Unsupervised Image-to-Image Translation Network (UNIT) replaces domain-specific latent spaces with a shared latent space across domains, further advancing the field of I2I translation.

While conditional GANs, CycleGAN, and UNIT have showcased impressive results, they predominantly focus on single-modal translations, neglecting the multi-modality inherent in I2I translation. BicycleGAN attempts to address this limitation by leveraging paired images during training, encouraging bijective consistency between latent and target spaces. BicycleGAN is known for generating images of high quality, featuring detailed and varied content. However, its main drawback is the significant computational resources it requires. More recent advancements include Multimodal Unsupervised Image-to-Image Translation (MUNIT) [38] and Diverse Image-to-Image Translation via Disentangled Representations (DRIT) [39]. These methods have introduced solutions for multi-modal, unpaired scenarios by learning disentangled representations with a domain-invariant content space and domain-specific attribute/style space. They have enabled the generation of diverse and high-quality images, even in scenarios where paired training data is unavailable. MUNIT is particularly suited for tasks where multiple output variations are desirable, offering a broader range of possible translations. As stated in the original paper [38], the MUNIT model achieves quality and diversity comparable to that of the fully supervised BicycleGAN, while also surpassing unsupervised models such as UNIT and CycleGAN. MUNIT's unsupervised nature offers an advantage in scenarios where labelled data is scarce or expensive to obtain.

3 Materials and methods

In this section, we will explain the methods used to create synthetic data using GANs and the process of training the GAN model. Next, we will discuss the implementation of the YOLO and RT-DETR algorithms for testing the augmented dataset.

3.1 Image-to-image translation

Achieving successful I2I translation requires an understanding of the underlying features shared between the source and target representations. In the realm of I2I translation, the ability to distinguish between domain-independent (content) and domain-specific (style) features is of paramount importance. Domain-independent features capture the spatial structure of the underlying content and should be preserved during the translation process. Learning the mapping between two or multiple domains poses significant challenges. Firstly, acquiring a paired dataset for training may be difficult or impractical, making supervised learning approaches infeasible. Secondly, performing multi-modal translation, where a single input image maps to multiple output images, adds further complexity to the problem.

Here, MUNIT emerges as a pioneering approach. MUNIT [38] decomposes image representation into a content space that is common across different domains and a style space that captures domain-specific characteristics. Content information is combined with a random style code from the target space to translate an image to the target domain. By blending content information with a randomly sampled style code from the target domain, MUNIT enables versatile image translation, yielding an array of diverse outcomes.

MUNIT, along with similar frameworks, shares a common architectural foundation: a generator comprising a style encoder and a content encoder (Fig. 3). The content encoder employs strided convolutional layers enhanced by residual blocks and instance normalization to downsample the input and capture content features. In contrast, the style encoder utilizes strided convolutional layers, global average pooling, and a fully connected layer to generate style codes, with the omission of instance normalization to preserve style information. The decoder reconstructs the input image using content and style codes using residual blocks, upsampling, and convolutional layers. Discriminators are employed, and the LSGAN objective, in combination with multi-scale discriminators, enhances the realism of the generated images.

A notable addition to the framework is the inclusion of the Domain-Invariant Perceptual Loss, which is a modified perceptual loss that exhibits greater domain-invariance and employs input images as references. By applying instance normalization to VGG features, domain-specific information is effectively removed, thereby improving the training process, particularly when working with high-resolution datasets.

The optimization process of MUNIT is underpinned by a comprehensive loss function, which includes the following components:

Bidirectional Reconstruction Loss: This loss enforces encoder-decoder pairs to act as inverses of each other, promoting reconstruction in both image → latent → image and latent → image → latent directions.

- Image Reconstruction: Successful encoding and decoding should result in the accurate reconstruction of an image from the data distribution.
- Latent Reconstruction: With a latent code comprising both style and content sampled during translation, successful decoding/encoding should allow for the faithful reconstruction of this latent code.

Adversarial Loss: MUNIT leverages GAN to align the distribution of translated images with the target domain's image distribution. Essentially, images generated by MUNIT should be indistinguishable from real images in the target domain.

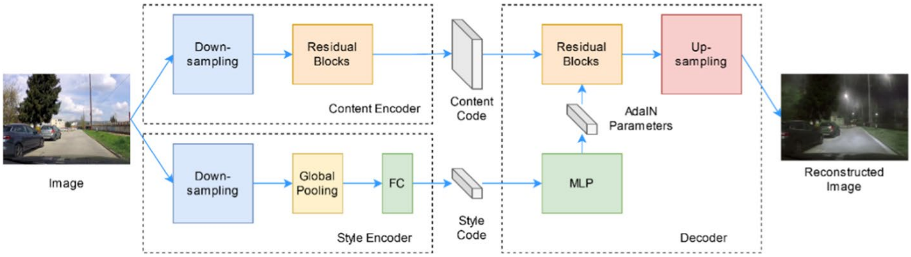


Fig. 3 MUNIT architecture overview: Content and style encoders transform input data, while a decoder generates images through AdaIN processing and convolutional layers

Recent advancements have propelled MUNIT to the forefront as a crucial framework for addressing the challenge of multimodal unsupervised I2I translation. However, it is important to note that this approach necessitates the use of multiple encoders and decoders for each domain, imposing significant computational demands. Furthermore, there is room for improvement in the conventional method of sampling style codes from a standard distribution during the translation process.

3.2 Pothole detection

Object detection is a crucial task in computer vision, involving identifying and localising objects within an image. YOLO family of object detection algorithms has gained significant attention due to its real-time performance and accuracy. This work focuses on two versions of the YOLO algorithm: the original YOLOv8 and our modification of YOLOv8, designed specifically for processing small objects in adverse visual conditions. Additionally, the performance of the detection transformer RT-DETR was evaluated.

3.2.1 YOLOv8

YOLOv8 [40], developed by Ultralytics, represents the most recent iteration of the YOLO series (refer to Fig. 4). The YOLOv8 backbone is composed of a chain of Conv (convolution, batch norm, SiLU activation) and a faster CSP bottleneck with two convolution blocks (C2f). It also includes a Spatial Pyramid Pooling Fast (SPPF) module to enhance the computational efficiency of the network. SPPF employs a series of max-pooling operations with kernels of identical sizes, as opposed to parallel max-pooling operations with varying kernel sizes implemented in SPP.

The neck structure of the Feature Pyramid Network-Path Aggregation Network (PAN-FPN) is similar to that in YOLOv5 [41]. The difference is that YOLOv8 uses C2f blocks instead of C3 blocks, and the number of Conv blocks has also been reduced.

The YOLOv8's decoupled head now processes objectness, classification, and regression tasks independently. An anchor-free approach for generating object proposals is utilized, meaning that predefined anchor boxes are not used. When it comes to real-time detectors that require Non-Maximum Suppression (NMS) post-processing, anchor-free detectors are more efficient in terms of inference time than anchor-based detectors with the same level of accuracy. This is because anchor-free detectors require significantly less post-processing time compared to their anchor-based counterparts [10]. The YOLOv8 utilizes Complete IoU (CioU) [42] and Distribution Focal Loss (DFL) [43] loss functions for bounding box loss and Binary cross-entropy (BCE) for classification loss.

3.2.2 RT-DETR

Baidu's Real-Time Detection Transformer [10] (RT-DETR) is a SOTA object detector that offers high accuracy and real-time performance (refer to Fig. 5). Unlike traditional real-time detectors, RT-DETR eliminates the need for NMS post-processing, making detection simpler and faster during inference.

In RT-DETR, the efficient hybrid encoder was redesigned with real-time performance in mind. It uses the final three stages of the HGNetv2 backbone as input for its encoding process. The multiscale features are processed using Attention-based Intra-scale Feature Interaction (AIFI) and CNN-based Cross-scale Feature-fusion Module (CCFM). The

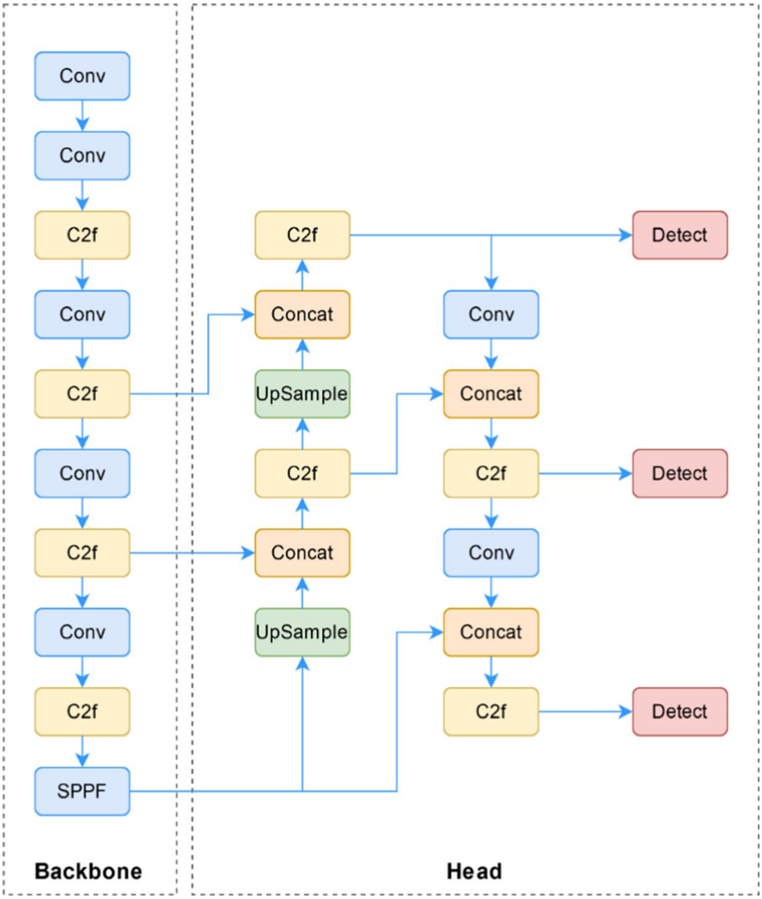


Fig. 4 The network structure of the YOLOv8

IoU-aware Query Selection process involves choosing a set amount of image features from the encoder output. The model selects the top K encoder features based on their classification score, and the prediction boxes corresponding to these features have high classification and IoU scores. By utilizing IoU-aware query selection during training, the model can generate encoder features of better quality. The final process utilizes a transformer decoder alongside auxiliary prediction heads to continually enhance object queries, resulting in the creation of boxes and confidence scores with increased accuracy.

3.2.3 Modification of YOLOv8

The modification proposed in this work for YOLOv8 (refer to Fig. 6) is aimed at improving the detection of small objects under adverse visual conditions. The architecture we started with, called YOLOv8-P2, performs the detection of extra-small, small, medium, and large objects. We decided to apply this four-detection head model due to the presence of primarily small-sized potholes in the dataset. At the end of the backbone, multi-level feature extraction was performed using the Attention-based Dense Atrous Spatial Pyramid Pooling

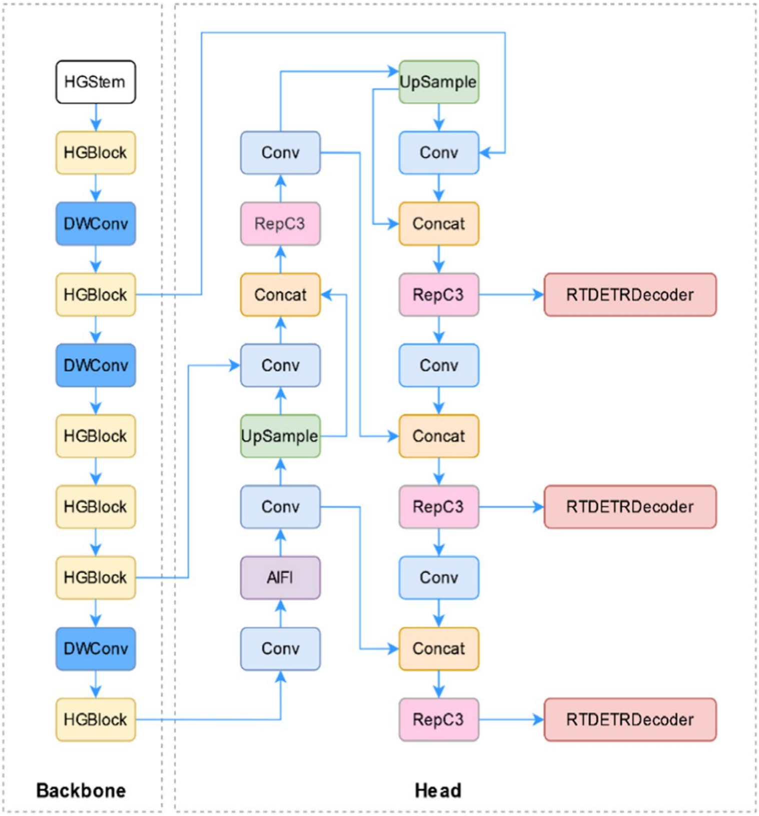


Fig. 5 The network structure of the RT-DETR

(ADASPP) block to enhance the convolutional receptive field in the feature pyramid computation. The CST/GAM blocks were used for the enhanced extraction of both local and global context from existing feature maps. Depthwise Convolutions (DWConv) perform a separate convolutional operation (using a depthwise kernel) for each input channel. This approach reduces the number of parameters without significantly compromising accuracy.

3.2.4 ADASPP

As mentioned in previous subsections, the original YOLOv8 employs the SPPF module for multi-level feature extraction. In this study, we have replaced the original SPPF module with a solution aimed at improving the detection of small objects and extracting more salient features under low visibility conditions. We proposed an attention-based module called ADASPP to enhance the convolutional receptive field. This module is based on Atrous Spatial Pyramid Pooling (ASPP) that utilizes multiple parallel filters with varying dilation rates. Figure 7(a) shows the implementation of the ASPP module from the YOLO Air repository [44]. In addition to adaptive average pooling, ASPP incorporates atrous convolutions with dilation rates of 1, 6, 12, and 18.

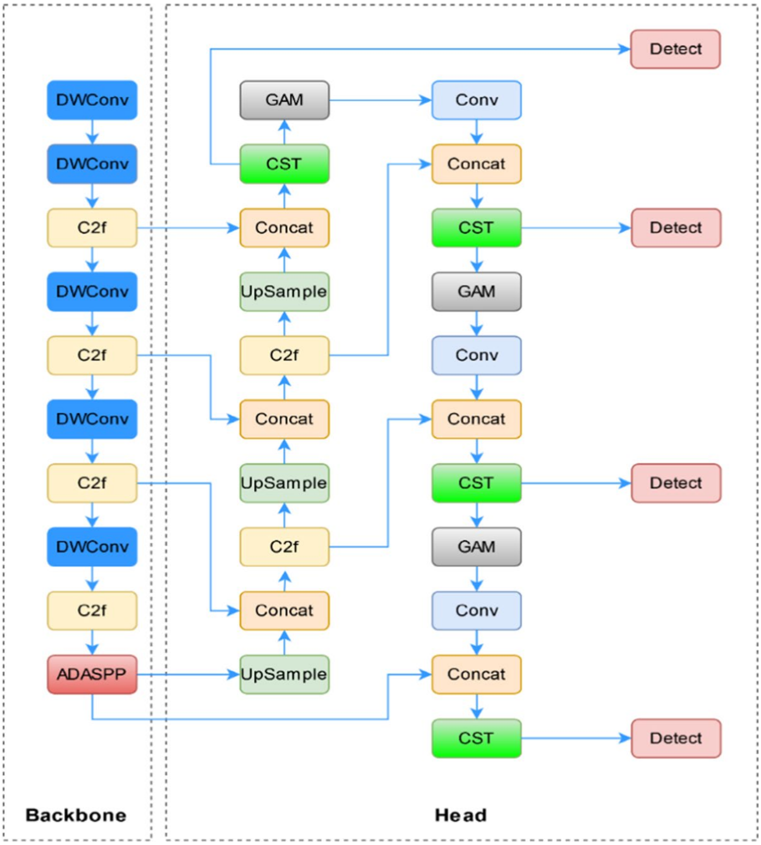


Fig. 6 The network structure of the YOLOv8 modification

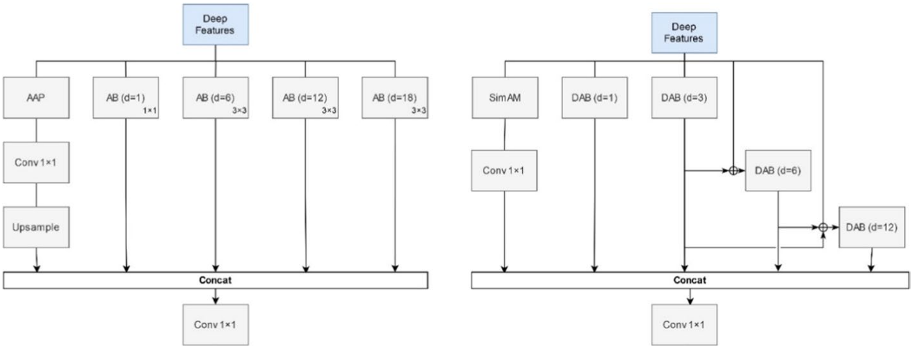


Fig. 7 (a) The ASPP module designed for multi-scale feature extraction (b) the modified ADASPP module with dense connections and attention for improved multi-level feature extraction

As described in [45], densely connected atrous convolutions in DenseASPP involve a larger receptive field in the computation of the feature pyramid. This fact led us to modify the ASPP so that it now contains skip connections with both previous dilated blocks and input feature maps. In the dense atrous block (shown as DAB in Fig. 7(b)) with dilation of 3, 6 and 12, a 1 × 1 convolution is applied before the dilated layer to reduce the size of the feature map to half of its original size. In the ADASPP, the adaptive average pooling has been replaced by a Simple Parameter-Free Attention Module (SimAM) to capture salient image contexts without significantly increasing the model's complexity. SimAM dynamically emphasizes relevant features and computes 3-D attention weights for the feature map without adding extra parameters to the network.

3.2.5 Swin transformer

In contemporary image analysis, extracting maximal information from images is crucial. To address this necessity, we incorporated the Convolutional Swin Transformer (CST) blocks (shows in Fig. 8(a)) capable of extracting latent features within image data. The latent features are hidden within the image and have the potential to offer valuable insights for recognition purposes. A distinctive feature of CST is the integration of Swin Transformer units, as exemplified in Fig. 8(b), where essential components such as Layer Normalization (LN) and Multi-Layer Perception (MLP) modules seamlessly converge. It is worth noting that the initial unit employs the window-based self-attention (WSA) mechanism, while its counterpart adopts the shifted window self-attention (SWSA) paradigm. In other words, Swin Transformer employs attention calculations within a local window rather than across the entire image to optimize time complexity. In the subsequent layer, the window shifts and crosses previous windows to expand the receptive field. This SWSA paradigm results in a network configuration that effectively emphasizes the intrinsic significance present in image data.

The procedural operation of this algorithm is outlined in Eqs. (1–4), where x represents the feature mapping of each layer after processing. Following the works discussing the benefit of LN for CNN [46, 47], we decided to remove LN from the Swin Transformer layers in our implementation.

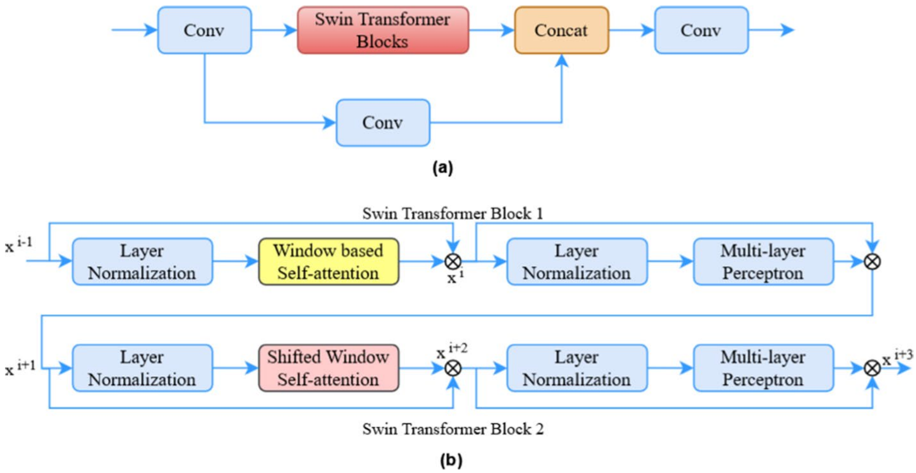


Fig. 8 (a) Introduces CST blocks, crucial for extracting hidden features in image data (b), the integration of Swin Transformer units, along with Layer Normalization and MLP modules

$$x^{i} = WSA\left(LN\left(x^{i-1}\right)\right) + x^{i-1} \tag{1}$$

$$x^{i+1} = MLP\left(LN\left(x^{i}\right)\right) + x^{i} \tag{2}$$

$$x^{i+2} = SWSA\left(LN\left(x^{i+1}\right)\right) + x^{i+1} \tag{3}$$

$$x^{i+3} = MLP\left(LN\left(x^{i+2}\right)\right) + x^{i+2} \tag{4}$$

3.2.6 Global attention mechanism

GAM is an advanced attention mechanism used in deep learning models to enhance global feature interactions and reduce message diffusion. It incorporates a sequential channel-spatial attention mechanism, modifying the processing of the Channel Attention Mechanism (CAM) and Spatial Attention Mechanism (SAM) from the Convolutional Block Attention Module (CBAM) submodule [48].

GAM is shown in Fig. 9. In this context, the input feature mapping, denoted as $F_1 \in \mathrm{R}^{C\times H\times W}$, represents the intermediate output from previous layers. M_C represents the channel attention map, M_S represents the spatial attention map, and $\otimes$ signifies element-wise multiplication. The CAM mechanism performs 3D permutation with a multilayer perceptron, operating on F_1 to produce an intermediate state F_2. This step captures enhanced feature interactions and relevant patterns while suppressing irrelevant information (Eq. (5)). Further SAM convolutional spatial attention processing results in F_3 as the output (Eq. (6)). These improved features can be used in subsequent layers or downstream tasks, significantly enhancing feature interactions and representation learning. The effectiveness of GAM led to improved performance across various deep-learning applications.

$$F_2 = M_C\left(F_1\right) \otimes F_1 \tag{5}$$

$$F_3 = M_S\left(F_2\right) \otimes F_2 \tag{6}$$

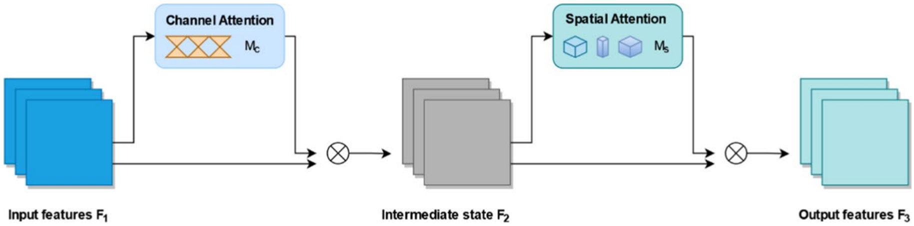


Fig. 9 Frame of Global Attention Mechanism

4 Experimental preparation

In this chapter, we delve into the details of the experimental preparation for our research on pothole detection using I2I translation. We discuss the datasets and pre-processing steps, evaluation metrics, the development of synthetic data, and the experimental setup.

4.1 Datasets and pre-processing

A Berkeley Diverse Driving Dataset [49] (BDD100k) was utilized for I2I conversion in this work. BDD100k contains a wide range of traffic object categories, including lane types, traffic lights and signs, and drivable areas. The dataset comprises images captured under diverse lighting and weather conditions, as detailed in Table 1. To perform image translation, we used the train subfolder (70,000 images) and validation subfolder (10,000 images) as our new training and test subsets. Table 1 provides a summary of the available training images, including their description. Using the labels available in JSON format, we were able to extract 69,863 unique instances belonging to three categories: weather, scene, and time of day. Due to the scarcity of data in the Foggy subset, additional images were included from datasets available on Roboflow [50, 51].

We evaluated the object detection task using a publicly available pothole dataset that was collected with a focus on adverse visual conditions [52]. The dataset is composed of over 1052 full HD images captured in clear weather and 1047 images influenced by adverse weather or low light. Image annotations contain two classes: pothole and manhole cover.

4.2 Evaluation metrics

Precision and recall were used to evaluate the performance of detection models. Precision is the ratio of true positives (TP) to all predicted objects (Eq. (7)), while recall is the ratio of TP to the total number of objects in the dataset (Eq. (8)). TP represents the correctly identified objects, while FP refers to objects wrongly detected as potholes. FN is defined as objects that the detector failed to identify as potholes.

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

Table 1 Description of training subfolder from BDD100k dataset sorted into the corresponding categories

	Weather	Scene	Time of day
count	69,863	69,863	69,863
unique	7	7	4
label	clear, rainy, undefined, snowy, overcast, partly cloudy, foggy	city street, highway, residential, parking lot, undefined, tunnel, gas stations	daytime, dawn/dusk, night, undefined
top	clear	city street	daytime
freq	37,344	43,516	36,728

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

Average precision (AP) is a commonly used evaluation metric in object detection that provides a comprehensive assessment of the detection model's performance across a range of Intersection over union (IoU) thresholds. IoU measures the overlap between the predicted and ground truth bounding boxes. AP is calculated by computing the area under the precision-recall curve at a specific IoU threshold. In practice, AP is obtained by averaging the precision values obtained at different IoU thresholds.

$$AP = \frac{1}{11}\sum\nolimits_{r\in\{0,0.1,\ldots,1\}} \rho_{interp(r)} \tag{9}$$

Mean average precision (mAP) is a variant of AP that provides an assessment of the detection model's performance across all classes. In this study, we used mAP@0.5 to evaluate the model's performance at a single IoU detection threshold of 0.5. Additionally, we used mAP@[0.5:0.95], which is averaged over several IoU thresholds from 0.5 to 0.95 with a step of 0.05.

$$mAP = \frac{1}{n}\sum\nolimits_{i=1}^{n} AP_i \tag{10}$$

4.3 Development of synthetic data

The unavailability of a sufficient number of paired images depicting varying visual conditions, from clear to adverse, led us to consider the scenario of unsupervised I2I translation. For the task of unsupervised translation, the numerous datasets of images captured under adverse visual conditions are already available. The BDD100k dataset was chosen for its extensive collection of traffic scenery images, which include diverse lighting and weather conditions such as clear, rainy, snowy, overcast, partly cloudy, and foggy. Additionally, the dataset offers a variety of time-of-day settings (daytime, dawn/dusk, night) and locations (city streets, highways, residential areas, etc.). In our work, we employed the MUNIT model for generating synthetic pothole images, primarily due to MUNIT's capability to learn a diverse set of high-quality image translations without requiring paired images. MUNIT contributes to the overall variation of training samples.

Our objective was to leverage the training images of potholes captured under clean weather and transfer them to low-light, foggy, or rainy conditions. The MUNIT models, trained on the diverse weather conditions present in the BDD100k dataset, were utilized individually to conduct I2I translation on the clear data from the pothole dataset. This strategy allows us to simulate pothole scenarios under various conditions, thus enhancing the dataset's diversity and realism.

The default configuration of MUNIT was used in our experiments. However, the training iterations were reduced to 300,000, the batch size was increased to 4, and the resize parameter for the shortest image side was increased to 640. The visual quality of the generated images was assessed through human judgment.

After a visual inspection of the generated images, it was found that the different nature of rainy and nighttime images from BDD100k prevented the model from being properly trained for our conditions. Therefore, new training images of night and rainy environments from our region were collected to match the required conditions. A more realistic

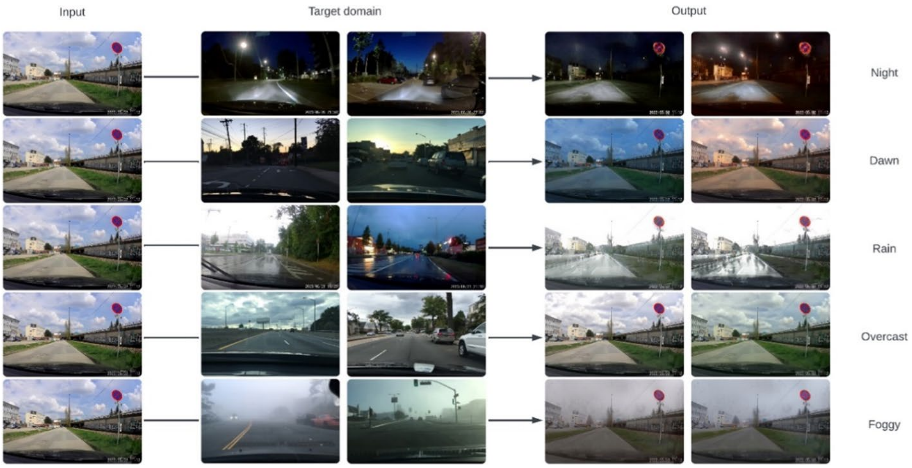


Fig. 10 The I2I translation process displaying unpaired images from the target domain alongside a single input that is transformed into multiple outputs

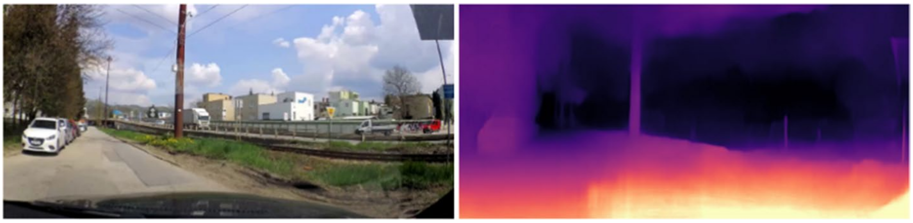


Fig. 11 The disparity image extracted by Monodepth2

appearance of the translated images was achieved. Thus, in the final experiments, images depicting dawn/dusk, foggy, and overcast conditions from the BDD100k dataset were utilized. Figure 10 depicts the process of translating a single input image into two distinct output images, selected from a total set of ten for each condition. Moreover, examples from the target domain are presented. The generated dataset underwent thorough evaluation to guarantee the realism and diversity of the synthetic images. Instances of newly generated low brightness and visually corrupted images were excluded from the dataset.

Moreover, for the generation of more realistic rainy images, the physically based rendering (PBR) approach implemented by [53] [54] was used. Here, disparity images are required for computations. The image depth information, or disparity, is computed from a pair of stereo images using the distance between the left and right pixel values. If a stereo image is not available, monocular depth estimation can be utilized via deep learning. In this work, disparity images from the monocular pothole images were generated by Monodepth2 [55], the result of which can be seen in Fig. 11. Rainy images generated by MUNIT were used before the application of PBR to obtain a more realistic appearance.

In our experiments with pothole detection, the pothole data was divided as follows: 70–15-15 partitions from clear weather were used for the train-validation-test subsets. The remaining images of real-world adverse conditions were used for the evaluation of models. As mentioned earlier, the generative method was employed to generate additional samples featuring diverse weather and lighting conditions based on clear weather data. Only one iteration of images, selected from a total set of ten variations, was chosen for each adverse condition. Up to a seven-fold increase in the size of training and validation subsets was achieved with the image translation into conditions such as dawn/dusk, foggy, night, overcast, rain, and images with 10 mm and 25 mm rain intensity, shown in Fig. 12.

4.4 Experimental setup

In this study, we conducted a series of experiments to test how well our proposed model works. To train deep learning models effectively, we used a set of well-known software tools, including Anaconda3, CUDA version 11.3 for faster model computations, cuDNN8 for optimized neural network calculations, and PyTorch 1.11.0 as our main training platform. The hardware consisted of a high-performance Intel Core i9 12900HX CPU and an NVIDIA GeForce RTX 3090 Ti 24 GB GPU.

For our assessment, we used models from the Ultralytics repository version 8.0.138 as our starting point [40]. These models were carefully trained on COCO dataset [56], which is widely used as benchmark data in computer vision tasks. During training, we limited the number of training epochs to a maximum of 500, with the first three epochs dedicated to the initial warm-up. To improve the learning process, we employed a technique called SGD optimization with an initial learning rate of 0.001.

The images from pothole dataset often contain many small objects. To achieve the balance between real-time performance and detection accuracy, we standardized the image size to 1088 px. This size ensures that our models can be used efficiently on devices with limited resources without losing important image details. Furthermore, we

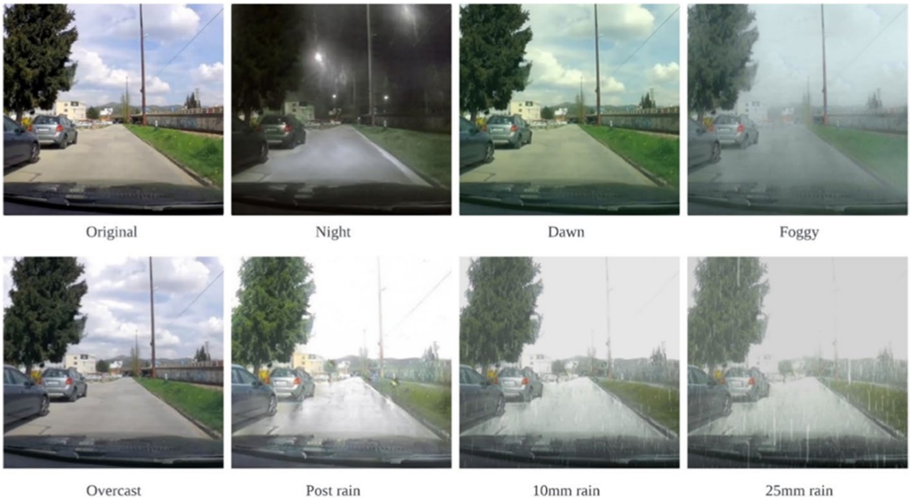


Fig. 12 Demonstration of the adaptability of a trained MUNIT network in diverse lighting and weather conditions

kept the settings for various parameters consistent across all training processes to ensure that our results could be compared accurately. The key parameter settings for the training process are summarized in Table 2.

5 Results and discussion

In this chapter, we present the results of our comprehensive analysis of pothole detection using SOTA architectures, such as YOLOv8, RT-DETR, and our modified YOLOv8. YOLOv8 and RT-DETR were specifically chosen due to their prominent performance in the field of object detection. One of the objectives of this research is to enhance the diversity of available dataset to bolster the robustness of detection models. We evaluate two training conditions: with and without data augmentation using GANs, to demonstrate the effectiveness of proposed approach across different scenarios. Our focus is also on improving the detection of small objects, particularly potholes, under adverse visual conditions (rain, sunset, evening, and night). The integration of efficient multiscale processing and visual attention mechanisms into the YOLOv8 model is therefore pursued with the intention of enhancing model accuracy. We discuss the impact of modifications made to YOLOv8, such as ADASPP and CST/GAM blocks for enhanced context extraction and DWConv for parameter reduction.

5.1 Performance comparison of pothole detection

Table 3 presents the results of pothole detection accuracy achieved by the different models and conditions The precision, recall, mAP@0.5), and mAP@[0.5:0.95] scores were evaluated. We investigated the impact of generative-based augmentation on model performance. Incorporating augmentation into the training process led to improvements in all evaluation metrics for each model. Notably, the evening and nighttime subsets showed the most significant improvement, with up to an 11% and 19% increase in mAP@0.5, respectively. The least significant change was observed in the rain and sunset subsets, with a 1–3% change in mAP@0.5 for the rain subset and a 0–0.7% change for the sunset subset. It’s worth noting that the expected improvement for the rain subset did not fully materialize despite augmenting the data for rainy conditions.

The best results among all models and conditions are indicated in bold. The proposed modification to YOLOv8 achieved the highest mAP@0.5 and mAP@[0.5:0.95] scores on

Table 2 Summary of parameter settings

Parameters	Setup
Epochs	500
Patience	30
Batch Size	8
Imgsize	1080
Initial Learning Rate	0.001
Final Learning Rate	0.01
Optimizer	SGD
Momentum	0.937
Weight-Decay	0.0005

Table 3 Pothole detection accuracy for different models and visual conditions

Model	Data subset	Augment	Precision	Recall	mAP@0.5	mAP@ [0.5: 0.95]
YOLOv8l	Clear		0.765	0.695	0.755	0.351
		✓	0.835	0.702	0.814	0.368
	Rain		0.607	0.476	0.412	0.153
		✓	0.638	0.527	0.426	0.172
	Sunset		0.485	0.387	0.355	0.131
		✓	0.487	0.386	0.355	0.132
	Evening		0.662	0.455	0.486	0.184
		✓	0.724	0.625	0.598	0.247
	Night		0.285	0.251	0.290	0.092
		✓	0.638	0.513	0.482	0.186
RT-DETR-l	Clear		0.838	0.730	0.831	0.372
		✓	**0.909**	0.785	0.856	0.384
	Rain		0.621	0.573	0.446	0.171
		✓	0.656	0.588	0.484	0.240
	Sunset		0.503	0.442	0.377	0.145
		✓	**0.514**	**0.461**	**0.384**	**0.148**
	Evening		0.689	0.487	0.512	0.198
		✓	0.738	0.667	0.623	0.262
	Night		0.304	0.299	0.313	0.107
		✓	0.656	0.581	0.505	0.194
Modified YOLOv8l	Clear		0.846	0.769	0.839	0.375
		✓	**0.909**	**0.796**	**0.867**	**0.391**
	Rain		0.637	0.578	0.454	0.185
		✓	**0.662**	**0.591**	**0.489**	**0.246**
	Sunset		0.501	0.441	0.376	0.144
		✓	0.512	0.458	0.382	0.147
	Evening		0.692	0.491	0.522	0.206
		✓	**0.739**	**0.674**	**0.632**	**0.266**
	Night		0.312	0.297	0.324	0.109
		✓	**0.663**	**0.585**	**0.511**	**0.203**

the clear data subset, with values of 0.867 and 0.391, respectively, outperforming both the original YOLOv8 and RT-DETR. The incorporation of the proposed modification in YOLOv8 yielded an 8.4% improvement pre-augmentation and a 5.3% improvement post-augmentation when compared to the original YOLOv8. In terms of detection accuracy under adverse conditions, the proposed model demonstrated a pre-augmentation improvement of 4.2%, 2.1%, 3.6%, and 3.4% (averaging 3.3%) in the rain, sunset, evening, and night subsets, respectively. Post-augmentation, the model achieved improvements of 6.3%, 2.7%, 3.4%, and 2.9% (averaging 3.8%) in the same subsets.

Figure 13 compares the performance of models under various adverse test conditions, including rain, sunset, evening, and night. It also provides a clearer illustration of how data augmentation impacts detection accuracy for each specific data subset. The impact of adverse visual conditions on detection accuracy can be indirectly assessed by examining

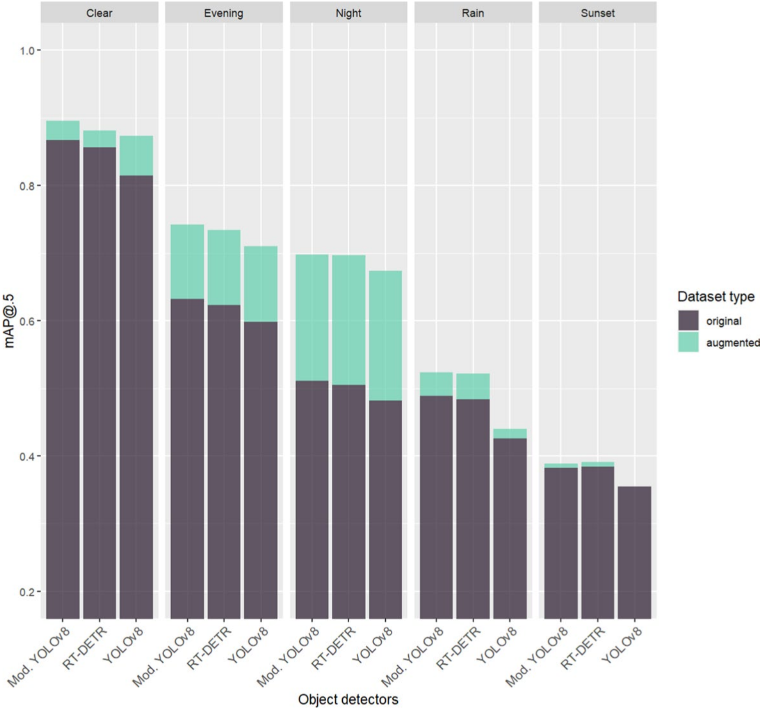
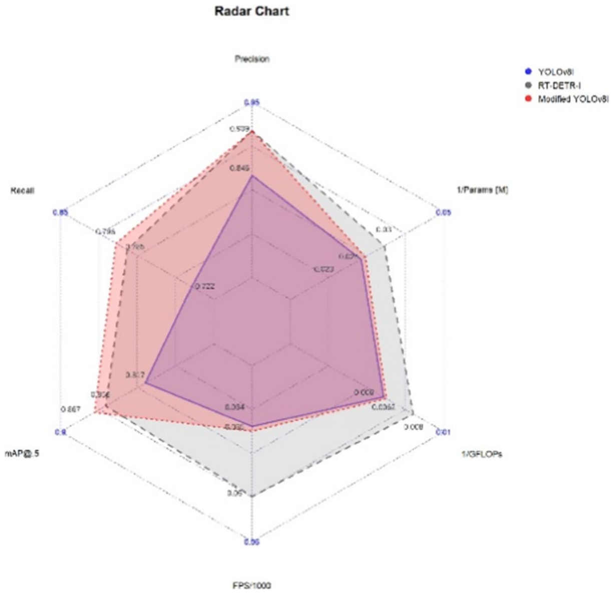


Fig. 13 Performance comparison of YOLOv8l, RT-DETR-l and modified YOLOv8l under adverse visual conditions and with/without data augmentation

Fig. 14 Comparison of parameters and performance among models



subsets where conditions other than clear weather are present, which consistently results in reduced mAP@0.5 scores for all models. The artificially generated data notably enhanced detection performance under conditions of low visibility, such as during the evening and night. It is also evident that the self-attention induced by transformers and global attention blocks is beneficial for detecting small objects under adverse conditions.

The radar chart (see Fig. 14) below visualizes the parameters of the tested architectures using augmented data. When it comes to accuracy, we can see that the modified YOLOv8 has reached the level of RT-DETR-l, outperforming the original YOLOv8. In terms of model size and computational requirements, the RT-DETR-l model stands out due to its low number of parameters (32.8 million), low floating-point operations per second (125.1 GFLOPs), and the shortest inference time, approximately 50 FPS. The parameters of the proposed modification to YOLOv8 (41.15 M parameters, 159.8 GFLOPs, ~ 35 FPS) closely follow the computational parameters of the original architecture (43.6 M parameters, 165.4 GFLOPs, ~ 34FPS).

Figure 15 compares the performance of models under various adverse test conditions, including rain, sunset, evening, and night. The same axis scales are kept to visually compare significant differences in individual subsets. It is obvious that the self-attention induced by the transformer and global attention blocks is beneficial for detecting small objects under adverse conditions. In our experiments, both modified YOLOv8 and RT-DETR are characterized by improving accuracy in poor visibility.

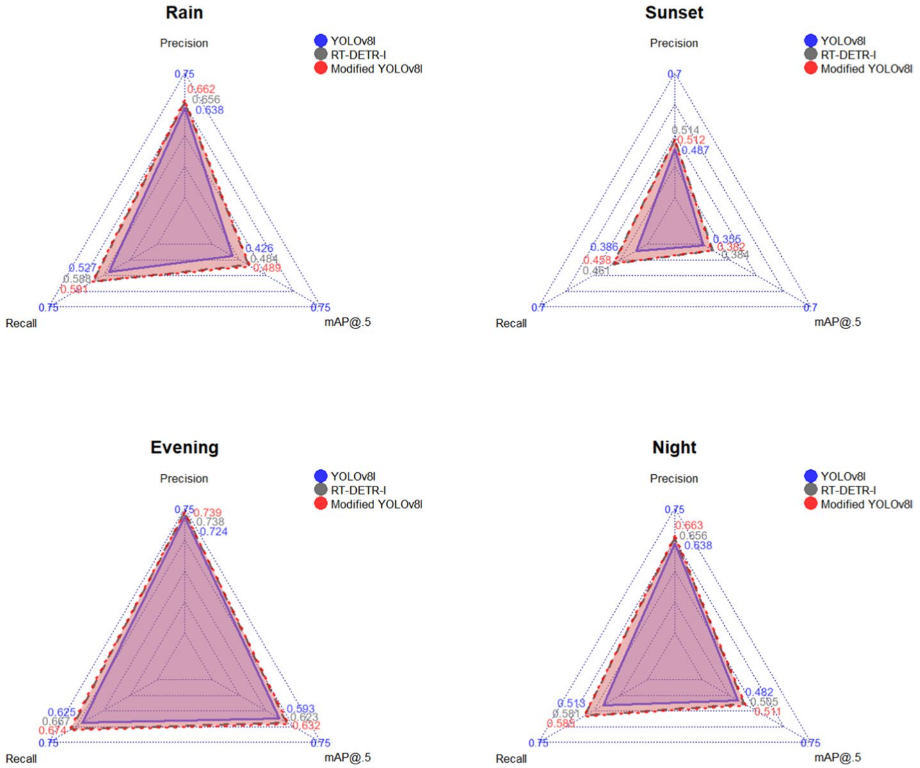


Fig. 15 Comparison of model performance under different adverse conditions

5.2 Ablation study

In our ablation study, we aimed to understand the contribution of each component to the performance of the modified YOLOv8 model. We started with the YOLOv8-P2 architecture, which already implements an additional detection head for small objects. The following components were then progressively added to the model: DWConv blocks, ADASPP, GAM, and CST blocks. Table 4 summarizes the results of the ablation experiments.

Summarizing the subcategory results and overall results in Table 4 the following conclusions can be drawn:

> Although the YOLOv8-P2 utilizes an additional detection head for very small objects, it has fewer parameters than YOLOv8 (43.6 million parameters). This reduction is achieved through a decrease in convolutional filters within the detection layers.
> The addition of DWConv blocks in the YOLOv8-P2 backbone reduced model parameters without a significant decrease in performance. This demonstrates the efficiency of DWConv in reducing model complexity while maintaining accuracy.
> The introduction of ADASPP further improved the model's accuracy, especially in terms of precision and recall. ADASPP enhances the model's ability to capture image context at different scales, which is crucial for small object detection.
> The CST and GAM blocks, designed to extract both local and global context from feature maps, contributed to significant improvements in precision, recall, and mAP@0.5. These blocks help the model better understand the spatial relationships within the image, aiding in accurate object localization.
> When all modifications were incorporated, the modified YOLOv8 model achieved a final accuracy that closely matched that of the RT-DETR transformer detector. This indicates that our approach effectively addresses the challenges posed by small object detection.

The detection outcomes are illustrated in Fig. 16. As shown, the proposed model enhances the detection capabilities of the original YOLOv8. The performance of the modified YOLOv8 is very similar to that of RT-DETR. However, the modified YOLOv8 shows better performance in detecting distant potholes during rainy and sunset conditions, identifying more potholes than RT-DETR. Furthermore, the most notable improvement is observed in the night example, where the modified YOLOv8 successfully detects potholes, surpassing RT-DETR in this specific instance.

Table 4 Results of ablation experiment on modified YOLOv8 model

Components	1	2	3	4	5	6
+ DWConv		✓	✓	✓	✓	✓
+ ADASPP			✓	✓	✓	✓
+ GAM				✓		✓
+ CST					✓	✓
P	78.9%	78.1%	80.4%	82.5%	83.3%	**84.6%**
R	63.1%	62.8%	68.4%	74.6%	72.2%	**76.9%**
mAP@0.5	76.7%	76.5%	78.1%	81.4%	82.7%	**83.9%**
Parameters [M]	42.8	**38.9**	39.5	40.3	40.4	41.2
Model Size [MB]	84.1	**65.5**	71.3	77.6	78.9	81.5

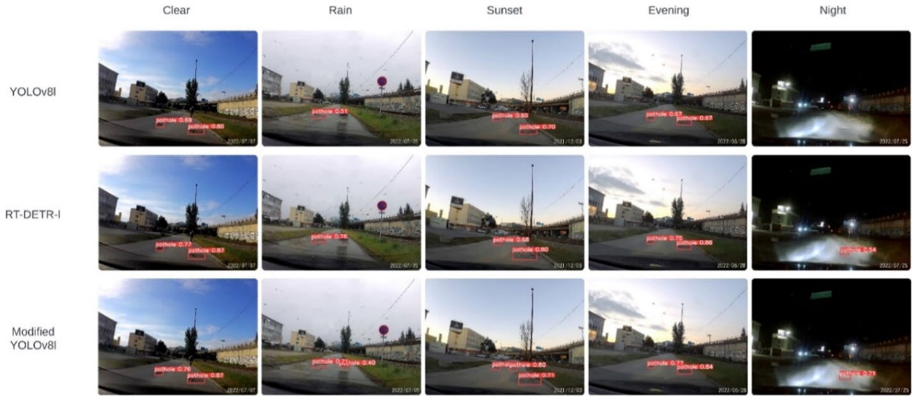


Fig. 16 Analysis of the detection performance of YOLOv8, RT-DETR, and modified YOLOv8 under varied environmental conditions

This study offers insights for developing automated pothole detection systems using GAN-generated synthetic data, which can be applied to enhance road safety and efficiency. Trained deep-learning algorithms can accurately identify potholes in various weather conditions, which could aid in road maintenance and reduce accidents and vehicle damage. The approach could also be expanded to detect other road defects like cracks and bumps. Despite the enhanced robustness of the model, potential biases inherent in the area-specific dataset could affect the model's performance in real-world scenarios. Moreover, the difference in road categories might affect the fidelity of the generated image. For example, the scenery of buildings closely encompassing a road in the training images can contribute to the generation of artifacts when depicting an open rural road. Therefore, for the chosen type of communication, appropriate data would have to be considered to create synthetic data. In future experiments, the proposed work could be enhanced by utilizing a dataset that includes a balanced representation of various road types and geographic locations.

6 Conclusion

Potholes present a persistent road hazard, often resulting in accidents and vehicle damage. However, detecting potholes in poor visibility conditions is a challenging task that requires innovative solutions. In this study, we assessed the accuracy and efficiency of detection models, such as YOLOv8, RT-DETR, and our modified version of YOLOv8, under degraded visibility conditions. We adopted two approaches to address the impact of poor visibility on detection accuracy. First, we enriched the dataset by incorporating artificially generated images utilizing MUNIT for I2I transfer, thereby bolstering the models' robustness. This sevenfold increase in data yielded enhancement in results, especially in low-light scenarios captured during evenings and nights, with an improvement of up to 11% and 19% in mAP@0.5 across all models. In most cases, the augmentation improved the performance of the models under varying visual conditions.

Our second approach involved the incorporation of self-attention mechanisms through transformer modules and global attention within the YOLOv8 detection pipeline. This refinement of architecture, together with an additional detection head for very small objects and a multi-scale feature module called ADASPP, also targeted the challenge of identifying potholes under degraded visibility. The Ablation study demonstrated that all included modules contribute to enhancing the detection accuracy of the model. When compared to the original YOLOv8, the implementation of the suggested modifications led to an 8.4% increase in accuracy (mAP@0.5) prior to the application of synthetic augmentation.

The proposed modification to YOLOv8 delivered accuracy on par with the detection transformer RT-DETR, all while keeping the same level of computational efficiency as the original YOLOv8. The proposed model achieved mAP@0.5, precision, and recall of 0.867, 0.909, and 0.796, respectively, while also maintaining a real-time inference speed of approximately 35 FPS on images with an input size of 1088 pixels. It's important to acknowledge that the RT-DETR-l model tested in this study surpasses our model in terms of model size and computational complexity. Therefore, in future experiments, we will focus on exploring innovative techniques to enhance the computational efficiency of the proposed architecture.

The limitation of this study lies in the fact that MUNIT inherently demands significant computational resources. In addition, the proposed method necessitates training a model for each condition individually, which is time-consuming. To address these challenges, future work could explore optimizing the architecture or employing more efficient training methods to reduce the computational burden. Additionally, investigating the trade-offs between model complexity and performance on larger datasets could provide valuable insights into making these techniques more scalable and applicable in various contexts. Our future efforts will also focus on exploring innovative approaches to image augmentation by leveraging synthesized images. Moreover, we will consider integrating these models with existing road infrastructure management systems to provide real-time feedback to traffic management systems, assisting in monitoring and improving road safety.

Funding Open access funding provided by The Ministry of Education, Science, Research and Sport of the Slovak Republic in cooperation with Centre for Scientific and Technical Information of the Slovak Republic. This project does not have any funding.

Data availability Data supporting reported results can be found at https://doi.org/10.6084/m9.figshare.21214400.v3.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Chen H, Yao M, Gu Q (2020) Pothole detection using location-aware convolutional neural networks. Int J Mach Learn Cybern 11(4):899–911. https://doi.org/10.1007/s13042-020-01078-7

2. Ye W, Jiang W, Tong Z, Yuan D, Xiao J (2021) Convolutional Neural Network for Pothole Detection in Asphalt Pavement. Road Mater Pavement Des 22(1):42–58. https://doi.org/10.1080/14680629.2019.1615533
3. Salaudeen H, Çelebi E (2022) Pothole detection using image enhancement GAN and object detection network. Electronics 11(12):1882. https://doi.org/10.3390/electronics11121882
4. Heo D-H, Choi J-Y, Kim S-B, Tak T-O, Zhang S-P (2023) Image-based pothole detection using multi-scale feature network and risk assessment. Electronics 12(4):826. https://doi.org/10.3390/electronics12040826
5. Singh G, Bansal D, Sofat S, Aggarwal N (2017) Smart patrolling: An efficient road surface monitoring using smartphone sensors and crowdsourcing. Pervasive Mob Comput 40:71–88. https://doi.org/10.1016/j.pmcj.2017.06.002
6. Li X, Goldberg DW (2018) Toward a mobile crowdsensing system for road surface assessment. Comput Environ Urban Syst 69:51–62. https://doi.org/10.1016/j.compenvurbsys.2017.12.005
7. Wu C et al (2020) An Automated Machine-Learning Approach for Road Pothole Detection Using Smartphone Sensor Data. Sensors 20(19):5564. https://doi.org/10.3390/s20195564
8. Aparna Bhatia Y, Rai R, Gupta V, Aggarwal N, Akula A (2019) Convolutional neural networks based potholes detection using thermal imaging. J King Saud Univ Comput Inf Sci 34:578–588. https://doi.org/10.1016/j.jksuci.2019.02.004
9. 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
10. Zhao Y et al (2024) Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 16965–16974
11. Gupta P, Dixit M (2022) Image-based crack detection approaches: a comprehensive survey. Multimed Tools Appl 81(28):40181–40229. https://doi.org/10.1007/s11042-022-13152-z
12. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 580–587
13. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1440–1448. https://doi.org/10.1109/ICCV.2015.169
14. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, Curran Associates, Inc., pp 91–99
15. Liu W et al (2016) SSD: single shot MultiBox detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision – ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham, pp 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
16. A Laha, N Zhang, L Li, (2018) 'Road damage detection using RetinaNet'. In: 2018 IEEE International Conference on Big Data (Big Data). IEEE,. p. 5197–5200
17. Pena-Caballero C, Kim D, Gonzalez A, Castellanos O, Cantu A, Ho J (2020) Real-Time Road Hazard Information System. Infrastructures 5(9):75. https://doi.org/10.3390/infrastructures5090075
18. Park S-S, Tran V-T, Lee D-E (2021) Application of Various YOLO Models for Computer Vision-Based Real-Time Pothole Detection. Appl Sci 11(23):11229. https://doi.org/10.3390/app112311229
19. Salcedo E, Jaber M, RequenaCarrión J (2022) A Novel Road Maintenance Prioritisation System Based on Computer Vision and Crowdsourced Reporting. J Sens Actuator Netw 11(1):15. https://doi.org/10.3390/jsan11010015
20. Mohan Prakash B, Sriharipriya KC (2022) Enhanced pothole detection system using YOLOX algorithm. Auton Intell Syst 2(1):22. https://doi.org/10.1007/s43684-022-00037-z
21. Deepa D, Sivasangari A (2023) An effective detection and classification of road damages using hybrid deep learning framework. Multimed Tools Appl 82(12):18151–18184. https://doi.org/10.1007/s11042-022-14001-9
22. Tayara H, Chong KT (2018) Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network. Sensors 18(10):3341. https://doi.org/10.3390/s18103341
23. Tang T, Zhou S, Deng Z, Zou H, Lei L (2017) Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. Sensors 17(2):336. https://doi.org/10.3390/s17020336
24. Silva LA, et al (2020) An architectural multi-agent system for a pavement monitoring system with pothole recognition in UAV images. Sensors 20(21):6205. https://doi.org/10.3390/s20216205

25. Xie X, Lang C, Miao S, Cheng G, Li K, Han J (2023) Mutual-assistance learning for object detection. IEEE Trans Pattern Anal Mach Intell 45(12):15171–15184. https://doi.org/10.1109/TPAMI.2023.3319634
26. Cheng G, Li Q, Wang G, Xie X, Min L, Han J (2023) SFRNet: fine-grained oriented object recognition via separate feature refinement. IEEE Trans Geosci Remote Sens 61:1–10. https://doi.org/10.1109/TGRS.2023.3277626
27. Gupta S, Sharma P, Sharma D, Gupta V, Sambyal N (2020) Detection and localization of potholes in thermal images using deep neural networks. Multimed Tools Appl 79(35):26265–26284. https://doi.org/10.1007/s11042-020-09293-8
28. Jakubec M, Lieskovská E, Bučko B, Zábovská K (2023) Comparison of CNN-Based Models for Pothole Detection in Real-World Adverse Conditions: Overview and Evaluation. Appl Sci 13(9):9. https://doi.org/10.3390/app13095810
29. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Process Syst 27:2672–2680
30. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434
31. Karras T, Aila T, Laine S, Lehtinen J (2018) Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=Hk99zCeAb
32. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T (2020) Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8110–8119. https://doi.org/10.1109/CVPR42600.2020.00813
33. Kang M, Zhu J-Y, Zhang R, Park J, Shechtman E, Paris S, Park T (2023) Scaling up GANs for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 10124–10134
34. Isola P, Zhu J-Y, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1125–1134
35. Zhu J-Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE. Venice, pp 2242–2251. https://doi.org/10.1109/ICCV.2017.244
36. Hertzmann A, Jacobs CE, Oliver N, Curless B, Salesin DH (2001) Image analogies. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, New York, pp 327–340. https://doi.org/10.1145/383259.383295
37. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
38. Huang X, Liu M-Y, Belongie S, Kautz J (2018) Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 172–189
39. Lee HY, Tseng HY, Huang J, Singh M, Yang MH (2018) Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 35–51
40. Jocher G, Chaurasia A, Qiu J (2023) Ultralytics YOLO (Version 8.0.0) [Software]. https://github.com/ultralytics/ultralytics
41. Jocher G (2024) GitHub - ultralytics/yolov5: YOLOv5 in PyTorch > ONNX > CoreML > TFLite. Available online: https://github.com/ultralytics/yolov5
42. Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D (2020) Distance-IoU Loss: faster and better learning for bounding box regression. Proc AAAI Conf Artif Intell 34(07):07. https://doi.org/10.1609/aaai.v34i07.6999
43. Li X, Wang W, Wu L, Chen S, Hu X, Li J, Tang J, Yang J (2020) Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Adv Neural Inf Process Syst 33:21002–21012
44. Intelligent Lab (2024) GitHub - iscyy/yoloair: YOLOAir: improved YOLO models and components. Available online: https://github.com/iscyy/yoloair
45. Yang M, Yu K, Zhang C, Li Z, Yang K (2018) DenseASPP for semantic segmentation in street scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA. IEEE, pp 3684–3692. https://doi.org/10.1109/CVPR.2018.00388
46. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450
47. Cao X, Zhang Y, Lang S, Gong Y (2023) Swin-transformer-based YOLOv5 for small-object detection in remote sensing images. Sensors 23(7):3634. https://doi.org/10.3390/s23073634

48. Liu Y, Shao Z, Hoffmann N (2021) Global attention mechanism: retain information to enhance channel-spatial interactions. arXiv preprint arXiv:2112.05561
49. Yu F et al (2020) BDD100K: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA. IEEE, pp 2633–2642. https://doi.org/10.1109/CVPR42600.2020.00271
50. HW (2023) Allfog dataset. Roboflow Universe. Available online: https://universe.roboflow.com/hw-zyvlq/allfog. Accessed 25 Jun 2024
51. Roboflow (2024) ACDCFOGG dataset. [Online]. Available: https://universe.roboflow.com/debasiskumaredugmailcom-oadi4/acdcfogg/dataset/3 . Accessed 25 Jun 2024
52. Bučko B, Lieskovská E, Zábovská K, Zábovský M (2022) Computer vision based pothole detection under challenging conditions. Sensors 22(22):8878. https://doi.org/10.3390/s22228878
53. Tremblay M, Halder SS, de Charette R, Lalonde J-F (2021) Rain rendering for evaluating and improving robustness to bad weather. Int J Comput Vis 129(2):341–360. https://doi.org/10.1007/s11263-020-01366-3
54. de Charette R et al (2012) Fast reactive control for illumination through rain and snow. In: Proceedings of the IEEE International Conference on Computational Photography (ICCP), Seattle, WA, USA. IEEE, pp 1–10. https://doi.org/10.1109/ICCPHOT.2012.6215217
55. Godard C, Mac Aodha O, Firman M, Brostow GJ (2019) Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), IEEE. Seoul, Korea, pp 3828–3838
56. Lin TY et al (2014) Microsoft COCO: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V. Springer International Publishing, pp 740–755. https://doi.org/10.1007/978-3-319-10602-1_48