Journal of Real-Time Image Processing (2024) 21:70
https://doi.org/10.1007/s11554-024-01451-7

RESEARCH

SDPH: a new technique for spatial detection of path holes from huge volume high-resolution raster images in near real-time

Murat Tasyurek[1]

Received: 4 January 2024 / Accepted: 3 March 2024

Abstract

Detecting and repairing road defects is crucial for road safety, vehicle maintenance, and enhancing tourism on well-maintained roads. However, monitoring all roads by vehicle incurs high costs. With the widespread use of remote sensing technologies, high-resolution satellite images offer a cost-effective alternative. This study proposes a new technique, SDPH, for automated detection of damaged roads from vast, high-resolution satellite images. In the SDPH technique, satellite images are organized in a pyramid grid file system, allowing deep learning methods to efficiently process them. The images, generated as 256×256 dimensions, are stored in a directory with explicit location information. The SDPH technique employs a two-stage object detection models, utilizing classical and modified RCNNv3, YOLOv5, and YOLOv8. Classical RCNNv3, YOLOv5, and YOLOv8 and modified RCNNv3, YOLOv5, and YOLOv8 in the first stage for identifying roads, achieving f1 scores of 0.743, 0.716, 0.710, 0.955, 0.958, and 0.954, respectively. When the YOLOv5, with the highest f1 score, was fed to the second stage; modified RCNNv3, YOLOv5, and YOLOv8 detected road defects, achieving f1 scores of 0.957,0.971 and 0.964 in the second process. When the same CNN model was used for road and road defect detection in the proposed SDPH model, classical RCNNv3, improved RCNNv3, classical YOLOv5, improved YOLOv5, classical YOLOv8, improved RCNNv8 achieved micro f1 scores of 0.752, 0.956, 0.726, 0.969, 0.720 and 0.965, respectively. In addition, these models processed 11, 10, 33, 31, 37, and 36 FPS images by performing both stage operations, respectively. Evaluations on geotiff satellite images from Kayseri Metropolitan Municipality, ranging between 20 and 40 gigabytes, demonstrated the efficiency of the SDPH technique. Notably, the modified YOLOv5 outperformed, detecting paths and defects in 0.032 s with the micro f1 score of 0.969. Fine-tuning on TileCache enhanced f1 scores and reduced computational costs across all models.

Keywords Near real-time · High-resolution imagery · Pat hole detection · Object detection with spatial location

1 Introduction

Damaged paths cause traffic accidents and injuries [1, 2]. Imperfections such as potholes, cracks, or trapezoids can make vehicles difficult to control, distract drivers, and cause them to make sudden maneuvers. Detecting damaged roads improves driver safety and helps prevent accidents [3]. However, it is very costly to constantly check all roads by assigning personnel to detect broken roads. On the other hand, with the widespread use of remote sensing technology in recent years, huge volume and high-resolution images are obtained [4–6]. High-resolution satellite images are images obtained from space or the air that show the details of the Earth in high resolution. These images usually have sufficient pixel density to distinguish very small objects or details [7]. However, these images take up a large amount of space on computer systems [8]. Large volumes of data are challenging to handle by classical data processing methods [9, 10].

Technology giants such as ArcGIS and Google Earth cut huge volumes of images into small pieces, allowing users to access these images at high speed [11–13]. Large-volume images are divided into small pieces in the form of small grids and accessed in pyramid grid file format [14]. This operation is called tile [15]. In the tiling process, huge volumes of satellite images are recorded on computer disks as z/x/y.mime type [16]. Z denotes the recording level of the image, x signifies the image's position along the X-axis, the y-value represents its position along the Y-axis, and the

---

✉ Murat Tasyurek
murattasyurek@kayseri.edu.tr

[1] Department of Computer Engineering, Kayseri University, Kayseri, Turkey

mime type specifies the format in which the image is to be stored. These images are produced and saved in 256 × 256 dimensions. When the user wants to access the image in a region, it takes a very long time to take the huge volume image and present the relevant region. Instead, the pyramid grid file system provides high-speed access to users because small-size images in the appropriate location are presented [17, 18].

In recent times, the advanced deep learning technique, a cutting-edge computer technology, has found widespread application in diverse domains such as image classification [19], object tracking [20], and pose estimation [21]. Numerous deep learning methodologies have gained widespread acceptance in this domain. Notably, among these techniques, convolutional neural networks (CNNs) have demonstrated remarkable efficacy, particularly in the realm of image classification [22].

Detection of road disturbances is a classification problem in computer vision. Within the literature, research on the identification of road disturbances through onboard cameras often employs CNN-based methods [23–25]. In this investigation, we propose a new method, referred to as SDPH, for identifying road disturbances along with their spatial coordinates within huge volumes and high-resolution satellite imagery. CNN models could not process satellite images ranging in size from 20 to 40 gigabits. In the suggested SDPH methodology, the conversion of these images into a pyramid grid file format is facilitated through the GeoServer-TileCache software, employing an open-source strategy. This format ensures compatibility for processing the images through CNN models. In the satellite images, there are disturbances in the soil. In the recommended SDPH technique, a two-stage deep learning technique has been developed to detect only road disturbances. As a deep learning method, RCNNv3 [25, 26], YOLOv5 [24, 27], and YOLOv8 [28, 29] were used.

1.1 Contributions

- A new technique, SDPH, is recommended to detect road disturbances and their corresponding spatial locations within vast volumes of high-resolution satellite imagery.
- The achievement of the recommended technique has been tested on real satellite image data with sizes between 20 and 40 gigabytes.
- A pyramid grid file system using GeoServer-TileCache has been recommended so that CNN models can process huge volume satellite images.
- Thanks to the recommended pyramid grid technique, a huge volume satellite image that occupies at least 20 gigabytes of space and, therefore cannot be processed has been converted into small and processable images, the largest of which is 23 kilobytes.
- A fine-tuning technique has been developed to improve the object detection achievement of the recommended SDPH technique. Thanks to the modified process, the micro f1 score achievement of the RCNNv3 model was increased by 0.204, while the micro f1 score achievement of the YOLOv5 and YOLOv8 were improved by 0.243 and 0.245, respectively.
- Thanks to the fine-tuning process performed in the Pyramid grid technique, the roads are visible as a whole at the 22nd zoom level.
- A two-stage technique is recommended to detect damaged roads, first road detection and then damaged road detection. In this way, the disturbances in the soil are eliminated.
- In the recommended SDPH technique, the modified YOLOv5 model outperformed by detecting damaged roads with a 0.969 micro f1 score on approximately 0.032 s.

1.2 Scope and outline

- In the context of this study, it is aimed to determine the location of the roads as points while determining their location. Determining the roads as polygons is out of the scope of this study.

The following organization is maintained in the rest of this paper: Sect. 2 provides an overview of the related researches. Sect. 3 explains the basic concepts. Section 4 introduces the technique put forth in this study. In Sect. 5, we delve into the evaluations carried out during experiments. Lastly, Sect. 6 outlines the conclusions drawn from the study and highlights avenues for future research.

2 Related work

Detection of damaged roads using CNN-based methods from satellite images has been examined in three categories. First, CNN models were presented, then studies on detecting damaged roads with CNN models were discussed, and finally, studies on object detection from satellite imagery were examined.

When exploring the evolution of CNN models; The CNN technique [30], a notable deep learning technique, has seen extensive application across diverse domains including computer networks [31], image detection [32], and disease classification [33] in recent times. Koller et al. [34] introduced CNN-based deep sing for continuous sign language recognition. Ciocca et al. [35] investigated the use of CNN-based features for the purpose of food recognition and retrieval. Achour et al. [36] used CNN to detect the dairy cow presence in the feeder zone. Bermejo et al. [37] used ensemble of deep convolutional neural networks for classification of interstitial lung abnormality patterns. Rao et al. [38] used CNN to detect the signs on the road in real-time and provide alerts to the driver to perform the action corresponding to the sign. Yang et al. [39] recommended a lenet-based low-computation neural assistance system for traffic sign recognition. Cao et al. [40] improved on the basis of the classical LeNet-5 convolutional neural network model for real-time traffic sign recognition. Lenet-5 is a 7-level convolutional network developed by LeCun et al [41]. Nevertheless, the capacity of the convolutional network falls short when attempting to analyze high-resolution images using the LeNet-5 model [42]. CNN models used are continuously improved to get higher accuracy and results faster [43]. For instance, AlexNet, developed in 2012, showed better results than previous CNN methods [44]. VggNet, developed in 2014, showed a very successful achievement in the ImageNet competition with an error rate of 7.3% [45]. GoogleNet, developed in 2014 and has a low error rate of 5.7%, was the winner of the ILSVRC competition [46]. Resnet was developed in 2015 and had a shallow error rate of 3.6% [47, 48]. The CNN technique has demonstrated success in feature extraction and classification for analyzing single-object images. Nevertheless, its efficacy in the context of multi-object image analysis has proven to be constrained. To tackle this issue, Girshick et al. introduced the RCNN method [49]. RCNN partitions the image into around 2000 regions, employing CNN within each region to address the challenge of multi-object analysis. However, the RCNN method incurs a high computational cost, primarily in terms of time. To address this constraint, Girshick introduced Fast RCNN (RCNNv2) [50], which alleviates the slow execution problem associated with RCNN. RCNN algorithms utilize regions to localize objects within an image. In addition to RCNN, another CNN-based method called YOLO was introduced by Redmon and others [51]. YOLO takes a different technique by directly examining parts of the image that are likely to contain the object, instead of dividing it into regions like RCNN. This strategy helps to mitigate the computational overhead associated with region-based methods.

When the studies on detecting damaged roads with CNN-based methods are examined, Maeda et al. [23] recommended a method using deep learning methods for detecting and classifying road defects from images captured with a mobile phone. In their recommended method, they achieved 0.95 accuracy with SSD MobileNet. Rath [24] studied the detection of road defects using images obtained from the camera mounted on the vehicle. With the YOLOv7 tiny, it achieved 35% mAP while it achieved 51% mAP with the classic YOLOv7. Parvathavarthini et al. [25] conducted a study on detecting damaged roads with CNN-based deep learning networks. They compared the achievements of classical CNN and RCNN models. Classic CNN achieved 88.43% accuracy, while RCNN achieved 97% accuracy.

When the studies on object detection from satellite imagery are examined, Kawauchi et al. [52] recommended a SHAP-based interpretable object detection technique for object detection from satellite images. In their recommended technique, they present a feature association technique that calculates to describe an approximate model and attributing input features to the output of a deep learning method. With the recommended technique, they achieved a f1 score of 0.85. Wu et al. [53] recommended a two-step technique to detect objects around airports and ports, together with their surrounding objects, from high-resolution satellite images. In their recommended technique, they first identify airports and ports at the lower scale. Then they move to the upper scale and detect small objects around these objects. They achieved a 0.92 f1 score achievement in the DC-FRCNN technique they suggested. Song et al. [54] recommended an technique using salience detection and CNN models for hierarchical object detection from huge volume satellite images. In their recommended method, they detected airports with 0.949 precision from satellite images and detected ships with 0.915 precision. Gong et al. [55] suggested an technique they named SPH-YOLOv5 for detecting objects with smaller dimensions from satellite images. In their recommended technique, they replaced the original convolutional prediction heads of the YOLOv5 model with swin transformer prediction heads. They obtained 0.806 precision with their recommended technique. Khan et al. [56] proposed a consolidated deep learning framework comprising multi-scale detectors for multi-class object detection in high-resolution satellite imagery. This suggested methodology involves a two-stage process. In the initial stage, multi-scale object proposals are generated, and subsequently, each proposal undergoes classification into distinct classes in the second phase. They achieved 0.97 precision success with their recommended technique.

Although there are studies in the literature on detecting damaged roads with a camera or mobile phone images mounted on the vehicle, obtaining photographs by traveling all roads with cars is costly. On the other hand, to the best of our knowledge, there has been no study related to detecting road disturbances along with their spatial coordinates using huge volume satellite image datasets. In the scope of this research, a new hybrid method, including a two-stage deep learning method, is recommended for detecting road disturbances together with their spatial locations from a huge volume and high-resolution satellite images. In the recommended hybrid method, the fine-tuning technique has been developed to improve the object detection achievement of deep learning methods. Free GeoServer and TileCache software are used, which allow access as pyramid grid files to process huge volume and high-resolution satellite images without sacrificing resolution quality and similar features. In the literature, YOLO and RCNN models outperform CNN models in studies for detecting road defects [24, 25]. Therefore, In the scope of this research, Faster-RCNN, which shows superior success in region-based object detection, and YOLO models (v5, v8), which achieves high achievement in real-time object detection, were preferred.

3 Basic concept

In the scope of this research, a new hybrid technique is recommended to detect road disturbances from huge volume and high-resolution satellite images. The recommended hybrid technique consists of large-scale imagery, pyramid grid access, and object detection with deep learning. For this reason, in this section, huge volume satellite images, pyramid grid access, and object detection methods are explained, respectively.

3.1 Large scale high resolution satellite images

Large volumes of data are datasets that cannot be processed effectively by traditional data processing methods and are difficult to store, manage, analyze, and interpret [57]. On the other hand, high-resolution satellite images are the images taken by satellite systems or drones. These high-resolution images enable detailed observation and analysis of the Earth's surface features [58]. High-resolution satellite images consist of pixels that show objects on the ground down to fine detail. Each pixel represents the light intensity in a specific area. The smaller the pixel size of high-resolution satellite images, the more detailed an image is obtained. However, processing huge volume and high-resolution satellite images is difficult because it requires computer systems with high-end features. In the scope of this research, high-resolution satellite images taken from Kayseri Metropolitan Municipality were used. The dataset consists of images in geotiff format with sizes ranging from 20 to 40 gigabytes. An image with a minimum of 20 gigabytes is challenging to process using conventional data processing methods. On the other hand, Google Earth and ArcGIS publish huge volumes of data via pyramid grid file system [12, 13]. There are three variables in the Pyramid grid system, z, x, and y. The Z value shows the scale (zoom level) information. The Pyramid grid system is presented in the Fig. 1.

As depicted in Fig. 1, the Pyramid grid system exhibits a single image at the initial zoom level, with this count quadrupling at each subsequent level. x and y represent the data coordinates at the same zoom level. The Pyramid grid system transforms huge volume satellite images into small images recorded with location information. Since these images are 256 × 256 by default, they are easy to process and publish.

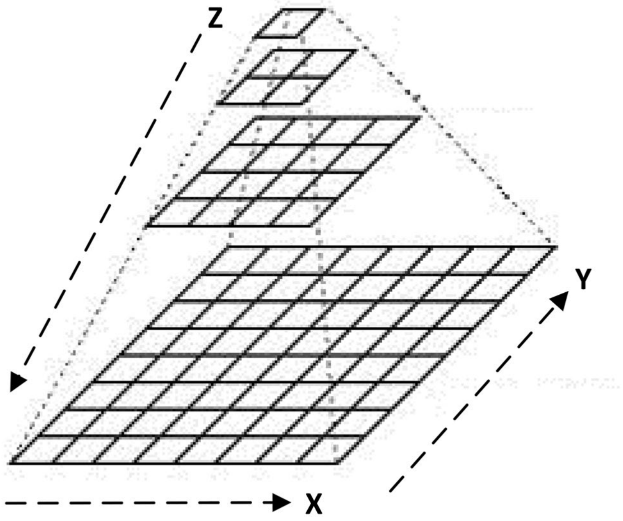


Fig. 1 Pyramid grid file system

3.2 Object detection

The process of detecting the position and class of certain objects in digital images or videos is called object detection. Object detection is an artificial intelligence technique widely used in computer vision. This research leveraged deep learning techniques rooted in CNN, a subset of artificial intelligence. Specifically, YOLO and RCNNv3 models were employed as instances of CNN-based deep learning methods within this study. Therefore, in this section, YOLO and RCNNv3 models are presented respectively.

3.2.1 YOLO

The YOLO methodology derives its name from the acronym 'You Only Look Once,' signifying a single comprehensive examination [51]. This technique enables swift and holistic predictions regarding the identity and spatial localization of objects within an image. Object detection is a computer vision problem that falls specifically into discovering what objects are in a given image and where they are in the image. Object detection is a more complex and challenging problem than image classification problems since it also finds where the object is in the image.

The YOLO technique has seen continuous development over time. The initial version, YOLO V1, was introduced by Redmon et al. [59]. YOLO V2 offered improved accuracy and speed and expanded its object recognition capabilities to 9000 objects. Redmon and others advanced the YOLO series with YOLO V3 [60]. YOLO V3 featured a more complex architecture compared to its predecessors. One notable feature was the ability to adjust the model's structure size, enabling flexibility in balancing speed and accuracy. Then, in 2020, Bochkovskiy et al. introduced YOLO V4 as an object detection method optimized for speed and accuracy [61]. Throughout its evolution, the YOLO series has consistently pursued practical and powerful object detection models, culminating in the YOLO V4 release. YOLOv5 model was introduced by Jocher in 2020 [62]. In contrast to the V4 model, the YOLOv5 model is instantiated using PyTorch. Previous research [63, 64] has illustrated that the YOLOv5 model provides more precise estimations at a reduced computational cost compared to the V4 model. Unlike its predecessors implemented in C, YOLOv5 was developed using Python, and it constitutes the chosen model for this study.

Moreover, YOLOv8 [65] improves on the features of previous versions, providing a balance of speed and accuracy. YOLOv8 is designed specifically for use in real-time applications and is optimized for high-speed object detection. For that reason, this study also used the YOLOv8 model.

3.2.2 RCNNv3

Faster RCNN [66] (RCNNv3) is an object detection library built on deep convolutional networks, including the Region Proposal Network (RPN) and the object detection network. The RCNNv3 model is the most widely used and advanced version of the RCNN models. The most significant difference between RCNN methods has been computational efficiency, reduction of experiment time, and performance improvement. RCNN was designed by Girshick and others to overcome the multi-object detection [49]. Addressing the issue of slow performance in RCNN, Ross Girshick introduced the Fast RCNN (RCNNv2), designed to operate more efficiently [50]. The RCNN architecture is designed for object detection in images, aiming to identify object classes along with their corresponding bounding regions. RCNNv3, recommended by Ren et al. in 2015 [66], introduced a significant improvement over the RCNNv2. In RCNNv2, the utilization of the selective search method for region proposals presented a bottleneck for the entire architecture. To address this limitation, the RCNNv3 model replaced the selective search method with a region proposal network (RPN). The RCNNv3 model follows a two-stage process. Compared to both the RCNN and Fast RCNN models, the RCNNv3 model offers improved computational efficiency, making it faster in terms of computational cost [66]. Additionally, the RCNNv3 model demonstrates superior performance metrics compared to RCNNv1 and RCNNv2, specifically in terms of mAP. Consequently, this study employed the RCNNv3 model.

4 Proposed technique

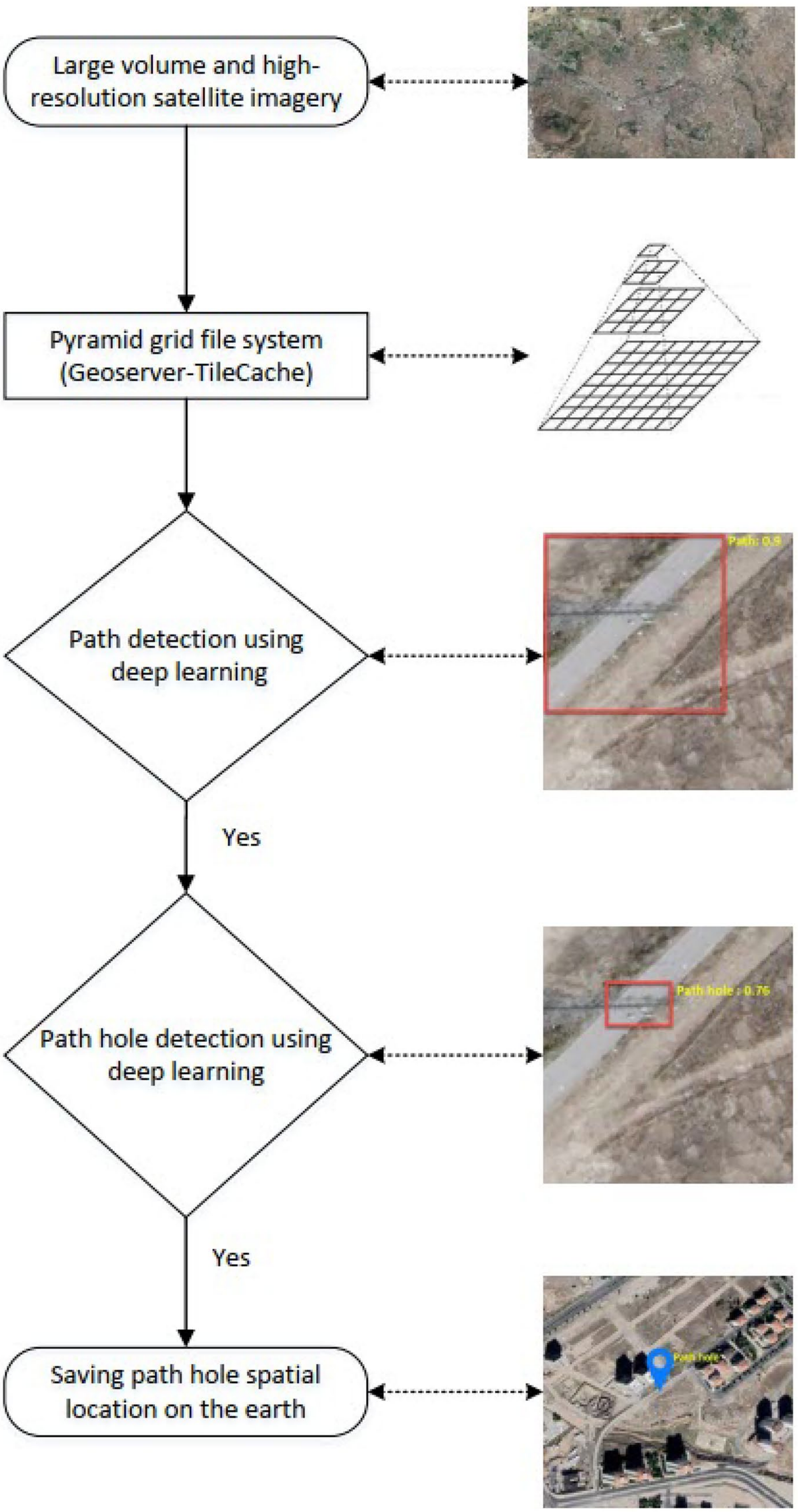
Various problems affecting the roads' physical condition are expressed as road disorders. These disturbances can cause roads to deviate from providing a smooth and safe driving surface. For this reason, it is important to detect the defects on the roads with their locations. However, it is not possible in terms of personnel and fuel costs to control all roads instantly with the vehicle to detect road defects. On the other hand, in recent years, remote sensing technologies have obtained earth images with advanced centimeter precision. Since these acquired images are in high resolution, the detection of damaged roads can be fulfilled on these images. However, these large images are difficult to process and analyze by classical data processing methods. Therefore, in the context of this study, a new method, SDPH (spatial detection of path hole), is recommended to detect the disturbances of paths with their real locations on Earth from huge volume and high-resolution satellite images. The architecture of the recommended method is presented in Fig. 2.

As presented in Fig. 2, a huge volume and high-resolution satellite image are given as input to the recommended system. The images used within the scope of the study occupy the smallest 20 gigabytes. It is very difficult to detect an object by processing such a large amount of data with a computer with standard features. For this reason, it has been recommended to use the pyramid grid file system, which is used for accessing huge volume images with high performance.

The Pyramid grid file system was created using the open-source strategy Geoserver and TileCache system in the recommended technique. GeoServer [67] is free software that stores geographic data in many different formats and publishes it as a web map service (WMS). TileCache [68] is a tile system that improves performance in geodata services. TileCache saves data published as WMS to computer disks as z/x/y.jpg. In the Geoserver-TileCache system, the data published by the GeoServer as WMS is saved to the disks by TileCache. After this process is finished, accessing the images is done via TileCache. The TileCache process performs well because it presents small images in response to requests and does not take any action. The recommended GeoServer-Tilecache-based pyramid-grid file system creates 256 × 256 images. The largest of these images created takes up 23 kilobytes of space on the disk. Thanks to this recommended technique, the image has been converted into small images that take up at least 20 gigabytes, the largest of which is 23 kilobytes.

In the scope of this research, RCNNv3, YOLOv5 and YOLOv8 models, widely used in recent years, were used

Fig. 2 System architect for spatial detection of path hole


Large volume and high-resolution satellite imagery
Pyramid grid file system (Geoserver-TileCache)
Path detection using deep learning
Path: 0.9
Yes
Path hole detection using deep learning
Path hole : 0.76
Yes
Saving path hole spatial location on the earth
Path hole

as a CNN-based model for road defect detection. Road disturbances are the regions of broken or contrary objects on the roads. RCNNv3 was preferred because it is an area-based model that searches for regions in the image. The YOLOv5 and YOLOv8, on the other hand, were chosen because of their high performance in real-time object detection.

In the scope of this research, road defect detection was tried by using the data produced by the GeoServer-TileCache system. However, the RCNNv3, YOLOv5 and YOLOv8 detect the disturbances in the fields as road disturbances. For this reason, a two-stage deep learning technique was used to detect the path disorder. First, models are trained separately for road and road disturbance detection. Then, as presented in the system design, the 256 × 256 image taken from the pyramid grid system is checked to see if it is a road. Upon successful detection of a path in the image by the trained deep learning methods for path detection, the subsequent step involves path defect detection.

Images at 22 levels ($z = 22$) were examined to stage high performance for road detection and defect detection on the road. However, deep learning methods have not been successful enough because a path at this level is in more than a small image. To overcome these issues and enhance the performance of deep learning methods, the adoption of fine-tuning techniques has been suggested. The default size parameter of the TileCache system is 256, 256. It allows the TileCache system to produce small images with a size of 256 × 256. Within the fine-tuning technique's scope, this parameter's value is set as 1024,1024. After this process, the TileCache system produces 1024 × 1024 images. The achievement of the recommended fine-tuning technique is examined in detail in the experimental sections.

If the image recorded in z, x, y format, presented as input in the recommended technique, is detected as a path by the first deep learning method, the same input image is given as input to the second deep learning method. If a path disorder is detected in the input image by the second deep learning method, the spatial location of this image is recorded in the database system. Spatial location refers to the geographic or physical location of an object or an event on earth [69, 70]. Spatial location can be determined using latitude and longitude coordinates. Latitude expresses the distance of a point from the equator, while longitude expresses the distance of a point from Greenwich. Combining these two coordinates can determine the spatial position of any point. Since the z, x, and y values of the image presented as input to the deep learning method are known, its position on the world is calculated as shown in the Algorithm 1 [71].

Algorithm 1 Get Latitude and Longitude from z, x, y

```
function GetLatitudeLongitude($z, x, y$)
    lon = $(x/Math.pow(2, z) * 360 - 180)$;
    lat = $(180/Math.PI * Math.atan(0.5 * (Math.exp(n) - Math.exp(-n))))$;
    return lon, lat
end function
```

The z, x, and y values presented as input in the Algorithm 1 is the registered address of the input image. Because the image is saved in the directory as z/x/y.jpg by TileCache. The Algorithm 1 returns latitude and longitude information. At these values, it ensures that the broken road is shown as a point in the world.

The achievement of the recommended SDPH technique is examined in detail in the experiments section.

5 Experimental evaluation

In this section, the achievement of recommended SDPH techniques for the detection of road disturbances from huge volume and high-resolution satellite images, along with their locations, is examined. The recommended SDPH technique consists of two stages: road detection and road disturbance detection. In the recommended SDPH technique, first of all, it is determined whether an image is a road or not; if this image is a road, it is determined whether there is a disorder. In order to evaluate the achievement of the recommended technique, answers to the following questions were sought respectively.

- What is the spatial location of paths detected by SDPH techniques?
- What is the path detection achievement of SDPH techniques?
- What is the damaged path detection achievement of the SDPH techniques?
- What is the run time of the SDPH techniques?
- What is the individual performance of deep learning-based object detectors in the SDPH technique?

The experiments in this study were carried out on a desktop computer with Intel i9 12,900 3.19 GHz, 64 GB Ram, 12 GB QUADRO graphics card, 2 TB SSD, and Windows 10 Pro operating system installed. Python 3.9 and Open-CV library were used for plate detection with deep learning.

5.1 Dataset, model setting and metrcis

In the scope of this research, huge volume and high-resolution satellite images taken from Kayseri Metropolitan

Municipality were used. There are 11 satellite images taken at different times in the dataset. The smallest of the images in geotiff format takes up 21 gigabytes, and the largest takes up 39 gigabytes. GeoServer software with open source strategy, this data has been published as WMS. Creating a layer group in GeoServer presents the data as a whole. The data presented as WMS by GeoServer were created with the TileCache system in small sizes. In the classical TileCache system, a total of 312,352 images with a size of 256 × 256 at 22 levels were analyzed as test images. In the modified TileCache system, 19,522 test images with dimensions of 256 × 256 at 22 levels were used as test images.

In the scope of this research, RCNNv3, YOLOv5 and YOLOv8 models were used as CNN models. The labeling process was done with the LabelImg program. Files in.txt format are used for the YOLO models. For RCNNv3. xml files in Pascal VOC format were used. Labeling was done once. The data in YOLO format was converted to the RCNNv3 model with a script written in the Python environment. In this way, it is ensured that the models work under equal conditions. Classic TileCache system produces 256 × 256 images. The modified TileCache recommended in the context of this study produces 1024 × 1024 images. CNN models using the dataset produced by the classic TileCache are called classical. CNN models using the modified data set are called modified. Models are trained separately for road detection and damaged road detection. The loss value produced by the CNN models in the path detection process is presented in Fig. 3.

The loss value of deep learning methods measures how far the model's predictions are from the actual values. A reduction in the Loss value, techniqueing zero, signifies the successful training of the deep learning method. As presented in Fig. 3, the loss value has decreased and techniqueed zero. This reveals that the models were trained successfully. The loss value of modified models is lower than classical models. This indicates that the prediction and real values of modified models are closer in the validation process.

The achievements of machine learning methods are evaluated with metrics. To obtain the metrics, *TP* (true positive), *TN* (true negative), *FP* (false positive), *FN* (false negative) values must be calculated. The meanings of the terms *TP*, *TN*, *FP*, and *FN* are explained below:

- *TP* indicates that there is an object, there is a detection, a box is drawn where the object is.
- *TN* is expressed as there is no object, and there is no detection.
- *FP* means that there is no object, there is a detection, a box has been drawn anywhere other than the object.
- *FN* denotes object present, no detection, no box drawn even though the object is present.

F1 score, which is called the expression of the accuracy of a deep learning method on a specific data set, is calculated using *TP*, *TN*, *FP*, and *FN* values. The F1 score value consists of the harmonic mean of the calculated *Precision* and *Recall* values of the model. The F1 score, *Precision* and *Recall* are calculated as presented in Eq. 1.

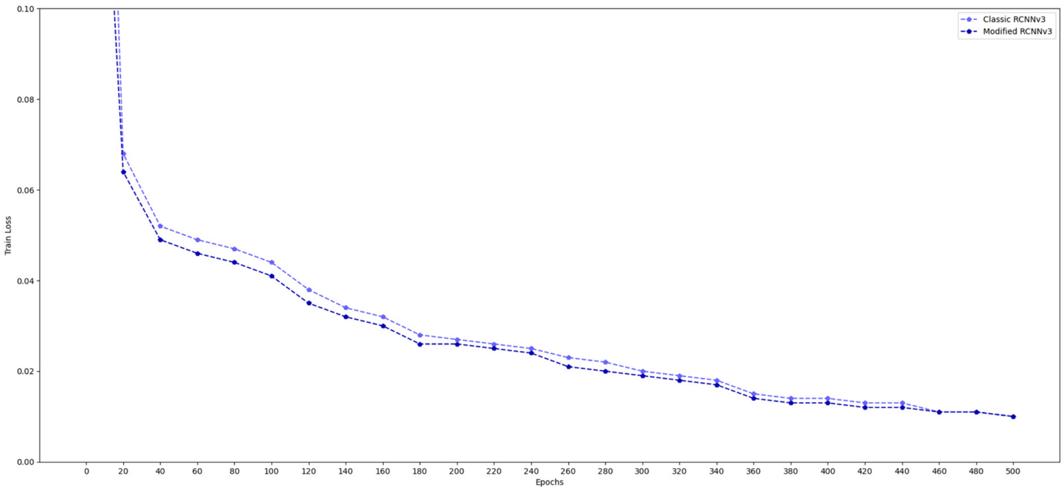
$$\begin{aligned}
Accuracy &= \frac{TP + TN}{TP + FP + TN + FN} \\
Precision &= \frac{TP}{TP + FP} \\
Recall &= \frac{TP}{TP + FN} \\
F_1 - score &= \frac{2 \times Precision \times Recall}{Precision + Recall}
\end{aligned} \tag{1}$$

5.2 Experiments

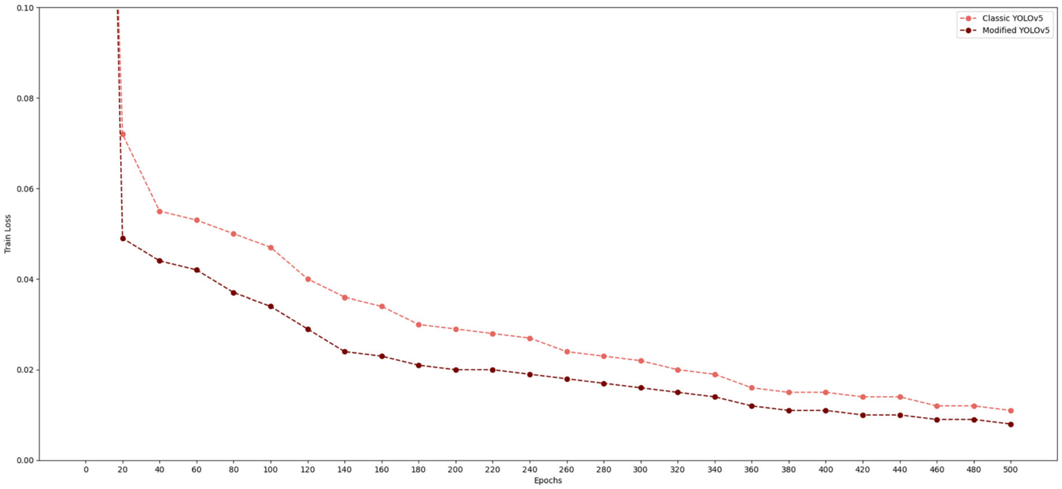
In this section, recommended SDPH techniques for spatial path hole detection from the huge volume and high-resolution satellite images, are discussed. First, the techniques' spatial path detection was examined, then the deep learning metrics and path detection achievement were compared. Next, the path hole detection achievement was presented, and then, the working time of the techniques was examined, and finally, the individual performance of CNN models in the SDPH technique was illustrated.

5.2.1 Geolocation of paths detected by SDPH techniques

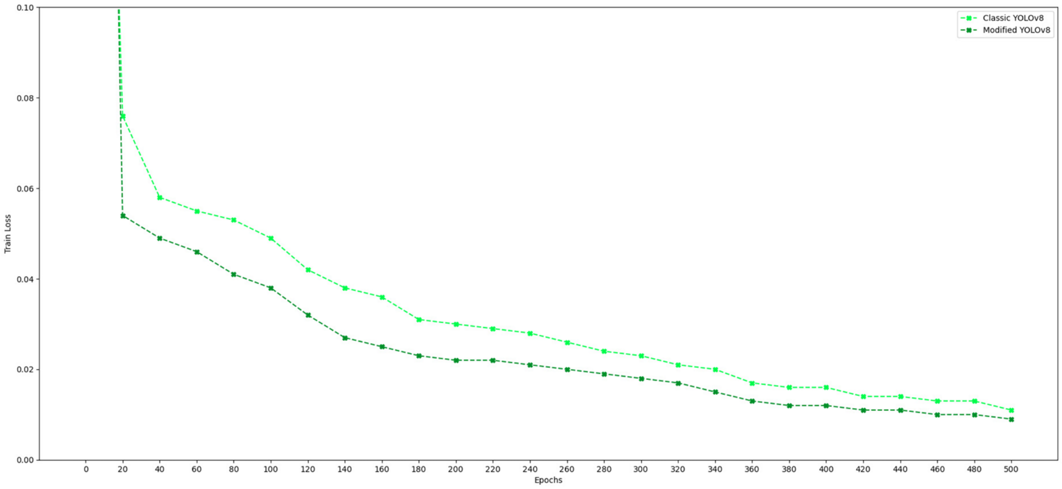
Within the scope of this experiment, the locations of the roads in the world, which were determined by the SDPH technique from huge volume and high-resolution satellite images, were examined. The recommended SDPH technique makes large-scale satellite image accessible in pyramid grid file format with GeoServer-TileCache software. This way, huge volume images were converted into small images addressed with location information. While classical TileCache produced 256 × 256 images, the modified TileCache recommended in the context of this study produced 1024 × 1024 images. In the recommended SDPH technique, RCNNv3, YOLOv5 and YOLOv8 models were run separately for both classical TileCache and modified TileCache for path detection. The open-source leaflet library [72] and the OpenStreetMap, serving as the map base [73], are employed to visually represent the global positions of roads identified through the proposed SDPH methodologies. The location of the roads detected by the recommended SDPH techniques is presented in Fig. 4.



(a) Training loss of RCNNv3 models

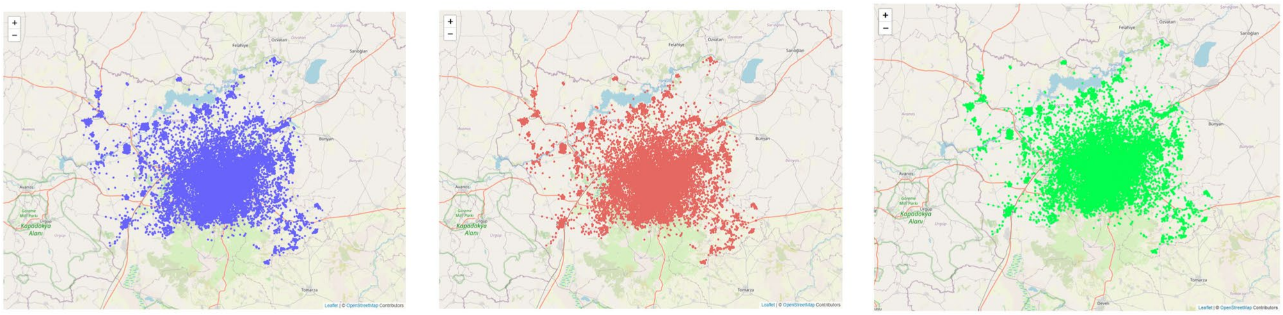


(b) Training loss of YOLOv5 models

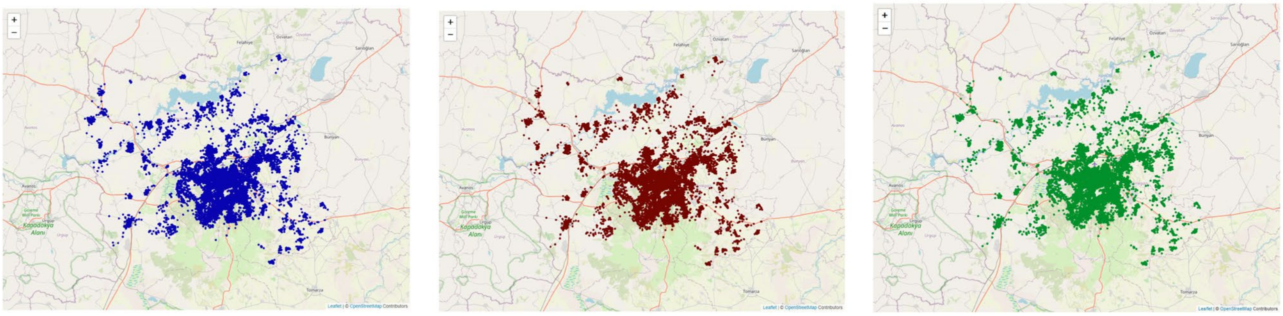


(c) Training loss of YOLOv8 models

Fig. 3 Training loss of CNN models



(a) Detected roads by RCNNv3 in size 256 × 256 (b) Detected roads by YOLOv5 in size 256 × 256 (c) Detected roads by YOLOv8 in size 256 × 256



(d) Detected roads by RCNNv3 in size 1024 × 1024 (e) Detected roads by YOLOv5 in size 1024 × 1024 (f) Detected roads by YOLOv8 in size 1024 × 1024

Fig. 4 Spatial location of detected roads by SDPH techniques

The position of the roads presented in Fig. 4 was calculated using Algorithm 1. In Fig. 4a, the RCNNv3 model using 256 × 256 tile images detects 233,747 road locations, while the RCNNv3 model, using 1024 × 1024 tile images, has been determined 19,683 road location in Fig. 4d. On the other hand, the YOLOv5 model, which takes 256 × 256 images as input, detects 190,626 road locations, while the YOLOv5 model, which takes 1024 × 1024 images, detects 18,497 road locations. The locations detected by the YOLOv5 model, which processes 256 × 256 and 1024 × 1024 images, are shown in Fig. 4b and e, respectively. In addition, YOLOv8 models detected 192,987 and 18,549 road locations, as shown in Fig. 4c and f, respectively. An image produced by modified TileCache in 1024 × 1024 size includes 16 images created by classical TileCache with a dimension of 256 × 256. For this reason, CNN models that process 256 × 256 images detect more positions than CNN models that process 1024 × 1024 images. RCNNv3 model, which processes 256 × 256 images, determined road location 11.88 times compared to the RCNNv3 model, which processes 1024 × 1024 images. Similarly, the YOLOv5 model, which processes 256 × 256 images, detected the road position at 10.31 times that of the YOLOv5 model, which processes 1024 × 1024 images. In addition, the YOLOv8 model, which processes images with dimensions of 256 × 256, detects the road location 10.42 times that of the YOLOv8 model, which processes images with dimensions of 1024 × 1024. However, when the location of the roads detected in Fig. 4 is examined, it is seen that the RCNNv3, YOLOv5 and YOLOv8 models, which process 256 × 256 images, also detect non-road points as roads. The detection achievement of road locations depends on the detection achievement of roads. The path detection achievement of the recommended SDPH techniques was examined as a separate experiment and presented in detail in the following experiment.

5.2.2 Path detection achievement of SDPH techniques

In this experiment, the path detection achievement of SDPH techniques from huge volume and high-resolution satellite images was investigated. SDPH techniques were run separately for classic TileCache, which produces 256 × 256 images, and modified TileCache, which has 1024 × 1024 images. The CNN model using images created by classic TileCache is called classic, while the CNN model using images produced by modified TileCache is expressed as modified. The path detection achievement of the recommended SDPH techniques is calculated with deep learning metrics and presented in Table 1.

In Table 1, classical RCNNv3, YOLOv5 and YOLOv8 models were tested on 312,352 images, while modified

Table 1 Metric results of all techniques for path detection

Technique	TP	TN	FP	FN	Acc.	Pre.	Rec.	F1 S.
Classic RCNNv3	**202,932**	3900	30,815	**109,420**	**0.596**	0.868	**0.650**	**0.743**
Classic YOLOv5	179,990	4200	**10,636**	132,362	0.563	**0.944**	0.576	0.716
Classic YOLOv8	179,451	4200	13,536	132,901	0.556	0.930	0.575	0.710
Modified RCNNv3	**18,712**	300	971	**810**	0.914	0.951	**0.959**	0.955
Modified YOLOv5	18,213	300	**284**	1309	**0.921**	**0.985**	0.933	**0.958**
Modified YOLOv8	18,165	300	384	1357	0.914	0.979	0.930	0.954

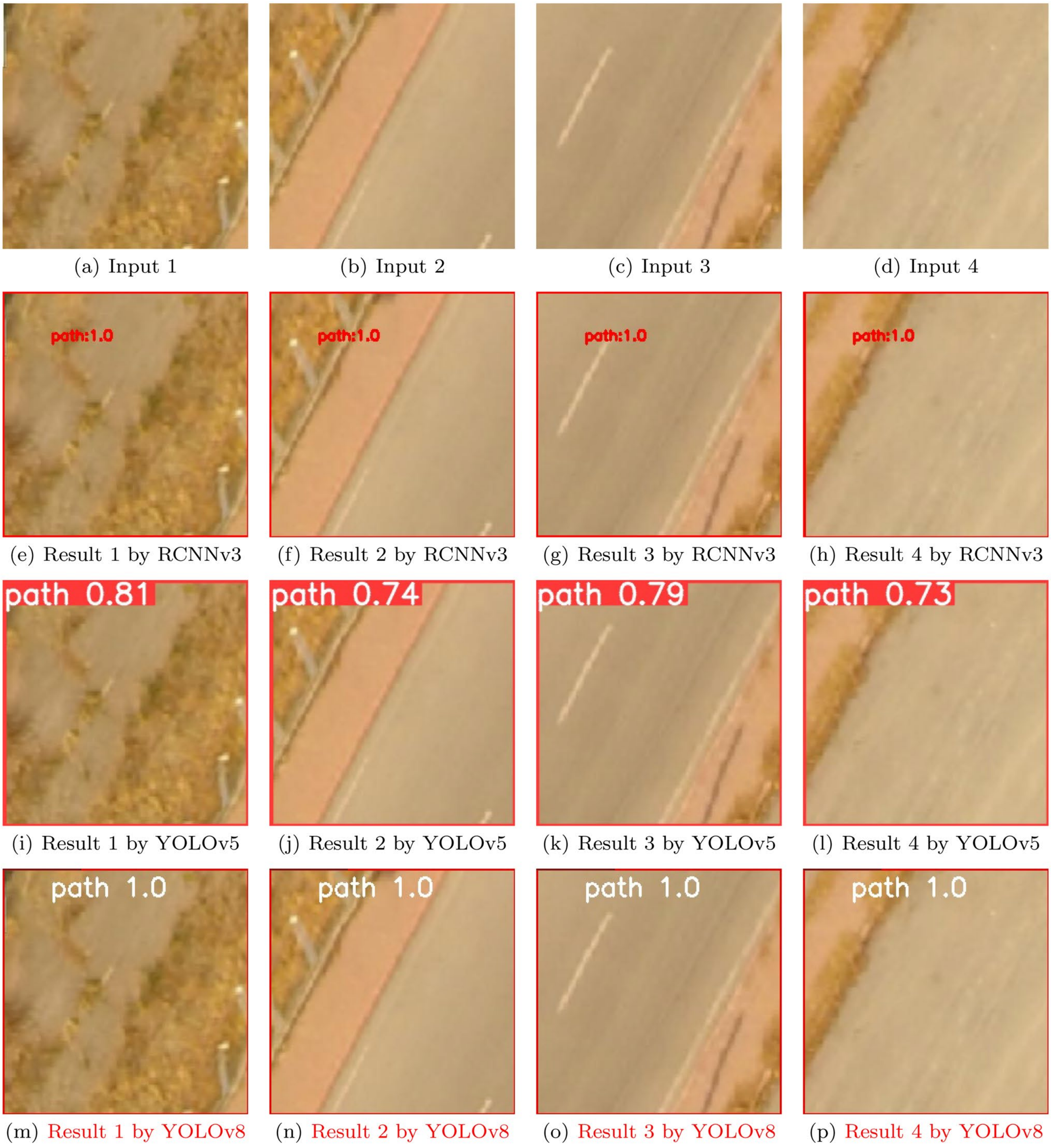
RCNNv3,YOLOv5 and YOLOv8 models were tested on 19,522 images. Superior metrics are marked in bold in the Table 1. Since the input image numbers of the classical and modified models are different, it is necessary to evaluate the models according to the metrics that show the ratios of these metrics instead of the TP, TN, FP, and FN numbers. The classic RCNNv3, YOLOv5 and YOLOv8 models scored 0.743, 0.716 and 0.710 f1, respectively, while the modified RCNNv3, YOLOv5 and YOLOv8 models scored 0.955, 0.958 and and 0.954 f1, respectively. In other words, thanks to the recommended modified technique, the f1 score achievement of the RCNNv3 model was increased by 0.211, while the f1 score achievement of the YOLOv5 and YOLOv8 models were improved by 0.242 and 0.244, respectively. CNN models using the recommended modified TileCache show much better achievement than CNN models using classical TileCache. To better describe the object detection achievement resulting from the classical TileCache and modified TileCache model, 256 × 256 and 1024 × 1024 images corresponding to the exact location were examined, and their results are shown in Figs. 5 and 6, respectively.

256 × 256 tile images presented in Fig. 5a–d as inputs to the classic CNN model presented. The image presented in Fig. 5a contains no path, while the image presented in Fig. 5b–d contains path. As shown in the images presented in Fig. 5e–h, classical RCNNv3 classifies all images as 'path'. However, when detecting TP in RCNNv3 Fig. 5f–h, while detecting the official path detected as FP in Fig. 5e. That is, the classical RCNNv3 model detected the image as an official road that does not fall into the road class type. Similarly, the classic YOLOv5 model detects as TP in Fig. 5j–l, while Fig. 5i' also detected as FP. Moreover, the classic YOLOv8 model made the same classifications as the classic RCNNv3 and YOLOv5 model. Classic RCNNv3, YOLOv5 and YOLOv8 models, which process 256 × 256 images, have high FP values as presented in Table 1 due to such false detections. Due to the high FP value of these models, they were not successful enough in path detection. To overcome these issues and improve the object detection performance of deep learning models, a fine-tuning process has been developed on TileCache. Sample results showing the performance of deep learning methods on images produced by fine-tuned TileCache are presented in Fig. 6.

The tile images of 1024 × 1024 presented in Fig. 6a are given as input to the Fine-Tuned CNN model. The detection results of Modified RCNNv3, YOLOv5 and YOLOv8 models are presented in Fig. 6b–d, respectively. Fined-tuned RCNNv3,YOLOv5 and YOLOv8 models detected TP as seen in Fig. 6. Due to such successful detections, modified CNN models outperform CNN models using classical TileCache, as seen in Table 1. CNN models are unsuccessful because the paths are divided in classic TileCache 256 × 256 images. The modified TileCache generates 1024 × 1024 images. The area covered by the modified TileCache image is 16 times that of a classical TileCache-generated image. In the images produced by the proposed TileCache, CNN models show superior performance in road detection because it is more clearly evident whether the image is a road or not.

5.2.3 Path hole detection achievement of SDPH techniques

Within the scope of this experiment, the damaged path detection achievement of SDPH techniques from huge volume and high-resolution satellite images was investigated. The proposed SDPH approach uses a two-stage deep learning methods to detect defects on the roads. In the images produced by Geoserver-TileCache, it is first determined whether there is a path or not. As shown in Fig. 2, if a path is detected in the image, it is checked with the second deep learning method to see any corruption in the path. In the experiments in Sect. 5.2.2, CNN models using modified TileCache outperformed classical CNN models. For this reason, modified CNN models were examined in this experiment. In the recommended SDPH technique, the input images to enter the second deep learning method should be detected as TP by the first deep learning method. As presented in Table 1, since the modified YOLOv5 model has the highest F1 score value in the first deep learning method, 18,213 images detected as TP by the modified YOLOv5 model are presented as input to the second deep learning method. The second deep learning technique trained for damaged roads is modified because it uses modified TileCache data. CNN models, which are the second deep learning technique and trained to detect broken paths, were tested on 18,213 images presented by the first deep learning method, and the achievement metrics are listed in Table 2.



(a) Input 1 (b) Input 2 (c) Input 3 (d) Input 4

(e) Result 1 by RCNNv3 (f) Result 2 by RCNNv3 (g) Result 3 by RCNNv3 (h) Result 4 by RCNNv3

(i) Result 1 by YOLOv5 (j) Result 2 by YOLOv5 (k) Result 3 by YOLOv5 (l) Result 4 by YOLOv5

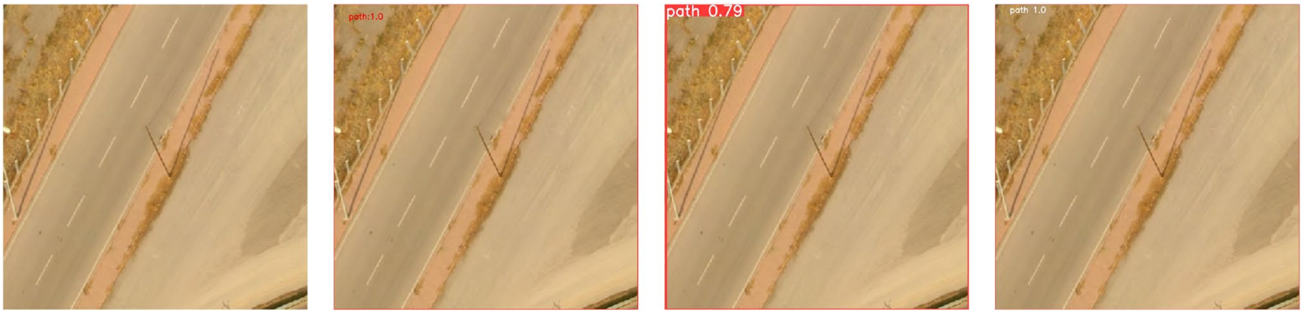
(m) Result 1 by YOLOv8 (n) Result 2 by YOLOv8 (o) Result 3 by YOLOv8 (p) Result 4 by YOLOv8

Fig. 5 Detection result of paths in size 256 × 256

When analyzing the metrics provided in Table 2, modified RCNNv3 and YOLOv8 achieved 0.998 accuracies, while modified YOLOv5 obtained 0.999 accuracies. The accuracy values are so high due to the low number of damaged roads in the data set. In other words, there is no damaged path object in 17,713 data in 18,213 data sets. Therefore, the TN value of deep learning methods is high. Modified RCNNv3, YOLOv5 and YOLOv8 models detected 463, 417 and 438 damaged paths as TP, respectively. In other words, these methods found the defects on the roads that were broken. The Fined-tuned RCNNv3 model catches the damage on the roads better than the modified YOLOv5 and YOLOv8 models. However, the FP of the modified RCNNv3 model is higher than that of the modified YOLOv5 and YOLOv8. The

RCNNv3 also detects some points on non-damaged roads as damaged. In order to better demonstrate the path hole detection achievement of the techniques, results on two different input images are presented in Fig. 7.

As seen in Fig. 7, a 1024 × 1024 input image produced by modified TileCache was presented to CNN models. The modified RCNNv3 model captured the defects found in Fig. 7c and d. However, because RCNNv3 is region-based and searches for the region within the image, it has determined the non-distorted region as a disorder with a confidence score of 1.0, as it is similar to a path disorder, as seen in Fig. 7c. Due to such determinations, the FP value of the



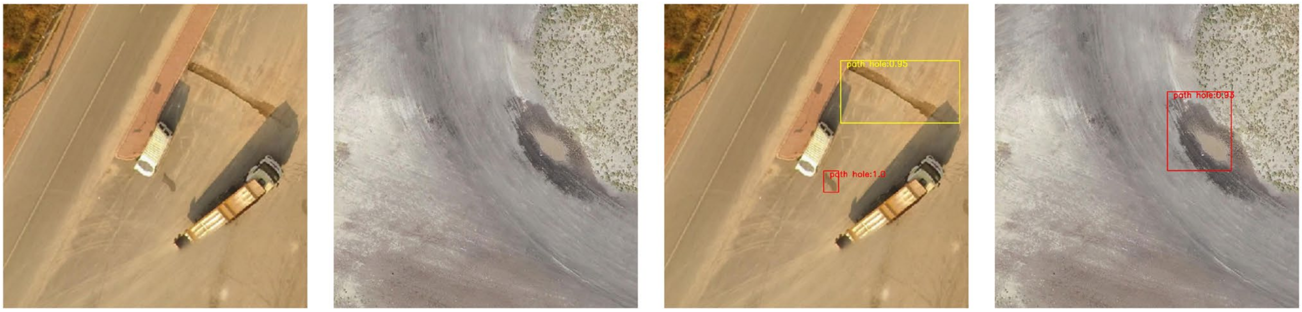
(a) Input (b) Result by RCNNv3 (c) Result 1 by YOLOv5 (d) Result 1 by YOLOv8

Fig. 6 Detection result of paths in size 1024 × 1024

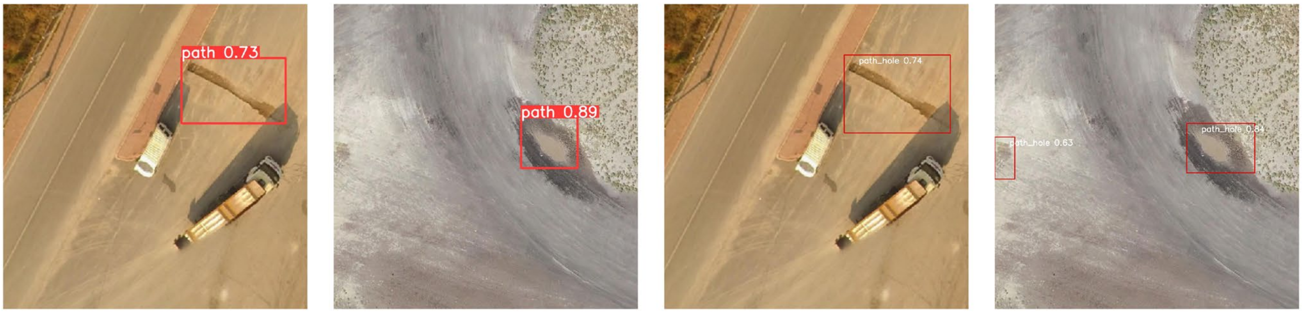
Table 2 Metric results of all techniques for path hole detection

Technique	TP	TN	FP	FN	Acc.	Pre.	Rec.	F1 S.
Modified RCNNv3	**463**	17,676	37	**5**	0.998	0.926	**0.989**	0.957
Modified YOLOv5	417	**17,701**	**12**	13	**0.999**	**0.972**	0.970	**0.971**
Modified YOLOv8	438	17,688	25	8	0.998	0.946	0.982	0.964

Bold text indicates that the relevant data are statistically superior



(a) Input 1 for path hole (b) Input 2 for path hole (c) Result 1 by RCNNv3 (d) Result 2 by RCNNv3



(e) Result 1 by YOLOv5 (f) Result 2 by YOLOv5 (g) Result 1 by YOLOv8 (h) Result 2 by YOLOv8

Fig. 7 Detection result of path holes in size 1024 × 1024

modified RCNNv3 model is high. On the other hand, the Fined-tuned YOLOv5 model caught path corruption in both images as seen in Fig. 7e and f. However, the FP value is low since the YOLOv5 model does not catch a point other than the disorder. The modified YOLOv8 model captured the defects found in images as seen in Fig. 7g and h. But, The modified YOLOv8 has determined the non-distorted region as a defect in disorder 7h. Due to such determinations, the FP value of the modified YOLOv8 model is high.

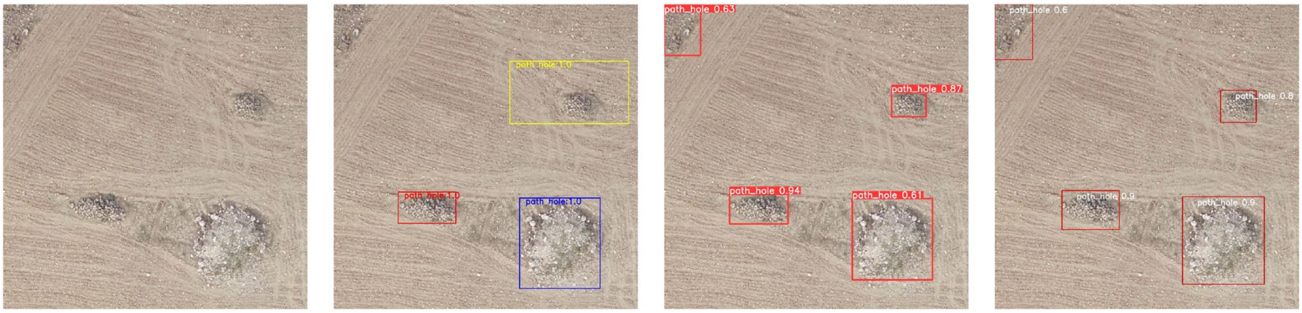
The recommended SDPH technique for detecting road disturbances with their positions from huge volume and high-resolution satellite images includes a two-stage process. First, with the CNN model recommended for path detection, it is determined whether the image has a road. Then, if the image is the path, it is determined whether there is a disorder. If the recommended SDPH technique is single-stage rather than two-stage, in other words, if the corrupted path is directly detected on the image produced by GeoServer-TileCache, the results as presented in Fig. 8 appear.

There are stones in the image shown in Fig. 8a and presented as input to the deep learning method. As shown in Fig. 8b–d, modified RCNNv3, YOLOv5 and YOLOv8 models detect these disturbances in the field as road disturbances. However, this disorder is not a road disorder. It can be expressed as the land disorder on the land plot. To overcome such problems, the SDPH technique has been built in two stages. The first CNN-based deep learning method checks for paths in the image. Then it is determined whether there is a disorder or not. The image presented in Fig. 8a does not pass through the second stage because it is detected that there is no path by the first deep learning method. From this perspective, the proposed SDPH technique performs better in detecting road disturbances along with the spatial positions of huge volume and high-resolution satellite images.

5.2.4 Runtimes of techniques

One of the important criteria in analysis studies on huge volumes of data is the computational cost. For this reason, the computational cost of path detection and path disturbance detection of the SDPH techniques recommended within the scope of this experiment has been examined. Two deep learning methods are used in the recommended SDPH technique. The first deep learning method determines whether there is a path in the input image. If the input image contains the path, it is determined whether there is a disorder with the second deep learning method. On the other hand, TilaCache was modified to improve the path detection and path disturbance detection achievement of the recommended SDPH technique. This process produces images as 1024 × 1024 with modified TilCache. When the images produced by classic TileCache are presented as input to the deep learning method, a total of 312,352 images are analyzed by deep learning methods. But, modified TileCache produces images of the same region; there are 19,522 images. In other words, the number of images produced by modified TileCache is 1 in 16 of the number of images produced by classic TileCache. This is because the 1024 × 1024 image contains 16 images of 256 × 256 images. In the recommended SDPH technique, CNN models are run separately on classical TileCache and modified TileCache. Run time cost in seconds of deep learning methods is presented in Table 3.

As can be inferred from Table 3, classic RCNNv3 detects whether the TileCache-generated images are paths in a total of 14,368 s, while the recommended modified RCNNv3 technique is 957 detects seconds. The recommended modified RCNNv3 technique makes path detection 15 times faster from huge volume satellite images than the classical RCNNv3 technique. On the other hand, the classical YOLOv5 model analyzes for 4685 s whether the images in the dataset are paths, while the modified YOLOv5 technique analyzes for only 312 s. Moreover, the classical YOLOv8 model analyzes for 4169 s whether the images in the dataset are paths, while the modified YOLOv8 technique analyzes for only 271 s. Thanks to the modified process, the path detection times of the modified YOLOv5 and YOLOv8 are also 15 times faster than their classical models. When the path detection times of CNN models are examined in



(a) Input without path (b) Result by RCNNv3 (c) Result by YOLOv5 (d) Result by YOLOv8

Fig. 8 Detection result of path holes in images that do not contain path

Table 3 Run times of all techniques

Method	Data set	Size	Image count	Run times (second)
Classic RCNNv3	Path	256 × 256	312,352	14368.19
Modified RCNNv3	Path	1024 × 1024	19,522	**956.58**
Classic YOLOv5	Path	256 × 256	312,352	4685.28
Modified YOLOv5	Path	1024 × 1024	19,522	**312.352**
Classic YOLOv8	Path	256 × 256	312,352	4169.90
Modified YOLOv8	Path	1024 × 1024	19,522	**271.75**
Modified RCNNv3	Path hole	1024 × 1024	18,213	899.45
Modified YOLOv5	Path hole	1024 × 1024	18,213	**296.41**
Modified YOLOv8	Path hole	1024 × 1024	18,213	**256.39**

Bold text indicates that the relevant data are statistically superior

seconds, the classic RCNNv3, classic YOLOv5, classic YOLOv8, modified RCNNv3, modified YOLOv5 and modified YOLOv8 techniques can analyze whether there is a path in an image of approximately 0.046, 0.015, 0.013, 0.049, 0.016 and 0.014 seconds, respectively. Modified CNN methods take a little longer to analyze an image than classical CNN methods. This is due to the large size of the image processed by modified models. However, when the whole working time is examined, modified models perform much faster analysis than classical models. This is because the number of images in the dataset is less.

Since modified models outperform classical methods in object detection and computational cost, only modified models were used to detect road defects. While the modified RCNNv3 model detects the disorder on the roads in 899 s, the modified YOLOv5 model detects it in 296 s and the modified YOLOv8 model detects it in 256 s. The modified YOLOv8 model detects road and road defects approximately 3.5 times faster than the modified RCNNv3 model.

5.2.5 The individual performance of deep learning-based object detectors in the proposed technique

Within the scope of this experiment, the individual performance of CNN models used for object detection in the SDPH approach, which is recommended for detecting road irregularities from large-volume satellite images, was examined. In the proposed SDPH approach, path detection is first performed. Later, if a road is detected in the image, it is determined whether there is any damage on the road. In the experimental investigations made in Sects. 5.2.3 and 5.2.4, the SDPH approach uses two different CNN models. In other words, while road detection is performed with the YOLOv5 model, road defect detection can be detected with the YOLOv8 model. However, in real-time systems, a single CNN model is usually used. For this reason, within the scope of this experiment, the performance of RCNNv3, YOLOv5, and YOLOv8 models in their singular use case was examined in the proposed SDPH approach. In other words, while road detection was done with the RNNv3 model, road defect detection was examined with the RCNNv3 model.

Deep learning models for road and road defect detection were trained with two different data sets. As shown in detail in Sect. 5.2.3, to successfully detect road defects from large volumes of satellite images, the input image must first be of road class. On the other hand, metrics such as micro average, macro average, and weighted average are used in the literature to determine the performance of the proposed method in deep learning-based techniques that perform different and multi-class detection [74, 75]. A micro average f1 score is preferred when each class cannot be ignored. The micro average f1 score may be a more appropriate measure of performance, especially if there is an uneven distribution between classes or some classes are less represented than others [76]. In the data set examined within the scope of this study, there is an unbalanced distribution because there are different numbers of road and road defects classes. In addition, classes cannot be ignored since road detection must be done first to detect road defects. For these reasons, the micro average f1 score was used to examine the individual performance of CNN models used as object detectors in the proposed SDPH approach. The micro average f1 score is calculated as shown in Eq. 2.

$$MicroF_1 - score = \frac{2 \times Micro\,Precision \times Micro\,Recall}{Micro\,Precision + Micro\,Recall} \tag{2}$$

While calculating the Micro Precision and Micro Recall values presented in Eq. 2, the total TP, TN, FP, and FN values in all classes are determined using Eq. 1.

In the proposed SDPH approach, the performance of the classical and modified RCNNv3, YOLOv5, and YOLOv8 models when used individually was examined in terms of micro f1-score and computational cost. The results obtained are presented in Table 4. In the proposed SDPH approach, in the first stage, it is examined whether the input image is in road class or not. If the first object detection model detects a road in the image (TP or FP), the

Table 4 Micro f1 score and run time performance of CNN Models

Object Detector	FPS	Processed image	Run times (minute)	Micro f1 score
Classic RCNNv3	**11**	545,469	836.39	0.752
Modified RCNNv3	10	**38,665**	**63.15**	**0.956**
Classic YOLOv5	**33**	502,978	251.49	0.726
Modified YOLOv5	31	**38,019**	**20.28**	**0.969**
Classic YOLOv8	**37**	505,339	224.88	0.720
Modified YOLOv8	36	**38,071**	**17.86**	**0.965**

Bold text indicates that the relevant data are statistically superior

object detection model in the second stage tries to detect whether there is any defect in the road. The number of images examined by the object detection method in the second stage is the total number of images detected as TP or FP in the first stage. As presented in Table 1, the number of images examined in Table 4 is different from each other because the models in the first stage of deep learning have different numbers of TP and FP values. On the other hand, frame per second (FPS), the term expressed as the number of frames that can be processed in one second, is an important metric used to evaluate the computational cost in real-time systems [77]. FPS affects response times, security, and quality of simulations in real-time systems and is therefore important.

As seen in the Table 4, the improved RCNNv3, YOLOv5, and YOLOv8 models showed a clear superiority in terms of the micro f1 score metric compared to the classic RCNNv3, YOLOv5, and YOLOv8 models. Improved RCNNv3, YOLOv5, and YOLOv8 models achieved micro f1 scores of 0.956, 0.969 and 0.965, respectively. Thanks to the improvements made on TileCache, deep learning models have achieved these high f1 scores. On the other hand, when models are examined according to computational cost, classical deep learning models have higher FPS values than improved models. Because, while the classical object detection method processes images in the size of 256×256, the improved object detection model processes images in the size of 1024×1024. However, when the overall computational cost is examined, the computational costs are very high for classical models because classical deep learning models process approximately 13 times more images than improved deep learning models. When the deep learning models are examined individually in the proposed SDPH approach, the improved RCNNv3, YOLOv5, and YOLOv8 models have a much superior f1 score metric and much lower computational cost than the classical models.

6 Conclusion

In this study or research, a hybrid method called SDPH is proposed to detect the corrupted paths of huge volume satellite images together with their geographical location. In the recommended method, the huge volume satellite image is first accessed as a pyramid grid file system using GeoServer and TileCache. Then, the images produced by TileCache are presented to the two-stage object detection model. The first deep learning technique detects whether there is a path in the image. The second deep learning technique detects whether there is a disorder if the image is a path. In the recommended SDPH technique, to increase the object detection achievement of the deep learning method, TileCache is modified to produce 1024×1024 images. Classic TileCache produces 256×256 images. Path detection achievement of RCNNv3, YOLOv5 and YOLOv8 deep learning methods on images produced by classical TileCache and modified TileCache were investigated. Classic RCNNv3, classic YOLOv5, classic YOLOv8, modified RCNNv3, modified YOLOv5 and modified YOLOv8 models scored 0.743, 0.716, 0.710, 0.955, 0.958 and 0.954 f1 in path detection, respectively. Since the modified YOLOv5 model has the highest f1 score value in path detection, the images detected by this model as a path are presented to the second deep learning method. In the second deep learning process, modified RCNNv3, modified YOLOv5, and modified YOLOv8 models were used to examine any distortion in the images detected as paths. In damaged road detection, the modified RCNN v3 scored a 0.957 f1 score, while the modified YOLOv5 and YOLOv8 were 0.971 and 0.964 f1 scores. Improved deep learning models show much superior performance than the classics. If the proposed SDPH approach uses the same CNN models for the first and second stages, the modified RCNNv3, YOLOv5, and YOLOv8 models obtain a micro f1 score of 0.9, 0.95, and 0.92, respectively. In the SDPH technique recommended to detect road disturbances from huge volume and high-resolution satellite images with their positions, the modified YOLOv5 model outperformed by detecting damaged roads with a 0.969 micro f1 score on approximately 0.032 s.

In the future, it is planned to carry out studies on detecting roads as polygons from huge volume and high-resolution satellite images.

Acknowledgements We express our gratitude to the Kayseri Metropolitan Municipality for generously providing large-scale, high-resolution satellite images. Furthermore, this research receives support from the Scientific and Technical Research Council of Turkey under the auspices of the Career Development Program (3501), Project No: 122E315.

Author Contributions All authors contributed equally.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). The Scientific and Technical Research Council of Turkey (122E315).

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Singh, H., Kushwaha, V., Agarwal, A.D., Sandhu, S.S.: Fatal road traffic accidents: causes and factors responsible. J. Indian Acad. Forensic Med. **38**(1), 52–54 (2016)
2. Srinivasu, P.N., Bhoi, A.K., Jhaveri, R.H., Reddy, G.T., Bilal, M.: Probabilistic deep q network for real-time path planning in censorious robotic procedures using force sensors. J. Real-Time Image Process. **18**(5), 1773–1785 (2021)
3. Munti, N.Y.S., Setiawan, B., et al.: Analysis of web-based geographic information system mapping of broken roads in Kampar regency. J. Eng. Sci. Technol. Manag. (JES-TM) **1**(1), 22–27 (2021)
4. Long, L., He, F., Liu, H.: The use of remote sensing satellite using deep learning in emergency monitoring of high-level landslides disaster in jinsha river. J. Supercomput. **77**, 8728–8744 (2021)
5. Li, X., Yirui, W., Zhang, W., Wang, R., Hou, F.: Deep learning methods in real-time image super-resolution: a survey. J. Real-Time Image Process. **17**, 1885–1909 (2020)
6. Chaudhary, V., Buttar, P.K., Sachan, M.K.: Satellite imagery analysis for road segmentation using u-net architecture. J. Supercomput. **78**(10), 12710–12725 (2022)
7. Huan, H., Zou, N., Zhang, Y., Xie, Y., Wang, C.: Remote sensing image reconstruction using an asymmetric multi-scale superresolution network. J. Supercomput. **78**(17), 18524–18550 (2022)
8. Anagnostopoulos, I., Zeadally, S., Exposito, E.: Handling big data: research challenges and future directions. J. Supercomput. **72**, 1494–1516 (2016)
9. Chen, C., Li, W., Gao, L., Li, H., Plaza, J.: Special issue on advances in real-time image processing for remote sensing. J. Real-Time Image Process. **15**, 435–438 (2018)
10. Tasyurek, M.: A novel approach to improve the performance of the database storing big data with time information. Balk. J. Electr. Comput. Eng. **10**(4), 388–396 (2022)
11. Ngo, T.D., Bui, T.T., Pham, T.M., Thai, H.T.B., Nguyen, G.L., Nguyen, V.: Image deconvolution for optical small satellite with deep learning and real-time gpu acceleration. J. Real-Time Image Process. **18**(5), 1697–1710 (2021)
12. Xu, H.: Arcgis data models for managing and procesing imagery (2012)
13. Gibin, M., Singleton, A., Milton, R., Mateos, P., Longley, P.: An exploratory cartographic visualisation of London through the google maps api. Appl. Spat. Anal. Policy **1**, 85–97 (2008)
14. Yin, F., Feng, M.: A webgis framework for vector geospatial data sharing based on open source projects. In: Proceedings. The 2009 International Symposium on Web Information Systems and Applications (WISA 2009), p. 124. Academy Publisher (2009)
15. Taşyürek, M.: Regenerating large volume vector layers with a denormalization-based method. In: 2021 6th International Conference on Computer Science and Engineering (UBMK). IEEE. pp. 124–128 (2021)
16. Guo, D., Zou, Y., Wang, S.: An effective tile caching mechanism of uav remote sensing map based on hilbert coding index. In: 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN). IEEE. pp. 535–541 (2019)
17. García, R., de Castro, J.P., Verdú, E., Verdú, M.J., Regueras, L.M.: Web map tile services for spatial data infrastructures: Management and optimization. In: Cartography-A Tool for Spatial Analysis, pp. 26–48 (2012)
18. Mahajan, S., Rajesh, M.A., Panigrahi, N.: Scale space visualization of dynamic track clusters in an enterprise gis. In: 2022 3rd International Conference for Emerging Technology (INCET). IEEE. pp. 1–6 (2022)
19. Khamparia, A., Gupta, D., de Albuquerque, V.H.C., Sangaiah, A.K., Jhaveri, R.H.: Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning. J. Supercomput. **76**, 8590–8608 (2020)
20. Qiu, L., Zhang, D., Tian, Y., Al-Nabhan, N.: Deep learning-based algorithm for vehicle detection in intelligent transportation systems. J. Supercomput. **77**, 11083–11098 (2021)
21. Toshpulatov, M., Lee, W., Lee, S., Haghighian Roudsari, A.: Human pose, hand and mesh estimation using deep learning: A survey. J. Supercomput. **78**(6), 7616–7654 (2022)
22. Tasyurek, M.: Odrp: a new approach for spatial street sign detection from EXIF using deep learning-based object detection, distance estimation, rotation and projection system. Vis. Comput. **40**(2), 983–1003 (2024)
23. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H.: Road damage detection and classification using deep neural networks with smartphone images. Comput.-Aided Civ. Infrastruct. Eng. **33**(12), 1127–1141 (2018)
24. Rath, S.: Fine tuning yolov7 on custom dataset (2022)
25. Parvathavarthini, S., Shreekanth, M., Vigneshkumar, S., Santhos, NS.: Road damage detection using deep learning. In: 2023 7th International Conference on Computing Methodologies and Communication (ICCMC). IEEE. pp. 314–318 (2023)
26. Öztürk, C., Taşyürek, M., Türkdamar, M.U.: Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the cnn-based deep learning models. Concurr. Comput. Pract. Exp. **35**(4), e7542 (2023)
27. Tasyurek, M., Arslan, R.S.: Rt-droid: a novel approach for real-time android application analysis with transfer learning-based cnn models. J Real-Time Image Process **20**(3), 1–17 (2023)
28. Talaat, F.M., ZainEldin, H.: An improved fire detection approach based on yolo-v8 for smart cities. Neural Comput. Appl. **35**(28), 20939–20954 (2023)
29. Terven, J., Córdova-Esparza, D.-M., Romero-González, J.-A.: A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Mach. Learn. Knowl. Extr. **5**(4), 1680–1716 (2023)
30. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J.: A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans. Neural Netw. Learn. Syst. **33**(12), 6999–7019 (2021)

31. Kehe, W., Chen, Z., Li, W.: A novel intrusion detection model for a massive network using convolutional neural networks. Ieee Access **6**, 50850–50859 (2018)
32. Nanni, L., Manfè, A., Maguolo, G., Lumini, A., Brahnam, S.: High performing ensemble of convolutional neural networks for insect pest image detection. Ecol. Inform. **67**, 101515 (2022)
33. Jadhav, S.B., Udupi, V.R., Patil, S.B.: Identification of plant diseases using convolutional neural networks. Int. J. Inf. Technol. **13**(6), 2461–2470 (2021)
34. Koller, O., Zargaran, O., Ney, H., Bowden, R.: Deep sign: Hybrid cnn-hmm for continuous sign language recognition. In: Proceedings of the British Machine Vision Conference 2016 (2016)
35. Ciocca, G., Napoletano, P., Schettini, R.: Cnn-based features for retrieval and classification of food images. Comput. Vis. Image Underst. **176**, 70–77 (2018)
36. Achour, B., Belkadi, M., Filali, I., Laghrouche, M., Lahdir, M.: Image analysis for individual identification and feeding behaviour monitoring of dairy cows based on convolutional neural networks (cnn). Biosyst. Eng. **198**, 31–49 (2020)
37. Bermejo-Peláez, D., Ash, S.Y., Washko, G.R., Estépar, R.S.J., Ledesma-Carbayo, M.J.: Classification of interstitial lung abnormality patterns with an ensemble of deep convolutional neural networks. Sci. Rep. **10**(1), 1–15 (2020)
38. Rao, A., Motwani, R., Sarguroh, N., Kingrani, P., Khandaskar, S.: Real-time traffic sign recognition using convolutional neural networks. In: Emerg. Technol. Data Min. Inf. Secur., pp. 223–234. Springer, NY (2021)
39. Yang, W.-J., Luo, C.-C., Chung, P.-C., Yang, J.-F.: Simplified neural networks with smart detection for road traffic sign recognition. In: Future of Information and Communication Conference. Springer. pp. 237–249 (2019)
40. Cao, J., Song, C., Peng, S., Xiao, F., Song, S.: Improved traffic sign detection and recognition algorithm for intelligent vehicles. Sensors **19**(18), 4021 (2019)
41. LeCun, Y., et al.: Lenet-5, convolutional neural networks. URL: http://yann. lecun. com/exdb/lenet 20(5):14 (2015)
42. Sultana, F., Sufian, A., Dutta, P.: Advancements in image classification using convolutional neural network. In: 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). IEEE. pp. 122–129 (2018)
43. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7310–7311 (2017)
44. Krizhevsky, A., Sutskever, I., Hinton, GE.: 2012 alexnet (2012)
45. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
46. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
47. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
48. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017)
49. Girshick, R., Donahue, J., Darrell, T., Malik, J.: 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 (2014)
50. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448 (2015)
51. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788 (2016)
52. Kawauchi, H., Fuse, T.: Shap-based interpretable object detection method for satellite imagery. Remote Sens. **14**(9), 1970 (2022)
53. Zhi-Ze, W., Wang, X.-F., Zou, L., Li-Xiang, X., Li, X.-L., Weise, T.: Hierarchical object detection for very high-resolution satellite images. Appl. Soft Comput. **113**, 107885 (2021)
54. Song, Z., Sui, H., Hua, L.: A hierarchical object detection method in large-scale optical remote sensing satellite imagery using saliency detection and cnn. Int. J. Remote Sens. **42**(8), 2827–2847 (2021)
55. Gong, H., Tingkui, M., Li, Q., Dai, H., Li, C., He, Z., Wang, W., Han, F., Tuniyazi, A., Li, H., et al.: Swin-transformer-enabled yolov5 with attention mechanism for small object detection on satellite images. Remote Sens. **14**(12), 2861 (2022)
56. Khan, S.D., Alarabi, L., Basalamah, S.: A unified deep learning framework of multi-scale detectors for geo-spatial object detection in high-resolution satellite images. Arab. J. Sci. Eng. **47**(8), 9489–504 (2021)
57. Johanson, M., Belenki, S., Jalminger, J., Fant, M., Gjertz, M.: Big automotive data: Leveraging large volumes of data for knowledge-driven product development. In: 2014 IEEE international conference on big data (Big Data). IEEE. pp. 736–741 (2014)
58. Zhang, C., Marzougui, A., Sankaran, S.: High-resolution satellite imagery applications in crop phenotyping: an overview. Comput. Electron. Agric. **175**, 105584 (2020)
59. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
60. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv: 1804.02767 (2018)
61. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv: 2004.10934 (2020)
62. Jocher, G., Nishimura, K., Mineeva, T., Vilariño, R.: Yolov5 (2020)
63. Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B.: A review of yolo algorithm developments. Procedia Comput. Sci. **199**, 1066–1073 (2022)
64. Fang, Y., Guo, X., Chen, K., Zhou, Z., Ye, Q.: Accurate and automated detection of surface knots on sawn timbers using yolo-v5 model. BioResources **16**(3), 5390 (2021)
65. Munawar, R., Jocher, G.: Ultralytics yolov8 (2023)
66. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28**, 91–99 (2015)
67. Youngblood, B.: GeoServer Beginner's Guide. Packt Publishing Ltd (2013)
68. MetaCarta. Tilecache-web map tile caching (2010)
69. Tasyurek, M., Celik, M.: Fastgtwr: a fast geographically and temporally weighted regression approach. J. Fac. Eng. Archit. Gazi Univ. **36**(2), 715–726 (2021)
70. Tasyurek, M., Celik, M.: 4d-gwr: geographically, altitudinal, and temporally weighted regression. Neural Comput. Appl. **34**(17), 14777–14791 (2022)
71. Stackexchange. Calculate lat lon bounds for individual tile generated from gdal2tiles (2012)
72. Agafonkin, V.: Leaflet (2011)
73. Bennett, J.: OpenStreetMap. Packt Publishing Ltd (2010)
74. Zhang, D., Wang, J., Zhao, X.: Estimating the uncertainty of average f1 scores. In: Proceedings of the 2015 International

conference on the theory of information retrieval. pp. 317–320 (2015)

75. Zhou, G., Yuan, S., Xing, H., Jiang, Y., Geng, P., Cao, Y., Ben, X.: Micro-expression action unit recognition based on dynamic image and spatial pyramid. J. Supercomput. **79**(17), 19879–902 (2023)
76. Nath, R.K., Thapliyal, H., Caban-Holt, A.: Machine learning based stress monitoring in older adults using wearable sensors and cortisol as stress biomarker. J. Signal Process. Syst. **94**, 513–525 (2021)
77. Ahmad, W., Mahdavi, H., Hamzaoglu, I.: An efficient versatile video coding motion estimation hardware. J. Real-Time Image Process. **21**(2), 25 (2024)

Murat Tasyurek was born in Turkey in 1986. He received the B.S. degree, the M.Sc. degree, and the Ph.D degree in computer engineering from Erciyes University in 2009, 2011 and 2020, respectively. He worked as a software developer computer engineer for the Kayseri metropolitan municipality between 2010 and 2021. He has been working as an assistant prof in the computer engineering department of Kayseri University since 2021. His research interests include image registration, data mining, GIS (Geographic Information Systems) analysis, big data, deep learning, and optimization algorithms.