Decision Analytics Journal 12 (2024) 100507

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A deep learning model to assist visually impaired in pothole detection using computer vision

Arjun Paramarthalingam [a], Jegan Sivaraman [a], Prasannavenkatesan Theerthagiri [b,*], Balaji Vijayakumar [a], Vignesh Baskaran [a]

[a] Department of CSE, University College of Engineering Villupuram, Villupuram, India
[b] Department of Computer Science and Engineering, GITAM School of Technology, GITAM Deemed to be University, Bengaluru, India

ARTICLE INFO

Keywords:
Deep learning
Computer vision
Pothole detection
You Only Look Once (YOLO) algorithm
Visually impaired

ABSTRACT

Visually impaired individuals encounter numerous impediments when traveling, such as navigating unfamiliar routes, accessing information, and transportation, which can limit their mobility and restrict their access to opportunities. However, assistive technologies and infrastructure solutions such as tactile paving, audio cues, voice announcements, and smartphone applications have been developed to mitigate these challenges. Visually impaired individuals also face difficulties when encountering potholes while traveling. Potholes can pose a significant safety hazard, as they can cause individuals to trip and fall, potentially leading to injury. For visually impaired individuals, identifying and avoiding potholes can be particularly challenging. The solutions ensure that all individuals can travel safely and independently, regardless of their visual abilities. An innovative approach that leverages the You Only Look Once (YOLO) algorithm to detect potholes and provide auditory or haptic feedback to visually impaired individuals has been proposed in this paper. The dataset of pothole images was trained and integrated into an application for detecting potholes in real-time image data using a camera. The app provides feedback to the user, allowing them to navigate potholes and increasing their mobility and safety. This approach highlights the potential of YOLO for pothole detection and provides a valuable tool for visually impaired individuals. According to the testing, the model achieved 82.7% image accuracy and 30 Frames Per Second (FPS) accuracy in live video. The model is trained to detect potholes close to the user, but it may be hard to detect potholes far away from the user. The current model is only trained to detect potholes, but visually impaired people face other challenges. The proposed technology is a portable option for visually impaired people.

1. Introduction

Humans see the images and instantly identify what it is, where it is, and how they relate. The human vision system is extremely fast and accurate, allow us to perform complicated works like driving. The good object recognition systems that involves identifying target objects in digital images and videos with high speed and precision [1]. The YOLO (You Only Look Once) is an object detection system that can be widely used in real time applications due to its ability to detect multiple object classes using a single neural network [2–6]. This system has been used in various applications such as automated cars and surveillance systems, industrial inspection systems, underwater object detection etc. [7].

Visually impaired individuals face numerous challenges while traveling, including navigating obstacles such as potholes. A pothole is a hollow area on the road where cracks have appeared and eventually formed a round pit that is more than 15 cm in diameter [8]. Potholes are dangerous for drivers and pedestrians especially visually impaired [9]. Potholes are particularly problematic as they can cause physical harm and damage to mobility devices, making it difficult for the visually impaired to maintain their balance and increasing the risk of falls. Potholes detection is classified into 3 types viz. vibration-based, 2D vision-based, and 3D re-construction-based methods [10]. Three key visual attributes of potholes, namely their round shape, darker color compared to their surroundings, and rough texture relative to the surrounding area, are utilized in computer vision-based techniques to identify and isolate the pothole's shape within an image. These features are then compared with the surrounding area to identify the pothole region. Image processing techniques are employed to detect defect regions by analyzing the texture, shape, and size of the damaged area [7,11–13]. In this paper, we developed a pothole detection approach that utilizes object detection technology such as YOLOv5 can help address this problem. Once a pothole is detected, the application

* Corresponding author.
E-mail address: prasannait91@gmail.com (P. Theerthagiri).

https://doi.org/10.1016/j.dajour.2024.100507
Received 27 December 2023; Received in revised form 9 July 2024; Accepted 1 August 2024
Available online 5 August 2024

provides an auditory or haptic notification to alert the user. This method utilizes YOLO algorithm to detect and locate potholes within an image captured by a camera. This innovative solution can improve the safety and accessibility of visually impaired individuals while traveling on the road.

Traditional methods such as edge detection and template matching are not effective in detecting potholes due to their irregular shapes and sizes [14]. Many authors stated and proved that Deep Neural Network (DNN) models give better pothole detection accuracy as compared with traditional machine learning algorithms [10,15–20]. In [18], Gajjar et al. used faster Region-based Convolutional Neural Network (RCNN), Single Shot Detector (SSD), and YOLOv3 models for real time pothole detection task, the YOLOv3 offers superior results and outperforms the other two models in real-time pothole detection. Chitale et al. [19] developed a pothole detection system designed to reduce human dependency in road maintenance and provide reasonably accurate results for both pothole detection and dimension estimation. Silvister et al. [20] used YOLO implementation in mobile application for real-time pothole detection with high accuracy in identifying objects within an image.

Similarly, the proposed system of pothole detection for visually impaired individuals uses YOLOv5, which involves splitting an image into a grids and making predictions on bounding boxes and class probabilities for each cell. The YOLOv5 algorithm is trained on extensive image datasets to develop the ability to detect objects of interest more accurately, like potholes in this case. By providing users with real-time information about potholes, the application can help them avoid accidents and prevents damage to their mobility devices, reducing travel time and increasing independence. In summary, the pothole detection application for visually impaired using YOLO is a practical and innovative solution that demonstrates the potential of object detection technology to create solutions that can improve people's lives. The approach incorporates various notification methods and employs the YOLO algorithm to effectively detect and pinpoint potholes in real-time, offering user's crucial information for safe and autonomous travel. The contributions and highlights of this work are follows:

- Proposed a You Only Look Once (YOLO) algorithm to detect potholes and provide auditory or haptic feedback to visually impaired individuals.
- Used the dataset of pothole images, which are trained and integrated into an application for detecting potholes in real-time image data using a camera.
- Achieved 82.7% image accuracy and 30 frames per second (FPS) accuracy in live video.
- Trained the model to detect potholes nearby or far away from the user.
- Introduced a portable option for visually impaired people.

2. Literature survey

Object detection is a method in the field of computer vision that enables the identification and localization of objects within an image or video. The development of object detection models is an ongoing process aimed at improving their accuracy and efficiency. Joseph Redmon et al. [2] coined the idea of YOLO (You Only Look Once) algorithm, which is a single-pass object detection technique. This algorithm is capable of detecting objects in an image through a single pass of the whole image through neural networks. It outperformed other object detection algorithms and achieved state-of-the-art results. Prior algorithms, such as RCNN, required multiple iterations, whereas YOLO is a single-iteration process. The YOLO algorithm employs CNN (Convolutional Neural Network) to identify objects within an image, dividing the input image into S × S grids, with each grid responsible for detecting an object if its center lies within that grid.

In YOLO9000 (YOLOv2) Joseph Redmon et al. [3] have enhancing the basic YOLO algorithm by improving its accuracy, speed, and ability to detect a wider range of classes. It uses Darknet-19, a variant of the VGGNet and anchor box to determine the final bounding box. The algorithm also incorporates a novel loss function that calculates the sum of squared errors between the predicted bounding boxes and the ground truth bounding boxes. Joseph Redmon et al. [4] again improved the YOLO algorithm by further increasing the speed and accuracy with the help of new Convolutional Neural Network (CNN) called Darknet-53, specially designed for object detection by modifying the ResNet architecture. The YOLOv3 algorithm utilizes anchor boxes of varying scales and aspect ratios, in contrast to its predecessor, which used anchor boxes of identical sizes. Consequently, YOLOv3 can detect objects with different sizes and shapes, whereas the previous version cannot. Later Alexey Bochkovskiy et al. [5] enhanced YOLOv3 by introducing YOLOv4 using the CSPNet (Cross Stage Partial Network), a new Convolutional Neural Network that is a ResNet variant. They also implemented a new approach for generating anchor boxes, called K-means clustering algorithm.

The YOLO algorithm used in this paper is YOLOv5 released by Ultralytics, which employs the more intricate EfficientDet architecture, based on EfficientNet for object detection task. This architecture facilitates improved accuracy and enables the usage of a wide variety of object classes. Furthermore, it employs a new technique for generating anchor boxes, termed Dynamic anchor boxes. The YOLO algorithm is used by many authors in the literature for pothole detection task [6,8,9]. Md. Milon Islam et al. [8] proposed a pothole detection system to assist the visually impaired people. They developed the system using the Convolutional Neural Network (CNN) and KITTI road and pothole detection datasets are used for training the model. But in practical case, CNN is not preferable for real-time object detection compared to YOLO due to speed constraints.

Rajshekar Hiremath et al. [9] developed a smart app to detect potholes using YOLO by implementing this pothole detection model in terms of mobile apps will be portable. Every user needs to create an account to take pictures and upload them for pothole detection. Then the app will predict the pothole and store the location of the pothole in the database and share the location of the pothole to other users by using Google Map API (Application Programming Interface). Later Kshitija Chavan et al. [6] developed a pothole detection system using the YOLOv4 algorithm. It has a simple Graphical User Interface (GUI) with two options: start and stop. This system requires a laptop or pc with a webcam to take input images from the camera and predict whether the pothole is there or not and alert the user to ensure road safety. This approach uses the same pre-trained model trained using MS COCO (Microsoft Common Objects in Context) dataset and do not train by using a custom dataset. A comparative analysis of various approaches of object detection interims of detecting potholes is given in Table 1.

A novel crowdsourced approach was proposed by Yangsong Gu et al. [21] to detect potholes in road surfaces and assist in pavement distress diagnosis. They predicted the occurrence of potholes with respect to time and space parameters using spatio-temporal kernel density estimation model. This approach will be helpful to road repair agencies for constant maintenance of the road infrastructure to improve efficiency in road transportation. A study by Eva Lieskovská et al. [22] performed automatic pothole detection task using deep learning algorithms viz. R-CNN and Yolov3, 5, 7 models. They collected road images from Slovak region to test the performance of these algorithms, here Yolo models performed well than R-CNN algorithm.

P C Nissimagoudar et al. [23] performed a study to detect potholes and speed breakers in the road early to avoid inconveniences and accidents. They used Yolov5 algorithm to accomplish their task, and claimed their approach can be useful for designing suspension systems in vehicles to ensure comfortable journey. A benchmark video dataset for pothole detection problems were created by Muhammad Ihsan et al. [24]. They formed their dataset with 619 HD road videos collected from South Kalimantan, Indonesia. According to authors claim, their

Table 1
Comparative study of existing literature for pothole detection for visually impaired using different versions of YOLO algorithm.

Ref.	Method	Dataset	Merits	Limitations
[2]	YOLO	PASCAL VOC, ImageNet 1000-class	High speed compared to RCNN and can be used for real time object detection.	Not specific for visually impaired people and potholes.
[3]	YOLO9000 (YOLOv2)	ImageNet and MS COCO dataset using Word Tree	Compared to YOLO, it can detect a broader range of object classes, and it achieves high accuracy and speed.	
[4]	YOLOv3	MS COCO dataset	Able to detect objects of different shape and size.	
[5]	YOLOv4	ImageNet and MS COCO dataset	High speed and accuracy compared to previous versions.	
[6]	YOLOv4	MS COCO dataset	It is specifically for pothole detection.	It is implemented on a laptop or pc which is not portable.
[8]	CNN	KITTI Road and Pothole detection dataset	It is specifically for the visually impaired to detect potholes.	It uses CNN. It is not preferable for real time object detection compared to YOLO.
[9]	YOLO	Kaggle pothole detection dataset	It is portable and used in mobile phones	It requires internet. Not suitable for visually impaired

dataset will be useful for researchers, policy makers to experiment pothole detection algorithm before applying in real use-cases.

A pre-trained visual pollution detection system was developed by Md. Yearat Hossain et al. [25] to identify environment pollutants. The Yolov5, FR-CNN (Fast R-CNN), and EfficientNet models were tested, here the Yolov5 model performs better than other two models. In this paper, authors developed an Android application for capturing and communicating the environment pollutant images for further analysis of visual pollutants. Transportation through road is dominant one, where more than sixty percentage of the people in the world are using road for transportation and commutation. In road transportation, potholes are the major challenging factor, which affects efficiency of roadways. M. Sathvik et al. [26] performed pothole detection using YOLOv7 model. They used recall and accuracy as performance assessment metrics for their proposed model.

A Cooperative driving scheme by G. Arvanitis et al. [27] performed pothole detection using computer vision and augmented reality visualization. Further, they share this information with other connected vehicle on the road to create the awareness of road conditions. The pothole detection is very important in navigating autonomous vehicles. G. Raja et al. [28], developed a hybrid model that combines voice and gesture information of driver to fine-tune their deep learning model (deep deterministic policy gradient). Their system considers lane changing parameters like speed, time, angle, etc to predict the potholes in real-time videos. The YOLOv5 based economic pothole detection system was developed by X. Chen et al. [29].

The image sensors and Light Detection and Ranging (LiDAR) sensors are used to capture input data for pothole detection. The robustness of Yolo-seg model was tested on GRDDC2020 dataset which consists of potholes of varying sizes. The multi-scale potholes detection using deep learning with transformers are used by Bibi et al. [30]. Their approach ensures robustness by handling potholes of varying sizes and aspect ratio in more effective way with different road conditions. Vinodhini et al. [31] developed a CNN based pothole detection approach for bituminous roads. Their approach designed an alert system about potholes on the road to make drivers more cautious and helpful for speedy road maintenance work.

Chougule [32] experimented their pothole detection mechanism in Raspberry kit with Kaggle pothole dataset using YOLOv3 and YOLOv5 models. From their experimental observation, YOLOv5 produced good precision recall results than YOLOv3 model. Singh et al. [33] developed a pothole detection system, which works well even on poor lighting conditions. Their approach used Improved Long Short-Term Memory (LSTM) model with Local Binary Patterns (LBP) texture feature for classification of potholes on the input data captured using accelerometer and gyroscope in mobile phones. The pothole size, scale, extend of the distance, speed and varying illumination conditions are making the pothole detection task more challenging. The CNN network is used in this paper for pothole detection task [34].

The accelerometers deployed in the smartphones are used to collect input data for pothole detection task [35], useful for pavement managers to efficiently handle road maintenance work. Ha et al. [36] performed a study to find different types of cracks on roads. The deep learning models viz. Squeeze-Net, U-Net, and Mobile-Net are considered for crack detection and severity measurement. It categorized the following types of cracks (vertical crack, horizontal crack, alligator crack, longitudinal crack, transverse crack, pothole, and patches) for efficient handling in field use. The road damage detection method in [37] used MN-YOLOv5 algorithm. It is an integrated model developed by basic YOLOv5 with MobileNetV3 model. The K-Means clustering, and label smoothing are used in the preprocessing stage for increased detection speed and efficiency. The potholes and small holes found in the road are detected in [38]. They experimented their methods on Nvidia Jetson Nano kit embedded on the vehicle with YOLOv7 model for detection purpose.

Maroš Jakubec et al. [39] performed an elaborated survey on pothole detection approaches with respect to usage computer vision and deep learning algorithms. From their study, they concluded that RCNN and YOLO algorithms perform well on pothole detection task. The major visual challenges in pothole detection are, climate, weather, rain, day–night visibility, timing, environmental clean conditions, illumination of the scene etc. A scalable pothole detection approach in [40] used CNN models working on smartphones to identify manholes, potholes and speed breakers in the road surfaces. Unlike the approaches using computer vision for pothole detection, the novel approach by Furkan Ozoglu [41] used vibration sensors, accelerometer and gyroscope for acquiring road surface information. These data are then converted into pixel format to process using CNN's.

An integrated framework by Abdullah et al. [42] used basic YOLOv5 with ECA-Net, K-means and focal loss for improved performance in road damage detection. They tested their approach on RDD 2022 dataset for detection of vertical–horizontal cracks and potholes. The light-weight version of Yolo Model (Yolov3) is used by Boris Bučko et al. [43], which requires minimum hardware specification to accomplish pothole detection in challenging weather and lighting conditions. In contrast with pothole detection in 2D input data, Bosurgi et al. [44] performs accurate detection using 3D input data of road surfaces. Their approach considers depth, area, perimeter of potholes for measuring the severity and shape of the damage. Muhammad Haroon Asad et al. [45] implemented different versions of YOLO algorithms on edge devices (Raspberry Pi with OAK-D) for real-time detection of potholes. From their experimental study, YOLOv5 produced higher accuracy than other models and it is more robust to different road illumination conditions.

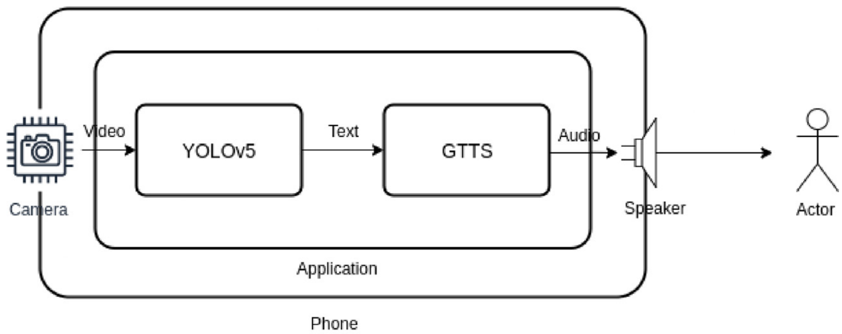

Fig. 1. System architecture diagram of the mobile application.

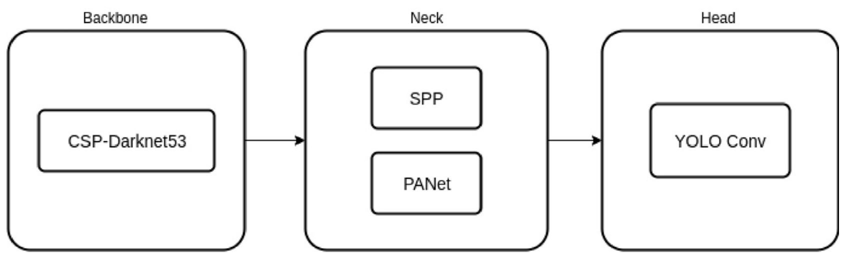

Fig. 2. Major components of YOLOv5 architecture.

The K-Means and label smoothing are used as preprocessing techniques for improved detection results [46]. Min-Li Lan et al. [47] used YOLOv5v model for automatic cracks detection for timely maintenance of roads. Rainy weather is one of the important reason for formation of potholes, hence automatic detection systems are required for regular maintenance of roads [48]. Their approach collects location information about potholes and share it to road maintenance authorities. Fernandes et al. [49] performed cracks detection on roads using supervised machine learning techniques and texture features. They experimented their method on road images taken from Brazilian road highways. A comprehensive survey by Chen et al. [50] considers various factors such as pothole input data acquisition methods, different detection algorithms, datasets, and different road surface conditions for broad analysis of existing pothole detection methods by different authors. The road pothole detection method by Srivani et al. [51] used CNN model with InseptionResNetV2 and DenseNet201 architectures to classify pothole and non-pothole roads. Survey on road pothole detection systems [52]. The pothole detection system by Cahya Buana et al. [53] used CNN with MobileNet SSDv2 architecture to find road damages.

3. Proposed methodology

We proposed the methodology that utilizes a mobile application connected to the Google Text-to-Speech (GTTS) engine to detect potholes on roads. The application is developed using the YOLOv5 algorithm and is specifically designed for visually impaired users, making it a portable solution while traveling. Unlike other pothole detection applications, this application does not require a Graphical User Interface (GUI) or user prompts to start or stop detecting potholes. Instead, it automatically records video when the user opens the app using any mobile default voice assist. The application continuously scans the live recording video to detect potholes, and when one is detected, the GTTS system alerts the user with a loud and clear message, "Whoa!!! Be careful, there is a pothole in front of you". The application does not require users to register an account, and a decent mobile camera is enough to capture live video for pothole detection. This proposed methodology provides an efficient and user-friendly solution for visually impaired individuals to avoid accidents caused by potholes on roads. Fig. 1 shows the system architecture of mobile app of the user.

3.1. YOLO v5 model

YOLO is an algorithm that identifies and recognizes various objects in real-time images. YOLOv5 is preferred over the latest version of YOLO because of its ability to process custom datasets quickly, provide fast inference on CPU systems, and maintain stable memory utilization during training. Object detection in YOLO uses regression to calculate the probabilities of detected classes in the image. The algorithm utilizes CNN to detect objects in the image, and as the name suggests, it only requires one pass through the network to identify objects. To enhance the You Only Look Once object detection model, Ultralytics introduced YOLOv5. YOLOv5 utilizes a more complex architecture called EfficientDet that is based on the EfficientNet architecture. This new architecture allows for higher accuracy and a wider range of object classes to be detected. Unlike YOLO that is trained on the PASCAL VOC dataset, YOLOv5 was trained on a larger dataset called D5, which includes 600 object categories. In addition, YOLOv5 employs a new method for generating anchor boxes known as Dynamic anchor boxes. The major components of YOLOv5 algorithm used in the proposed Pothole detection system is shown in Fig. 2.

YOLOv5 itself contains five models. YOLOv5n: It is a nano model, YOLOv5s: It is a small model, YOLOv5m: It is a medium model, YOLOv5l: It is a large model, YOLOv5x: It is an extra-large model. The performance of different versions of YOLOv5 with EfficientDet is performed by Jocher Glenn et al. [54] is shown in Fig. 3. Implementing the proposed YOLOv5 for pothole detection involves preparing a dataset of annotated images and training the model to predict pothole bounding boxes and classes. Then the model has been integrated into an application for real-time inference. Key steps include data augmentation, model training with appropriate loss functions, and post-processing with techniques like non-maximum suppression. The process aims to optimize detection accuracy and robustness for real-world scenarios.

To build predictions, YOLO v5 comprises three components: Backbone, Neck, and Head. Its backbone is CSP-Darknet53 (Cross Stage Partial – Darknet53). It incorporates Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet) as the neck of the architecture. The detailed network architecture diagram of YOLOv5 is shown in Fig. 4.

The backbone of YOLOv5 serves as the feature extractor, responsible for capturing features from the input image. It is based on a

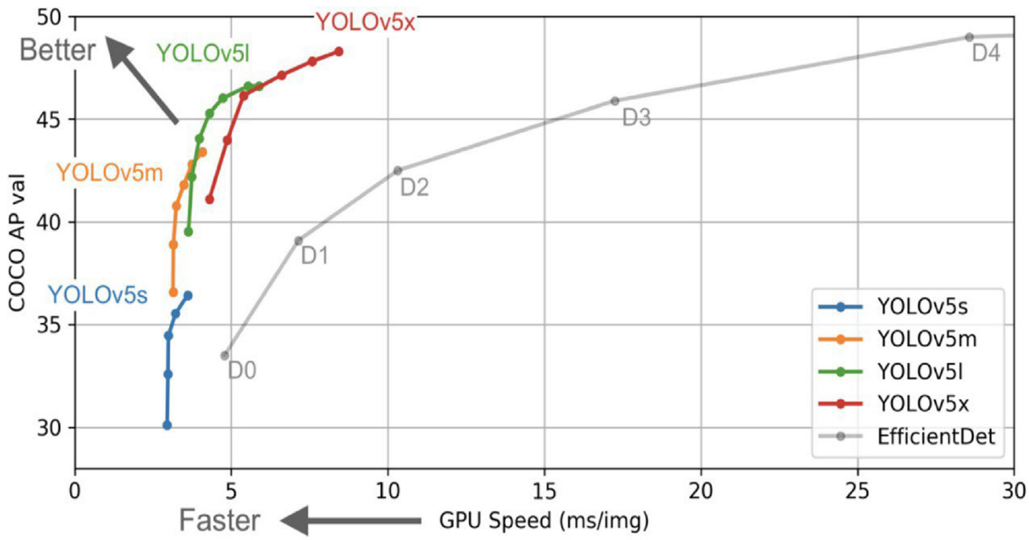

Fig. 3. Performance of YOLOv5 different sizes models [54].

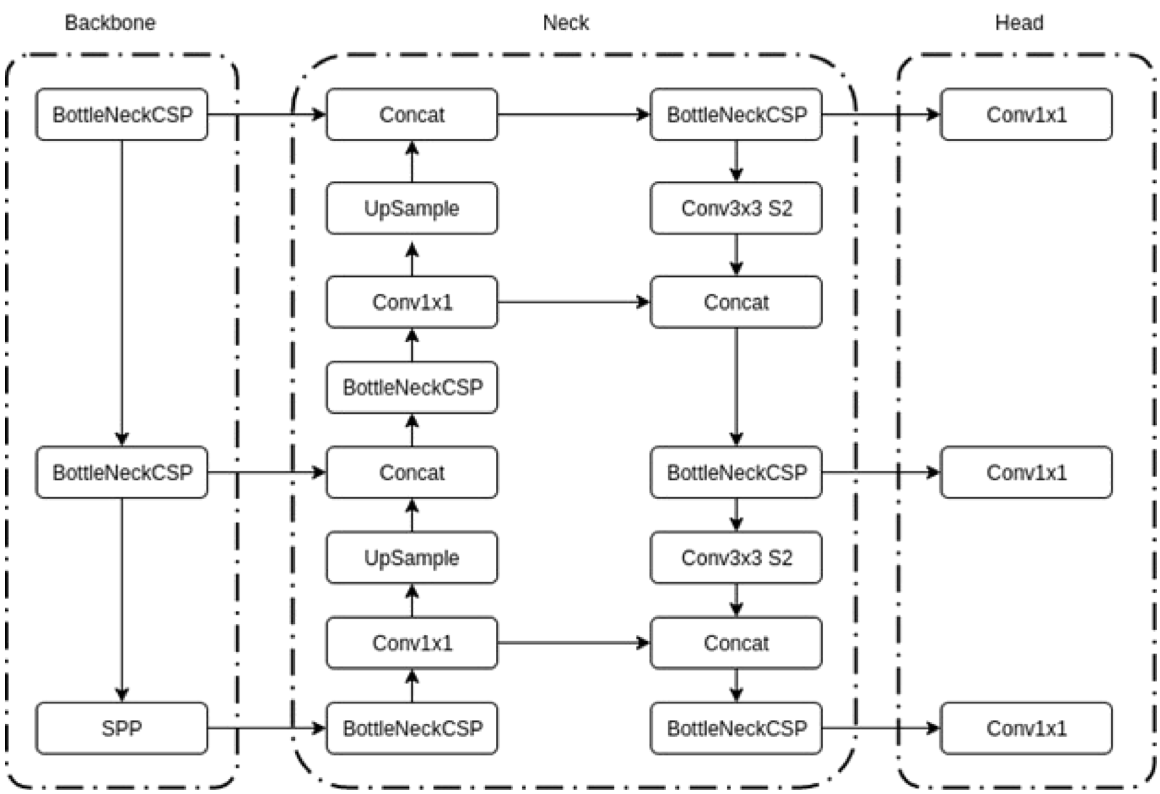

Fig. 4. YOLOv5 network architecture.

modified version of CSPDarknet53, a convolutional neural network architecture. By utilizing various convolutional layers with different filter sizes and strides, the backbone captures features at different scales, leveraging operations like convolutions, batch normalization, activation functions (e.g., Leaky ReLU), and spatial pooling. This process enables the extraction of both low-level details and high-level semantic information. The neck component in YOLOv5, known as PANet (Path Aggregation Network), enhances the feature representation obtained from the backbone. PANet aggregates information from feature maps of different resolutions by combining them through upsampling and concatenation operations. This fusion of multi-scale features allows the network to capture fine-grained details and global context, facilitating the detection of objects of varying sizes.

The head component generates the final predictions, including bounding box coordinates and class probabilities. It receives the fused feature maps from the neck and further processes them using additional convolutional and linear layers. The head predicts bounding boxes by applying convolutional filters to the feature maps. For each grid cell, YOLOv5 predicts a fixed number of bounding boxes defined by their four coordinates (x, y, *width*, and *height*). Additionally, the head predicts confidence scores for each bounding box, indicating the likelihood of containing an object. Class probabilities are also predicted for each bounding box, representing the probability of belonging to different object classes. Non-maximum suppression (NMS) is then applied to remove duplicate and overlapping predictions, retaining only the most confident and non-overlapping detections.

Fig. 5. Sample images from dataset.

3.2. Bounding boxes

A bounding box is a shape that emphasizes an object in an image. It has several characteristics, including height (b_h), width (b_w), class (c), and center coordinates (b_x, b_y). The x and y coordinate of the grid cell is represented by (c_x, c_y). The way to calculate the target coordinates for the bounding boxes has been modified from previous versions.

$$b_x = \left(2 \cdot \sigma\left(t_x\right) - 0.5\right) + c_x \tag{1}$$

$$b_y = \left(2 \cdot \sigma\left(t_y\right) - 0.5\right) + c_y \tag{2}$$

$$b_w = p_w \cdot \left(2 \cdot \sigma\left(t_w\right)\right)^2 \tag{3}$$

$$b_h = p_h \cdot \left(2 \cdot \sigma\left(t_h\right)\right)^2 \tag{4}$$

3.3. Intersection over union (IOU)

In object detection, IOU measures the degree of overlap between bounding boxes. YOLO utilizes this metric to generate a precise output box that fully encloses the object. Each grid cell is tasked with estimating the bounding boxes and their corresponding confidence scores. If the predicted bounding box matches the actual bounding box, the IOU value is 1, and any inaccurate bounding boxes are disregarded through this approach.

3.4. Loss function

Once the input image undergoes a single pass through the network, the YOLOv5 algorithm produces the object classes, their corresponding bounding boxes, and objectness scores. YOLOv5 employs Binary Cross Entropy (BCE) to determine the loss of classes and objectness, while Complete Intersection Over Union (CIOU) is used to calculate the location loss. The final loss equation for YOLOv5 is provided below.

$$\text{Loss} = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \tag{5}$$

- L_{cls}: The classification loss, which measures the discrepancy between predicted class probabilities and the true class labels for the detected objects.
- L_{obj}: The objectness loss, which penalizes incorrect predictions of object presence or absence.
- L_{loc}: The localization loss, which measures the difference between predicted bounding box coordinates and the ground truth box coordinates.
- The loss function aggregates these components by multiplying each of them with their respective weight (λ_1, λ_2, and λ_3) and summing them up.

4. Experimental analysis

YOLO is implemented for pothole detection by utilizing a pre-trained model initially trained on general datasets COCO. The dataset is curated containing diverse images of potholes annotated with bounding boxes. This dataset includes various pothole types, lighting conditions, weather scenarios, and road surfaces. The proposed model adjusts the model's weights to specialize in detecting potholes, optimizing for accuracy metrics like mAP. Post-training, the YOLO model is deployed for real-time pothole detection. It analyzes input images or video frames, predicts bounding boxes around detected potholes, and assigns confidence scores to each detection. This approach leverages YOLO's efficiency and speed, making it suitable for applications requiring rapid and accurate identification of potholes in road inspection and maintenance contexts.

4.1. Data collection

Gathering data is the initial stage of the work, and the quality and quantity of the dataset utilized for training can have a notable impact on the model's performance. The path detection accuracy can be increased by using a more comprehensive dataset containing images of different lighting conditions by [15]. For our proposed pothole detection mobile application, we used a public "New pothole detection Image Dataset" from roboflow, which consists of 9240 images [55]. The sample images of the dataset are shown in Fig. 5. To ensure the accuracy and effectiveness of the model, the dataset is partitioned into three distinct subsets as given in Table 2: one for training with 6091 images, one for validation with 2094 images, and one for testing with 1055 images. During the training phase, the model was trained using the training dataset, which contained txt format labels for each image. These labels consisted of class_id, center_x, center_y, width, and height attributes, which were used to train the model to recognize and locate potholes accurately.

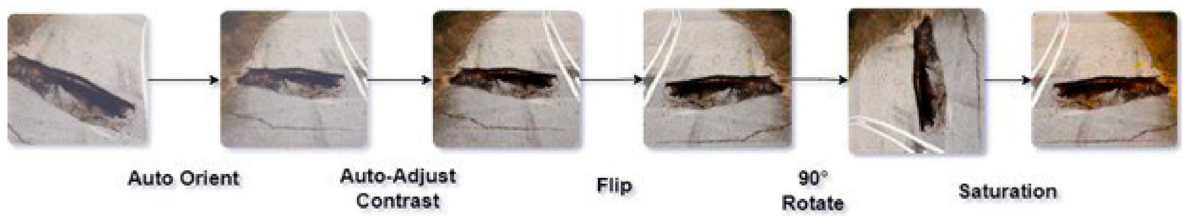

Fig. 6. Step by step pre-processing results of a sample input image.

Table 2
Details about public pothole detection dataset images [55].

Dataset	No of images
Training dataset	6091
Validation dataset	2094
Testing dataset	1055
Total	9240

Table 3
Parameter/methods used in training the YOLO v5 model.

Parameter/Method name	Value
Mode	Single GPU
Number of GPUs	0
Dataset size	425
Batch size per GPU	16
Batch accumulate	1
Total batch size	16
Effective batch size	16
Iterations per epoch	50
Gradient updates per epoch	50
Image size	640 × 640
Learning rate	0.001
Optimizer	Adam optimizer
Loss function	A combination of binary cross-entropy, focal, and GIoU loss function

By using a public dataset, we are able to train the model on a diverse range of images and scenarios, which helped to improve its accuracy and effectiveness. Furthermore, the use of txt format labels enabled us to extract relevant information from the dataset easily.

Overall, the dataset used in training the model was crucial to the success of our proposed pothole detection mobile application. By using a high-quality and diverse dataset, we were able to train a model that accurately detects potholes.

4.2. Data pre-processing

In pothole detection using YOLO algorithm, data preprocessing plays a crucial role in improving the accuracy of the model. We followed some of the specific preprocessing and augmentation steps involved in pothole detection are listed below:

- *Auto-Orient*: This preprocessing step is applied to ensure that all images are correctly oriented.
- *Auto-Adjust Contrast*: The contrast of the images is adjusted using contrast stretching to ensure that the features of potholes are visible and distinguishable.
- *Flip and 90° Rotate*: Horizontal flipping, clockwise and counter-clockwise rotation are performed as an augmentation step to increase the number of training examples and to make the model more robust to pothole detection with different orientations.
- *Saturation*: Saturation is adjusted between −25% and +25% as an augmentation step to make the model more robust to potholes in different lighting conditions.

The preprocessing results on a sample input image is shown in Fig. 6. By applying these preprocessing and augmentation steps, the YOLO model can be trained to accurately detect potholes in a variety of conditions, including varying lighting, orientation, and appearance.

4.3. Training the YOLO algorithm

The YOLOv5 model comprises three distinct components, namely the backbone, neck, and head networks. The backbone network is in charge of extracting features from the input image. The neck network then processes these features further, while the head network generates the ultimate bounding boxes and predict object class probabilities. The purpose of the loss function is to evaluate the disparity between the predicted bounding boxes and the actual ground truth bounding boxes. In the YOLOv5 model, a combination of binary cross-entropy loss, focal loss, and GIoU loss functions are used. Fine-tuning hyperparameters is a crucial process that enhances the model's efficiency by modifying different parameters, including the learning rate, batch size, and number of epochs etc. The hyperparameters and methods used in training the YOLO v5 Model is listed in the Table 3.

The proposed approach is implemented on google colab platform. Google colab is a cloud-based service that provides free access to GPU and TPU resources, was used for model training. Throughout the training process, the model computed losses and metrics for each epoch, including box loss, objectness loss, and classification loss. The box loss evaluates the correspondence between the predicted bounding box and the actual ground truth bounding box, while the objectness loss evaluates the model's ability to predict the presence of an object in a specified bounding box. The classification loss measures how well the model is able to classify the detected object into the correct category.

4.4. Testing the YOLO algorithm

After training the YOLO algorithm, the next step is to evaluate its performance on a test set using precision, recall, and mean average precision (mAP) metrics. To begin the testing process, a separate test set of images with ground truth labels is used to run the trained model on unseen data. Once the YOLO algorithm is applied to the test set, the predicted bounding boxes and object classes are compared to the ground truth labels, then precision and recall values are calculated. Precision measures how many of the predicted objects are correct, while recall measures how many of the true objects are detected by the algorithm. The mAP is a widely used assessment metric for evaluating object detection models. It computes the mean precision across all object classes and IoU thresholds.

The IoU threshold determines the extent of overlap between the predicted and actual ground truth bounding boxes that qualifies as a correct detection. A higher mAP value indicates better performance, while a lower mAP value indicates poorer performance. Therefore, a higher mAP value is desirable. In addition to precision, recall, and mAP, we also evaluated the performance of the YOLOv5 model on live video stream data. It was able to detect potholes in real-time video, processing 30 frames per second at a resolution of 1280 × 720. Overall, testing the YOLO algorithm after training ensures that the model performs well on unseen data and can accurately detect objects in real-world scenarios.

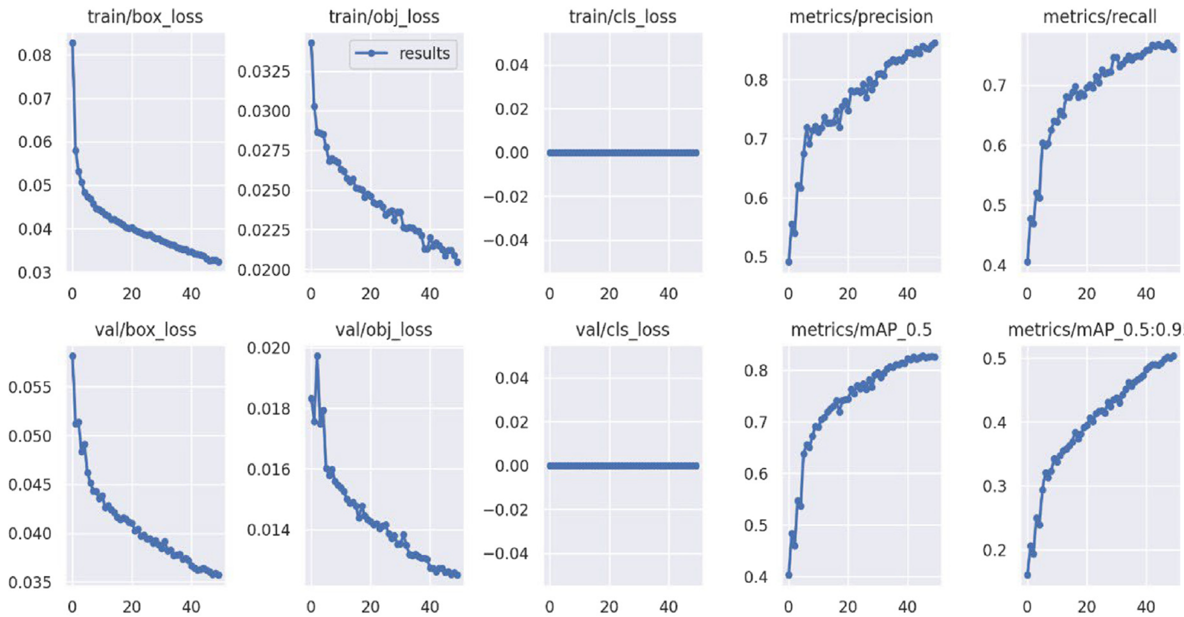

Fig. 7. Plotting the results of three losses and precision/recall metrics.

4.5. Results

To assess the performance of the trained model, a set of graphs are plotted to show the losses and metrics for each epoch (figure 7). The graph exhibits three types of losses: box loss, objectness loss, and classification loss. The box loss determines how accurately the predicted bounding box matches the ground truth bounding box, while the objectness loss measures how well the predicted bounding box encompasses the object in the image. The classification loss measures how correctly the model predicts the object's correct class in the image. As only one class was used in this study, the classification loss remained unchanged at zero. Val Box, Val Objectness, and Val Classification are the model's measures of box loss, objectness loss, and classification loss during validation. The X-axis denotes Number of Epochs where as Y-axis varies 0.0 to 1.0 as percentage for evaluation to 100% (see Fig. 7).

Precision and recall were calculated to measure the accuracy of the YOLOv5 model on pothole image dataset. Recall is the fraction of the true positive detections (TP) over the total number of actual positives (TP + FN), where FN is the number of false negatives.

$$\mathrm{Recall} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{6}$$

Precision is the fraction of the true positive detections (TP) over the total number of predicted positives (TP + FP), where FP is the number of false positives.

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{7}$$

The Mean average precision (*mAP*) is the average precision over different recall levels. The precision–recall curve is first computed and then the area under the curve (AUC) is calculated. The AUC value is then averaged over all classes to get the MAP. The MAP is calculated as:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{8}$$

where N is the number of classes, AP_i is Average precision of class i. At the end of 50 epochs the model achieved results for various metrics is given in Table 4. It compares the performance metrics such as precision, recall, mAP at 0.5, validation box loss, objectness loss, and class loss for the proposed YOLO5, YOLO-NAS, YOLO8, and YOLO7 algorithms. It has been found that the proposed YOLO5 algorithm produces improved results as 0.86214, 0.75923, 0.82531, 0.035721, 0.012505, and 0 percentage for precision, recall, mAP at 0.5, validation box loss, objectness loss, and class loss respectively as compared with other algorithms.

Table 4
Various metrics and its values.

Metrics	Proposed YOLO5	YOLO-NAS	YOLO8	YOLO7
Precision	**0.86214**	0.014	0.738	0.693
Recall	**0.75923**	0.8759	0.631	0.63
mAP @ 0.5	**0.82531**	0.3755	0.667	0.639
Validation box loss	**0.035721**	0.2140492	1.988	–
Validation obj loss	**0.012505**	5.2964644	1.234	–
Validation cls loss	**0.009851**	1.6696	1.062	–

As shown in Fig. 8, the precision for proposed YOLO5 is 86.2% whereas, the YOLO-NAS, YOLO8, and YOLO7 algorithms produce 1.4, 73.8, 69.3% respectively; the proposed YOLO5 produced 12%–84% of improved results than the compared algorithms. The mAP @0.5 for proposed YOLO5 is 82.5% whereas, the YOLO-NAS, YOLO8, and YOLO7 algorithms produce 37.5, 66.7, 63.9% respectively; the proposed YOLO5 produced 16%–45% of improved results than the compared algorithms. The recall for YOLO-NAS is 87.5, but for the proposed YOLO5 is 75.9; however, the overall performance for the proposed YOLO5 in other metrics precision, mAP at 0.5, validation box loss, objectness loss, and class loss is much higher than the YOLO-NAS algorithm. Similarly, the proposed YOLO5 algorithm has reduced validation box loss, and class loss compared to the YOLO-NAS, YOLO8, and YOLO7 algorithms as depicted in Fig. 9. Overall, the proposed YOLO5 algorithm has better performance with lower validation loss and improved precision values.

The *mAP* was also calculated for different IoU thresholds, such as 0.5 and 0.5 to 0.95 IoU. The *mAP* metric provides a single number to evaluate the overall performance of the YOLOv5 algorithm. The pothole detection results from dataset images and live video stream are illustrated in Figs. 8 and 9 respectively. The pothole detection results on images and videos are shown in Figs. 10 and 11 respectively. From the figures, our approach precisely marked the pothole region using bounding boxes. The detection accuracy is also printed along the bounding boxes to make the representation very clear.

After training, the YOLOv5 model was tested against a set of unseen images of size 416 × 416. The model achieved accuracy of 82.7% when detecting images in the testing set. When detecting live

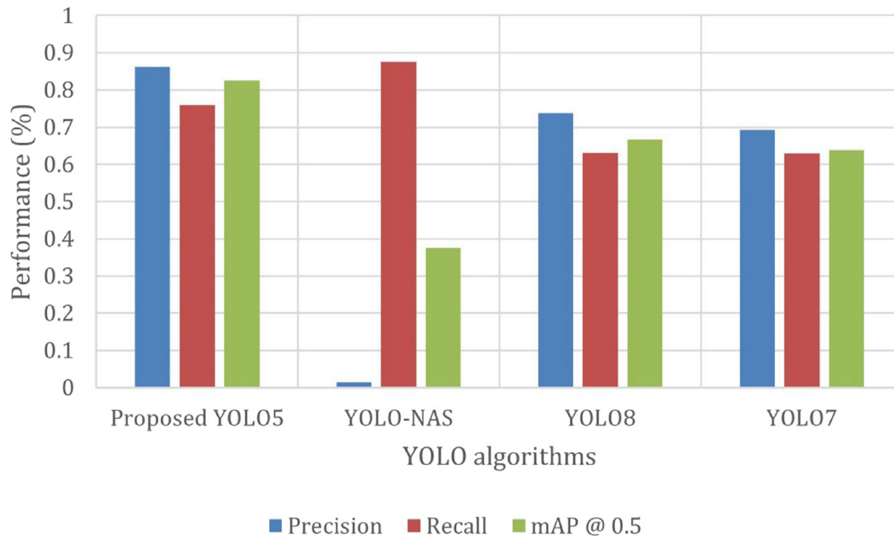

Fig. 8. YOLO algorithms vs. performance metrics.

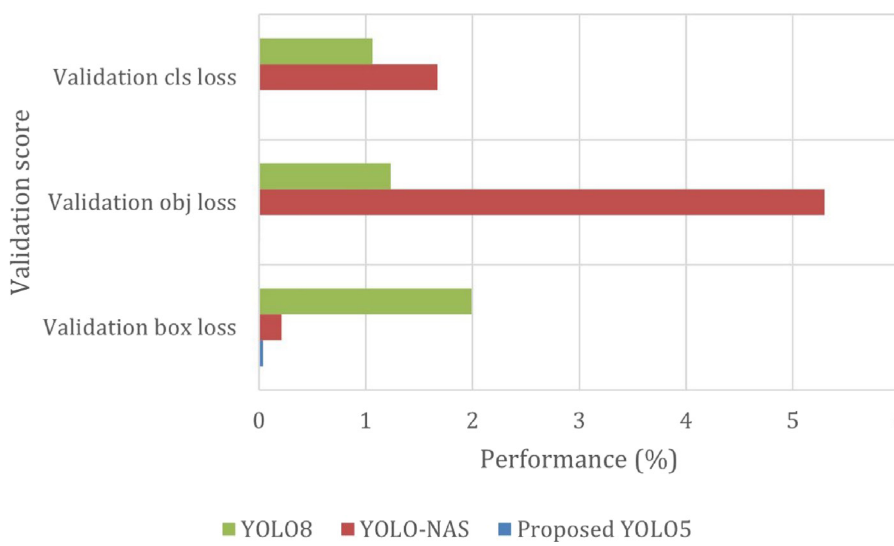

Fig. 9. YOLO algorithms vs. performance metrics.

video, it achieved rate of 30 frames per second at a resolution of 1280 × 720. Overall, the experimental analysis demonstrates that the YOLOv5 model can effectively detect potholes with high accuracy and speed, making it a promising tool for pothole detection in real-world scenarios.

5. Conclusion

In this work, we developed an application that can detect potholes and alert visually impaired individuals to prevent accidents and injuries. The idea presented is a significant step toward improving the safety of visually impaired individuals while navigating roads. We have implemented a pothole detection model and tested on public pothole detection dataset that achieved 82.7% accuracy in detecting potholes in dataset images. Our model also provides faster responses and good accuracy in case of real-time video data for pothole detection task. However, the implementation of the model on a mobile phone has not been carried out yet. Additionally, the model may have limitations in detecting potholes that are far away from the user, and it is currently only trained to detect potholes. To address these limitations, the future work plan is to enhance the system's performance by training it to detect other objects besides potholes. This improvement will help visually impaired individuals to avoid other hazards while navigating roads. Furthermore, future work is aimed at validating and improving generalizability to diverse road conditions. To improve the accuracy of the proposed model, the comprehensive evaluation using metrics like IoU, diverse training data, and continuous learning strategies are focused. The future work also focuses on detecting potholes at greater distances involves integrating advanced depth estimation techniques, leveraging alternative camera angles, enhancing sensor fusion capabilities, applying tailored machine learning models, and ensuring real-time processing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

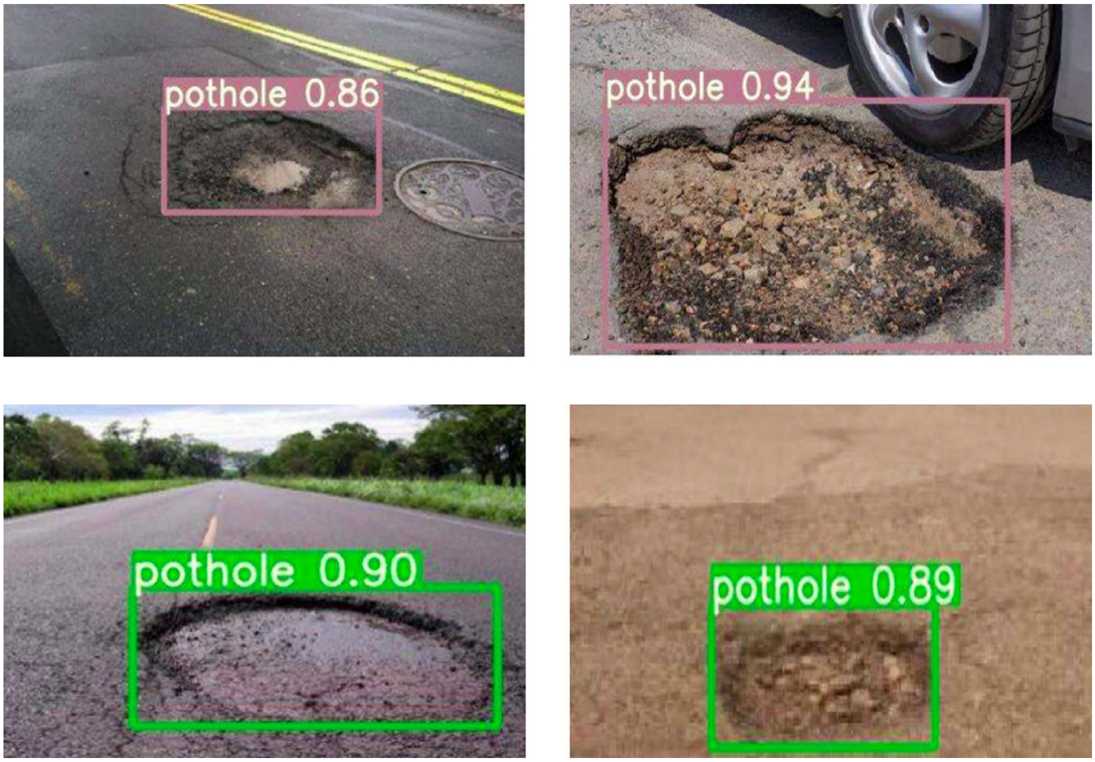

Fig. 10. Pothole detected in images.

Fig. 11. Pothole detected in video.

References

[1] P. Arjun, J. Anitha, A.H. Thahseen, A survey on object recognition system, Int. J. Appl. Eng. Res. 10 (72) (2015) 295–300.

[2] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, 2016, pp. 779–788, http://dx.doi.org/10.1109/CVPR.2016.91.

[3] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, USA, 2017, pp. 6517–6525, http://dx.doi.org/10.1109/CVPR.2017.690.

[4] Joseph Redmon, Ali Farhadi, YOLOv3: An incremental improvement, 2018, http://dx.doi.org/10.48550/arXiv.1804.02767, arXiv:1804.02767v1 [cs.CV].

[5] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004.10934 [cs.CV]. https://doi.org/10.48550/arXiv.2004.10934.

[6] Kshitija Chavan, Chinmay Chawathe, Vatsal Dhabalia, Amruta Sankhe, Pothole detection system using YOLO v4 algorithm, Int. Res. J. Eng. Technol. 09 (04) (2022) e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[7] R. Anusuya, P. Arjun, R. Haripriya, C. Paulraj, A survey on machine parts recognition and defect detection using image processing techniques, J. Appl. Sci. Comput. 6 (3) (2019) 1–5.

[8] M.M. Islam, M.S. Sadi, Path hole detection to assist the visually impaired people in navigation, in: 2018 4th International Conference on Electrical Engineering and Information & Communication Technology, iCEEiCT, Dhaka, Bangladesh, 2018, pp. 268–273, http://dx.doi.org/10.1109/CEEICT.2018.8628134.

[9] R. Hiremath, K. Malshikare, M. Mahajan, R.V. Kulkarni, A smart app for pothole detection using Yolo model, in: S. Fong, N. Dey, A. Joshi (Eds.), ICT Analysis and Applications, in: Lecture Notes in Networks and Systems, vol. 154, Springer, Singapore, 2021, http://dx.doi.org/10.1007/978-981-15-8354-4_16.

[10] P. Ping, X. Yang, Z. Gao, A deep learning approach for street pothole detection, in: 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications, BigDataService, Oxford, UK, 2020, pp. 198–204, http://dx.doi.org/10.1109/BigDataService49289.2020.00039.

[11] P. Theerthagiri, A.U. Ruby, B.N. Chaithanya, R.R. Patil, S. Jain, D-Resnet: deep residual neural network for exploration, identification, and classification of beach sand minerals, Multimedia Tools Appl. (2023) 1–25.

[12] C. Gopala Krishnan, A.H. Nishan, T. Prasannavenkatesan, I. Jeena Jacob, G. Komarasamy, Two dimensional and gesture based medical visualization interface and image processing methodologies to aid and diagnose of lung cancer, in: Advanced Machine Learning Approaches in Cancer Prognosis: Challenges and Applications, Springer International Publishing, Cham, 2021, pp. 297–314.

[13] S. Saraswathi, C. Gopala Krishnan, P.V. Theerthagiri, Gesture-based smart-assistive device for elderly and disabled people using IoT, in: IoT Based Smart Applications, Springer International Publishing, Cham, 2022, pp. 71–88.

[14] N. Camilleri, T. Gatt, Detecting road potholes using computer vision techniques, in: 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, ICCP, Cluj-Napoca, Romania, 2020, pp. 343–350, http://dx.doi.org/10.1109/ICCP51029.2020.9266138.

[15] D. J, S.D. V, A. S A, K. R, L. Parameswaran, Deep learning based detection of potholes in Indian roads using YOLO, in: 2020 International Conference on Inventive Computation Technologies, ICICT, Coimbatore, India, 2020, pp. 381–385, http://dx.doi.org/10.1109/ICICT48043.2020.9112424.

[16] R. Rastogi, U. Kumar, A. Kashyap, S. Jindal, S. Pahwa, A comparative evaluation of the deep learning algorithms for pothole detection, in: 2020 IEEE 17th India Council International Conference, INDICON, New Delhi, India, 2020, pp. 1–6, http://dx.doi.org/10.1109/INDICON49873.2020.9342558.

[17] R. Agrawal, Y. Chhadva, S. Addagarla, S. Chaudhari, Road surface classification and subsequent pothole detection using deep learning, in: 2021 2nd International Conference for Emerging Technology, INCET, Belagavi, India, 2021, pp. 1–6, http://dx.doi.org/10.1109/INCET51464.2021.9456126.

[18] K. Gajjar, T. van Niekerk, T. Wilm, P. Mercorelli, Vision-based deep learning algorithm for detecting potholes, in: Journal of Physics: Conference Series, vol. 2162, IOP Publishing, 2022, 012019.

[19] P.A. Chitale, K.Y. Kekre, H.R. Shenai, R. Karani, J.P. Gala, Pothole detection and dimension estimation system using deep learning (YOLO) and image processing, in: 2020 35th International Conference on Image and Vision Computing New Zealand, IVCNZ, Wellington, New Zealand, 2020, pp. 1–6, http://dx.doi.org/10.1109/IVCNZ51579.2020.9290547.

[20] S. Silvister, et al., Deep learning approach to detect potholes in real-time using smartphone, in: 2019 IEEE Pune Section International Conference, PuneCon, Pune, India, 2019, pp. 1–4, http://dx.doi.org/10.1109/PuneCon46936.2019.9105737.

[21] Yangsong Gu, Yuandong Liu, Diyi Liu, Lee D. Han, Xiaoyang Jia, Spatiotemporal kernel density clustering for wide area near Real-Time pothole detection, Adv. Eng. Inform. 60 (2024) 102351.

[22] Eva Lieskovská, Maroš Jakubec, Boris Bučko, Katarína Zábovská, Automatic pothole detection, Transp. Res. Procedia 74 (2023) 1164–1170.

[23] P.C. Nissimagoudar, Sanjana R. Miskin, Varsha N. Sali, J. Ashwini, S.K. Rohit, S.K. Darshan, H.M. Gireesha, Rohini S. Hongal, Supriya V. Katwe, Basavaraj, C.I. Nalini, Detection of potholes and speed breaker for autonomous vehicles, Procedia Comput. Sci. 237 (2024) 675–682.

[24] Muhammad Ihsan, Muhammad Alfian Amrizal, Agus Harjoko, A pothole video dataset for semantic segmentation, Data Brief 53 (2024) 110131.

[25] Md. Yearat Hossain, Ifran Rahman Nijhum, Md. Tazin Morshed Shad, Abu Adnan Sadi, Md. Mahmudul Kabir Peyal, Rashedur M. Rahman, An end-to-end pollution analysis and detection system using artificial intelligence and object detection algorithms, Decis. Anal. J. 8 (2023) 100283.

[26] M. Sathvik, G. Saranya, S. Karpagaselvi, An intelligent convolutional neural network based potholes detection using Yolo-V7, in: 2022 International Conference on Automation, Computing and Renewable Systems, ICACRS, Pudukkottai, India, 2022, pp. 813–819.

[27] G. Arvanitis, N. Stagakis, E.I. Zacharaki, K. Moustakas, Cooperative saliency-based pothole detection and AR rendering for increased situational awareness, IEEE Trans. Intell. Transp. Syst. 25 (5) (2024) 3588–3604.

[28] G. Raja, S. Anbalagan, S. Senthilkumar, K. Dev, N.M.F. Qureshi, SPAS: Smart pothole-avoidance strategy for autonomous vehicles, IEEE Trans. Intell. Transp. Syst. 23 (10) (2022) 19827–19836.

[29] X. Chen, N. Ma, M. Wang, YOLOv5-pothole: An improved pothole perception method based on YOLOv5-seg, in: 2023 2nd International Conference on Artificial Intelligence, Human-Computer Interaction and Robotics, AIHCIR, Tianjin, China, 2023, pp. 75–79.

[30] Bibi, K. Ali, A. Raza, S. Kausar, Real-time multi-scale pothole detection using transformer, in: 2023 International Conference on Frontiers of Information Technology, FIT, Islamabad, Pakistan, 2023, pp. 114–119.

[31] Kanchi Anantharaman Vinodhini, Kovilvenni Ramachandran Aswin Sidhaarth, Pothole detection in bituminous road using CNN with transfer learning, Meas.: Sens. 31 (2024) 100940.

[32] S. Chougule, A. Barhatte, Smart pothole detection system using deep learning algorithms, Int. J. ITS Res. 21 (2023) 483–492, http://dx.doi.org/10.1007/s13177-023-00363-3.

[33] P. Singh, A.E. Kamal, A. Bansal, et al., Road pothole detection from smartphone sensor data using improved LSTM, Multimedia Tools Appl. 83 (2024) 26009–26030, http://dx.doi.org/10.1007/s11042-023-16177-0.

[34] P. Kumar, Pooja, N. Chauhan, et al., A vision-based pothole detection using CNN model, SN Comput. Sci. 4 (2023) 716, http://dx.doi.org/10.1007/s42979-023-02153-w.

[35] J. Lekshmipathy, S. Velayudhan, S. Mathew, Effect of combining algorithms in smartphone based pothole detection, Int. J. Pavement Res. Technol. 14 (2021) 63–72, http://dx.doi.org/10.1007/s42947-020-0033-0.

[36] J. Ha, D. Kim, M. Kim, Assessing severity of road cracks using deep learning-based segmentation and detection, J. Supercomput. 78 (2022) 17721–17735, http://dx.doi.org/10.1007/s11227-022-04560-x.

[37] G. Guo, Z. Zhang, Road damage detection algorithm for improved YOLOv5, Sci. Rep. 12 (2022) 15523, http://dx.doi.org/10.1038/s41598-022-19674-8.

[38] Chen-Chiung Hsieh, Han-Wen Jia, Wei-Hsin Huang, Mei-Hua Hsih, Deep learning-based road pavement inspection by integrating visual information and IMU, Information 15 (4) (2024) 239, http://dx.doi.org/10.3390/info15040239.

[39] Maroš Jakubec, Eva Lieskovská, Boris Bučko, Katarína Zábovská, Comparison of CNN-based models for pothole detection in real-world adverse conditions: Overview and evaluation, Appl. Sci. 13 (9) (2023) 5810, http://dx.doi.org/10.3390/app13095810.

[40] Gyulim Kim, Seungku Kim, A road defect detection system using smartphones, Sensors 24 (7) (2024) 2099, http://dx.doi.org/10.3390/s24072099.

[41] Furkan Ozoglu, Türkay Gökgöz, Detection of road potholes by applying convolutional neural network method based on road vibration data, Sensors 23 (22) (2023) 9023, http://dx.doi.org/10.3390/s23229023.

[42] Abdullah As Sami, Saadman Sakib, Kaushik Deb, Iqbal H. Sarker, Improved YOLOv5-based real-time road pavement damage detection in road infrastructure management, Algorithms 16 (9) (2023) 452, http://dx.doi.org/10.3390/a16090452.

[43] Boris Bučko, Eva Lieskovská, Katarína Zábovská, Michal Zábovský, Computer vision based pothole detection under challenging conditions, Sensors 22 (22) (2022) 8878, http://dx.doi.org/10.3390/s22228878.

[44] G. Bosurgi, M. Modica, O. Pellegrino, G. Sollazzo, An automatic pothole detection algorithm using pavement 3D data, Int. J. Pavement Eng. 24 (2) (2022) http://dx.doi.org/10.1080/10298436.2022.2057978.

[45] Muhammad Haroon Asad, Saran Khaliq, Muhammad Haroon Yousaf, Muhammad Obaid Ullah, Afaq Ahmad, Pothole detection using deep learning: A real-time and AI-on-the-edge perspective, Adv. Civ. Eng. (2022) 9221211, http://dx.doi.org/10.1155/2022/9221211, 13 pages.

[46] Zhuo Diao, Xianfu Huang, Han Liu, Zhanwei Liu, LE-YOLOv5: A lightweight and efficient road damage detection algorithm based on improved YOLOv5, Int. J. Intell. Syst. 2023 (2023) 8879622, http://dx.doi.org/10.1155/2023/8879622, 17 pages.

[47] Min-Li Lan, Dan Yang, Shuang-Xi Zhou, Yang Ding, Crack detection based on attention mechanism with YOLOv5, Eng. Rep. (2024) e12899, http://dx.doi.org/10.1002/eng2.12899.

[48] M. Sharma, S.R. Saripalli, A.K. Gupta, R. Talwar, P. Dadheech, U.K. Kanike, Real-time pothole detection during rainy weather using dashboard cameras for driverless cars, in: B. Pandey, D. Pandey, R. Anand, D. Mane, V. Nassa (Eds.), Handbook of Research on Thrust Technologies' Effect on Image Processing, IGI Global, 2023, pp. 384–394, http://dx.doi.org/10.4018/978-1-6684-8618-4.ch023.

[49] Anita Maria da Rocha Fernandes, et al., Detection and classification of cracks and potholes in road images using texture descriptors, J. Intell. Fuzzy Systems 44 (6) (2023) 10255–10274.

[50] Xuejing Chen, Sira Yongchareon, Martin Knoche, A review on computer vision and machine learning techniques for automated road surface defect and distress detection, J. Smart Cities Soc. 1 (4) (2022) 259–275.

[51] B. Srivani, Ch. Kamala, S. Renu Deepti, G. Aakash, Pothole detection using convolutional neural network, AIP Conf. Proc. 2935 (1) (2024) 020009.

[52] Nutan Patel, Vipul Dabhi, Rachit Adhvaryu, Review on identify road potholes using image semantic segmentation for advance driver assistant system, AIP Conf. Proc. 3107 (1) (2024) 040001, http://dx.doi.org/10.1063/5.0208439.

[53] Cahya Buana, Hera Widyastuti, Catur Arif Prastyanto, S. Rahardhita Widyatra, Application of image sensors to obtain potholes data on road damage, AIP Conf. Proc. 3026 (1) (2024) 080035, http://dx.doi.org/10.1063/5.0199794.

[54] Jocher Glenn, et al., YOLOv5 by ultralytics, 2020, http://dx.doi.org/10.5281/zenodo.3908559, v7.0. https://github.com/ultralytics/yolov5.

[55] Dataset : https://universe.roboflow.com/smartathon/new-pothole-detection/dataset/2 (Accessed 12 September 2023).