

The road conditions detection using the convolutional neural network

Sujittra Sa-ngiem¹, Kwankamon Dittakan², Saroch Boonsiripant³

¹College of Digital Science, Prince of Songkla University, Songkhla, Thailand

²College of Computing, Prince of Songkla University, Phuket Campus, Phuket, Thailand

³Faculty of Engineering, Kasetsart University, Bangkok, Thailand

Article Info

Article history:

Received Aug 19, 2024

Revised Jun 3, 2025

Accepted Jul 3, 2025

Keywords:

Deep learning

Object detection

Road detection

Road surface condition

Road surface detection

ABSTRACT

Poor road conditions present considerable obstacles for individuals, resulting in asset loss, bodily harm, and time inefficiency. Approximately 1.35 million fatalities are attributable to road traffic incidents. The Department of Public Works and Town & Country Planning conducted road surveys to assess and strategize maintenance efforts. The manual car survey requires additional time and an excessive budget. The automated system of artificial intelligence (AI) is widely recognized. This paper presents a model to detect road surface conditions utilizing video data. Four versions of convolutional neural networks (CNN) were utilized for this work. The model evaluation employed the mean average precision (mAP) measure. The video data was acquired via a smartphone mounted in a vehicle, comprising 10,984 photos for training and 2,198 images for testing. We trained and evaluated four versions of CNN architectures named YOLO, utilizing our data and GPU, with a specific emphasis on identifying cracks, potholes, and the condition of manhole covers. Of the architectures evaluated, YOLO V6 attained the greatest mAP score in comparison YOLO V5 to YOLO V8. The testing results with batch sizes of 4, 8, 16, and 32 are effective. The batch size of 32 yields the highest performance, achieving 87.38% mAP. Conduct the dropout normalization using rates of 0.25, 0.50, 0.75, and 1. The maximum mAP is observed with a dropout rate of 0.25, yielding a mAP of 85.40%. The model indicates that the government conducted road surface inspections with enhanced efficiency, enabling the planning of road repairs for public utility issues, which can lower transportation costs. Additionally, the model can be utilized to identify hazardous road conditions and minimize vehicular accident rates.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Kwankamon Dittakan

College of Computing, Prince of Songkla University, Phuket Campus

Phuket, 83120 Kathu, Thailand

Email: kwankamon.d@phuket.psu.ac.th

1. INTRODUCTION

The road functions as an avenue capable of propelling an economy ahead. A smooth road surface facilitates faster and more efficient travel. Road traffic incidents result in 1.35 million fatalities [1]. Undoubtedly, roads are important to sustaining the economy and the livelihoods of individuals. This engenders economic and cultural advancement, enhancing numerous facets, including shipping, travel, communication, and others within the nation. Nonetheless, if the road surface is deteriorated and dangerous, it impedes traffic and endangers the entire transportation system. This problem leads to the loss of both assets and life. Moreover, the quality of the road surface is essential for maintaining safe driving, especially during

inclement weather. Moist road surfaces reduce tire traction, thereby increasing the likelihood of accidents. The condition of the road is crucial in assessing the safety of humans and their belongings [2]. Examples of road conditions include cracks, potholes, and manhole covers. Generally, the designated people would conduct a manual inspection of the road to evaluate its condition. Nevertheless, developments in technology have enabled cameras and sensors mounted on vehicles to gather video footage and crucial data. The personnel thereafter conduct a manual verification of the obtained video material. Despite being time-consuming, this process is crucial for the safety of lives and assets [3].

Currently, various methodologies are utilized to evaluate road surface conditions. Video images and cameras have been utilized for this purpose. Artificial intelligence (AI) and machine learning (ML) enhance road quality detecting systems. The convolutional neural network (CNN) is widely employed in artificial intelligence research. CNN is adept at evaluating road conditions using images, signals, and video data. Many research projects employ video or sensor data. YOLO, a term for "You Only Look Once," is extensively employed for real-time object detection tasks. Currently, many variants of YOLO improve performance in terms of mean average precision (mAP).

Categories of accident factors in road accident data management reports include rider, weather, vehicle performance, and road surface conditions. The collision was caused by the manhole cover, a crack, a pothole, and a bumper, attributable to the road conditions. The kind of road surface affects vehicle traction. If the irregular terrain guarantees safe transit, the wheels can maintain traction. Thailand faces multiple challenges resulting from the substandard quality of road surfaces, resulting in the development of potholes and cracks over time [4], [5]. The manhole cover is situated on the roadway. This generates an elevation and depression on the highway resulting from the ascent and descent of the manhole cover. A multitude of object types exists. The manhole cover comprises a small circle and a large rectangle. This confined the model's use exclusively to the data collection domain. This work proposes the utilization of a CNN on video data to create a classifier capable of identifying road surface quality issues, including potholes, cracks, and manhole covers. The results of this research can be utilized in i) assisting government agencies in optimizing road surveys and assessing road quality more effectively, thus improving road repair strategies, ii) reducing transportation expenses by resolving public utility concerns via the proposed method, iii) incorporating the classifier into an application to provide rider alerts upon identifying hazardous road conditions, and iv) potentially decreasing car accident rates through the application of this classifier. The framework of this endeavor begins with data collection through mobile applications that include video data. The preliminary stage entails frame extraction. The picture labeling is operational. The concluding phase involves developing the object detection model with CNN. The results are compared with several iterations of YOLO.

The subsequent sections of this document are structured as follows: i) introduction, ii) related work discusses relevant studies; iii) methods, encompassing the proposed framework, data collection, pre-processing procedures, and details of the CNN; iv) result and discussion, and v) conclusion discusses the final observations, and acknowledgments.

2. RELATED WORK

This section discusses multiple relevant research studies concentrated on the detection of abnormal objects on the roadway. Related research includes multiple domains, such as road quality assessment, the association between road speed and vibrations, and road classification. The study in this domain employs many hardware types, primarily categorized into two groups: i) smartphones and ii) accelerometers. The gramian angular summation field (GASF) is utilized to convert traffic time-series data into an image format, resulting in decreasing the error rate [6]. The study extracted data from single-axis and three-axis accelerometers. Machine learning and deep neural networks were utilized. The evaluation of classification performance employs many parameter sets. The data collection using an iPhone 6. Three categories of vehicles were utilized: i) Ford Focus sedan, ii) Ford Focus hatchback, and iii) Subaru Outback SUV. The Vibration Recorder application utilizes accelerometer data and video capture with DJI Osmo. The results indicated that utilizing all three axes of the accelerometer provided more precise outcomes compared to employing a single axis [7]. Utilized pre-processing techniques to remove noise in data preparation. Subsequently, feature extraction was conducted, followed by the promotion of predictive analysis.

The random forest (RF) and decision tree (DT) algorithms are employed for classification and the identification of pavement distress kinds [8]. A machine learning model utilizing a support vector machine (SVM) was developed [9]. The condition of the road pavement surface is evaluated by vibration measurements. The findings demonstrated 93% accuracy for the RF model, 90% for the DT model, and 96% for the SVM model. The U-type deep learning image segmentation model, known as RCNN-UNet, is utilized to extract centerlines and identify roadways. The findings indicate a completeness (COM) score of 0.9871, a correctness (COR) score of 0.9959, a quality (Q) score of 0.9744, and an F1 score of 0.9876 [10]. The deep

residual U-Net is utilized to delineate the road area in an aerial image [11]. Previous studies on road condition detection have highlighted the importance of devices for recording and storing video data. A variety of devices have been employed for road condition analysis.

The identification of potholes has been achieved using thermal cameras, as evidenced by the research conducted by Bhatia *et al.* [12]. These cameras, when integrated with a Raspberry Pi, provide a collection of data for the identification of potholes and bumps. Smartphones have gained popularity as an effective option for road data collecting, due to their high-resolution cameras and accessibility. Video data acquired from cellphones has been utilized to identify various road conditions, such as cracked surfaces, smooth roads, uneven terrain, potholes, rumble strips, and water [12]. Numerous studies have utilized smartphones to collect data for the detection of road conditions. These devices play a crucial role in the collection of video data. Researchers and practitioners evaluate road conditions and develop methods for monitoring and identifying road surface problems.

The CNN is a type of bio-inspired neural network specifically engineered to replicate human vision and identify objects. CNN is primarily employed for addressing image-related issues. The fundamental principle of CNN is the employment of convolutional layers to extract characteristics from images. The resultant model is qualified to make precise predictions. CNNs differ from neural networks (NN) by their ability to effectively handle complex multiple datasets, especially ones consisting of images. This approach efficiently mitigates the issue of data variation, wherein the model encounters difficulties in predicting unseen data. CNN surpasses NN in image classification tasks. The essential components of a CNN consist of the convolutional layer, pooling layer, and fully connected layer. The fundamental concept of CNN is the convolutional layer, which is tasked with feature extraction, including the detection of object edges. CNN utilizes complex mathematical methods and the principle of spatial convolution for image processing. Feature extraction is executed by filters or kernels, each designed to extract a particular feature of interest. The utilization of several filters further enhances the network's capabilities. The convolution process in a CNN produces smaller matrices as outputs. Thereafter, the pooling layer extracts significant information and improves data processing efficiency. There are two varieties of pooling: i) max pooling, which identifies the maximum value within each grid, and ii) average pooling. The fully-connected layer constitutes the concluding element of the CNN architecture. It links the output from the pooling and convolution layers.

The convolution layer often produces a three-dimensional volume, whereas a fully-connected layer necessitates a one-dimensional vector [13]-[15]. Consequently, the output of the pooling layer is transformed into a vector prior to entering the fully-connected layer. Moreover, dropout is a method employed to regularize convolutional neural networks and mitigate overfitting concerns. It randomly sets a proportion of neuron outputs during training. Ensembles can reduce overfitting by averaging the outputs of several models; nonetheless, they are resource-intensive, time-consuming, and involve the management of multiple models [3], [15].

Numerous studies have employed deep learning methodologies to identify abnormal objects in videos. A prevalent method involves the use of deep convolutional neural networks (DCNNs) [16] with the Google TensorFlow object detection (GTOD) API. The study examined five categories of road conditions: i) smooth road, ii) uneven road, iii) pothole, iv) incline, and v) bump. Moreover, the fuzzy algorithm can be utilized to determine the speed limit on the highway [17]. A comparative study was performed comparing features derived from three axes and those from a single axis. The research employed SVM, decision trees, and neural networks for classification purposes. The image processing pipeline encompassed labeling, filtering, and feature extraction. Deep neural networks were utilized to categorize the road conditions. The data collection utilized three vehicle types: i) Ford Focus Sedan, ii) Ford Focus Hatchback, and iii) Subaru Outback SUV. The video recordings were obtained with an iPhone 6. The primary focus of the road surface study was the identification of potholes [7].

Wiratmoko et al. conducted a study to identify potholes on the roadway. The specific criteria are a diameter exceeding 10 centimeters and a depth of at least 5 centimeters. The wrapping and cropping techniques are utilized for object detection, and a CNN based on LeNet5 may produce the model. The findings indicated an accuracy of 92.8% [12]. In a separate study, a pothole detecting system utilizing a mixture of Gaussians (MoG) combined with an SVM model and faster R-CNN. In the MoG model, linear SVM and radial basis function SVM (RBF-SVM) were utilized. Nonetheless, the study of the video data indicated that the MoG + SVM was inappropriate. Conversely, faster RCNN demonstrated superior performance in vehicle recognition at night. A deep belief network (DBN) is utilized to develop the function for assessing global road safety performance (SFP). This model can forecast the frequency of accidents across diverse locations. The outcome of enhanced performance is a reduction of the maximum error rate to 23.15% [18]. The refined mask R-CNN (RM R-CNN) utilizes an end-to-end learning approach for the acquisition of real-time road pictures. To identify the traffic signs for the comparison of fast R-CNN and mask R-CNN [19]. The random forest regressor (RFR), multioutput regressor (MOR), and artificial neural network (ANN) can be designed to assess movement at intersections. The analysis of the correlation between

volumes and compatible turning movements [20]. The assessment of pavement performance incorporates various criteria, including base type, hot mix asphalt concrete (HMAC) layer thickness, base layer thickness, traffic load, age, and climatic conditions such as precipitation and temperature. This study employs various statistical methodologies, including generalized linear models (GLM), binary logistic regression (BLR), and random forest (RF). The long-term pavement performance program (LTPP) excels in data analysis. Furthermore, the performance of regression models (GLM and BLR) is assessed in relation to RF by the cross-validation method. The findings reveal the importance of design characteristics, such as cutting, roughness, and alligator cracking [21].

Mukesh *et al.* [21] identified cracks and potholes in the road by comparing Random Forest and CNN methodologies. The demonstrated accuracy is 96% [22]. Road detection may identify features from several image sources, including satellite imagery. The model can be constructed using ANN, CNN along with morphological enhancement and segmentation techniques [23]-[25]. Research has been conducted to mitigate shadows through a shadow-invariant feature space [26]. Another form of image utilized for road detection is very high resolution (VHR) remote sensing. This image format is suitable for segmentation [27]. The aerial image can be utilized to produce a map based on semantic segmentation. An unmanned aerial vehicle (UAV) is employed to autonomously create the monitoring road area based on rapid homography, utilizing a 2D Bird's eye view (BEV) image to recognize roads, lanes, and driver behavior, yielding favorable results [28]-[30]. The recurrent convolutional neural network U-Net (RCNN-UNet) is utilized for road detection and centerline extraction [31]. The multi-centered hough forest (MCHF) method can be employed for curvilinear detection in images as a replacement to euclidean distance [32]. Regarding color road detection, enhancement can be achieved by utilizing illuminant intrinsic images to produce a pixel-level confidence map [33]. The detection of line segmentation is accomplished by the probabilistic Hough transform, which delineates the road section in aerial images utilizing a histogram [34], [35]. The road boundary can be extracted using the s-FCN-loc approach, which operates 30% faster than FCN [36]. An autonomous detection system was implemented to identify diverse road conditions, such as potholes, street bumps, and driver behaviors, utilizing a combination of long short-term memory (LSTM) neural networks and a reservoir computing (RC) model. This approach achieves an accuracy of 0.98 in distinguishing between potholes and their absence [37].

Previous research has introduced multiple methodologies for evaluating road surface conditions. Pre-processing processes encompass activities such as labeling, filtering, and feature extraction. Deep learning models such as YOLO V2, YOLO V3, and up to YOLO V8 are frequently utilized for this objective. Images can be collected utilizing cellphones and video cameras.

3. METHODS

This section explains the process of this paper. There are four subsections. Subsection 3.1 discusses the proposed framework outlines the utilized framework. Subsection 3.2 Data collection specifies the methodology for data acquisition. Subsection 3.3. Data preprocessing outlines the preliminary procedures before model development. And Subsection 3.4. CNN analyzes the historical evolution of CNN and each version of YOLO up to YOLO V8

3.1. Proposed framework

This section presents a framework designed for the identification of road surface conditions. Figure 1 demonstrates the framework. This enables three stages for evaluating road surface conditions with video data. The procedure includes data collection, data pre-processing, and model generation. The framework initiates by gathering data and employing a smartphone's camera to record video data. The frame extraction involves converting the movie into several pictures. Upon completion of the data collection and frame extraction phase, the next stage is data pre-processing. Prior to commencing this phase, data augmentation is required. This aims to improve the data quality. Diverse image processing techniques are utilized to guarantee the data is formatted correctly for further processing. These mechanisms encompass lane detection and object detection. This study concentrates on the vehicular lane and subsequently the lane detecting procedure. Subsequent to the enhancement of picture labeling, this phase delineates the object from the backdrop, a crucial step for enabling the machine to recognize and differentiate the object. The outcome of the labeling operation is an XML file employed to construct the model utilizing the CNN.

3.2. Data collection

The data on road condition, particularly about asphalt roads, is utilized to develop the classifier. Data collection conducted by smartphones from May 2020 to September 2021 in Phuket and Bangkok,

Thailand. A bracket was utilized to hold the smartphone tightly in the vehicle, as seen in Figure 2. An application was employed to save and directly upload the collected data to the cloud.

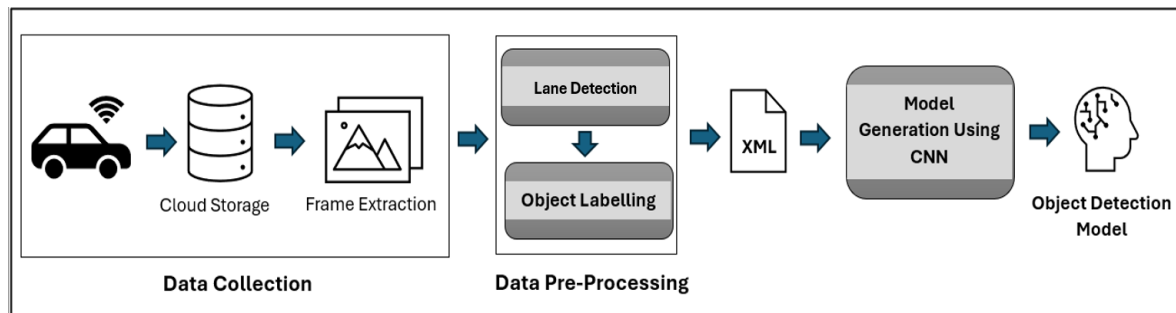


Figure 1. The road surface conditions detection framework



Figure 2. The setup of equipment

Subsequently, frame extraction is executed to obtain images from the video, with each image scaled to 1920×1080 pixels. The dataset comprises 10,984 images designated for training and 2,198 images allocated for testing. Table 1 presents comprehensive information concerning the videos and the procedure of frame extraction. Figure 3 provides an illustration of a custom dataset.

Table 1 illustrates the frame extraction statistics. A total of 134,176 seconds resulted in the extraction of 1,341,760 frames. Among these frames, every three frames will select one frame. Then, 24,276 contained objects were chosen for generating the object detection model. And the frame extraction code is shown in Algorithm 1. The video file is designated as 'videocap', the image count is referred to as 'count', and the state of the video read is indicated by the parameter 'success' (lines 2-4). Subsequently, the recorded video is analyzed, and the count parameter is executed (lines 7-8). The image is thereafter saved to the directory after every three images (lines 9-12), constituting the output.

Table 1. The intricacies of video and frame extraction

Video	Time(s)	All Frame	Frame with object	Selected frame
125	134,176	1,341,760	32,803	24,276

3.3. Data preprocessing

Following data collection using a smartphone application, frame extraction is performed. This section addresses data pre-processing, comprising three subsections. Subsubsection 3.3.1 addresses the category of data. Next 3.3.2 constitutes a component of the lane detecting procedure. Subsubsection 3.3.3 delineates the specifics of object labeling, a crucial procedure prior to data input for model generation.



Figure 3. A demonstration of a custom dataset

Algorithm 1. Frame extraction

```

1: Input:
2:   videocap = The video file
3:   count    = The image count
4:   success  = The status of the video read
5: Output:
6: while success do
7:   success = videocap.read()
8:   count   = count + 1
9:   If count mod 3 = 0
10:    If success = true
11:      cv2.imwrite("image path" % count, image, [cv2.IMWRITE_JPEG_QUALITY, 100])
12:    end if
13:  end if
14: end while

```

3.3.1. Data categories

This study identifies three specific objects for inspection by the Department of Highways in Thailand. These objects include road damage, classified as structural failure and functional failure. The structure failure category encompasses potholes and cracks, whereas manholes are classified under the functional failure category. The impacts of functional failure relate to concerns of convenience and safety. Figure 4 illustrates an example of these things. Figure 4(a) depicts a crack, Figure 4(b) illustrates a pothole, and Figure 4(c) showcases a manhole cover. An example of a custom object is illustrated in Figure 5.

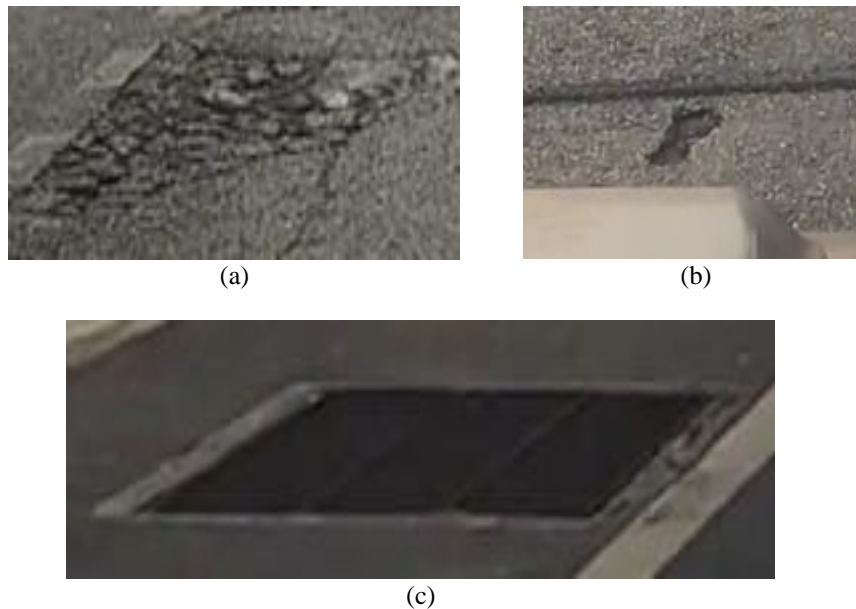


Figure 4. An illustration of the three objects (a) an example of a crack (b) an example of a pothole, and (c) an example of a manhole

3.3.2. Lane detection

Following the collection of video data, including three object types, the frame extraction process commences. The frame extraction technique involves converting a video collection into several images. Improvements were implemented in the lane detection procedure, particularly focusing on identifying lanes for vehicular driving. The lane detecting phase is essential, including a sequence of actions. Initially, Canny Edge detection is implemented, succeeded by area segmentation, and ultimately, the application of the Hough Transform. Figure 6 illustrates the results, while Algorithm 2 displays the lane detecting code.

Algorithm 2 elevates the image collection to the parameter 'frame' (line 2). Subsequently, implement a Canny method to identify edges within an image. The segmentation is executed to produce the pixel region in an image. Utilize the Hough transform to identify the shape (lines 4-6). The final step involves calculating the line and seeing the output (lines 7-10).

Algorithm 2. Lane detection

```

1: Input:
2: frame = The collection of read image files
3: Output:
4: canny      = do_canny(frame)
5: segment    = do_segment(canny)
6: hough      = HoughLinesP(segment, 2, minLineLoth = 100, maxLineGap = 50)
7: lines      = calculate_lines(frame, hough)
8: line_visualize = visualize_lines(frame, lines)
9: output     = addWeighted(frame, 0.9, line_visualize, 1, 1)
10: print output

```

Subsequent to the acquisition of video data, the videos were analyzed to extract discrete pictures. The initial dimensions of these photos were 1920×1080 pixels. For the purposes of our investigation, we resized the photos to 1280×720 pixels. To eliminate unwanted environmental components, we excised 360 pixels from the top and 320 pixels from both the left and right sides of the photos. Figure 7 depicts the original image along with the specified crop region. The designated area of the image, marked by the red rectangle, was utilized for subsequent analysis and processing in this study.

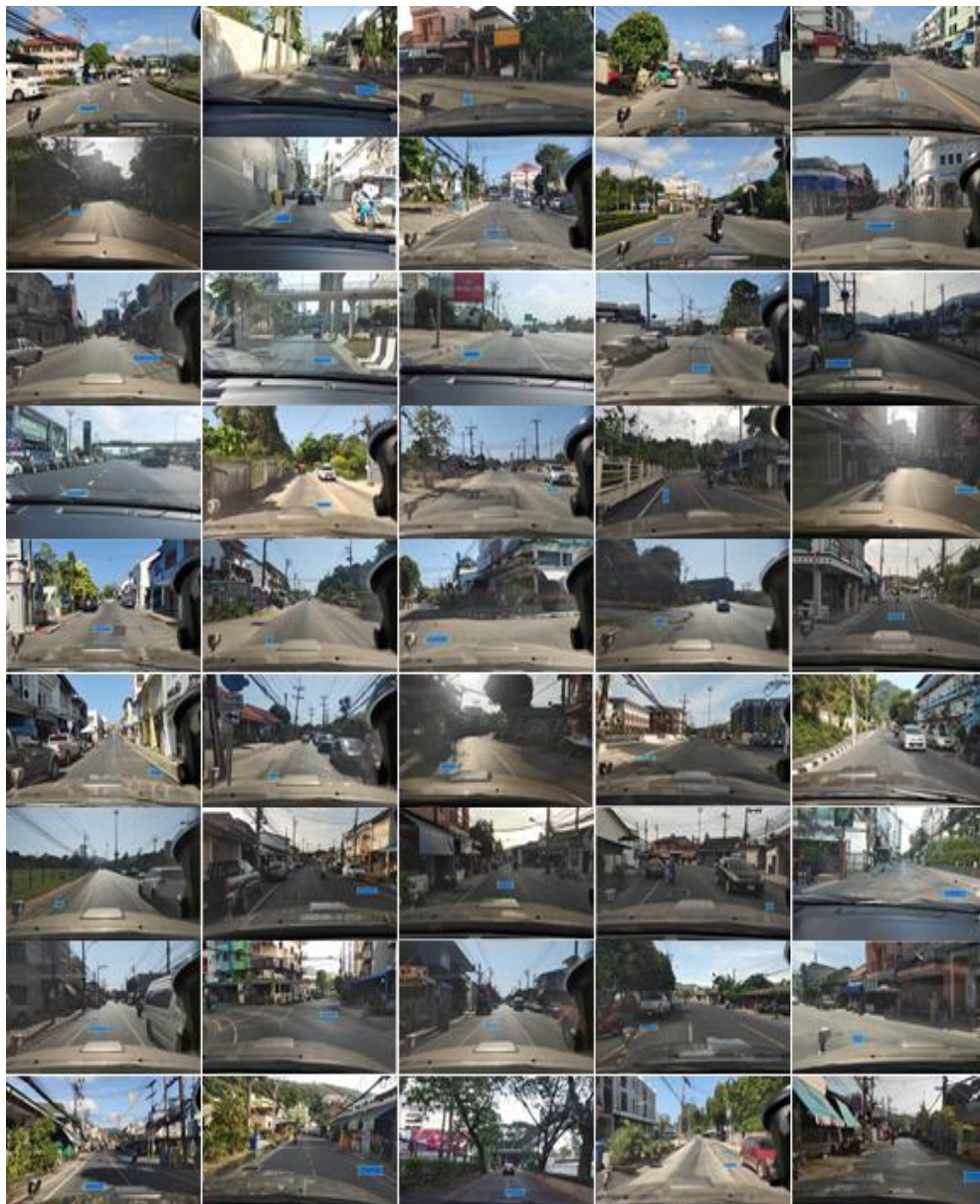


Figure 5. A demonstration of a custom object



Figure 6. The example of lane detection



Figure 7. The image area

3.3.3. Object labelling

The frame extraction procedure was enhanced following the collection of the video data. The subsequent phase entails object labeling, executed with LabelMe, LabelImg, and Laberu in this study. These tools exhibit varying usage methodologies, with Laberu being notably user-friendly and able to support many users concurrently. The labeling findings from Laberu are recorded in .xml format, which can be immediately employed in later stages. For instance, Figures 8 to 10 illustrate the annotated instances of the three objects.

The data for this investigation is gathered utilizing smartphones mounted on the car's windshield. Thus, the data collection is confined to the region right ahead of the vehicle. This research primarily focuses on objects situated four meters from the vehicle. The items are designated with rectangles that must conform to the frame. In the context of adjacent frames, the rectangles ought to intersect as little as feasible.

As illustrated in Figures 8-10, each object has multiple different characteristics. A manhole cover can possess both circular and rectangular forms. Table 2 summarizes the object labeling results, revealing a total of 17,391 photos of cracks, 14,606 images of manholes, and 3,573 images of potholes in sequential sequence.



Figure 8. Illustration of a labeled crack



Figure 9. Illustration of a labeled pothole



Figure 10. Illustration of a labeled manhole

Table 2. The count of object labeling

Crack	Pothole	Manhole
17,391	3,573	14,606

3.4. Convolutional neural network

The convolutional neural network (CNN) originated in the 1980s, invented by Yann LeCun. The first architecture is LeNet, which was utilized for digital recognition jobs. The architecture comprises convolutional layers, pooling layers, and a fully linked layer. The subsequent architectures are AlexNet, ImageNet, VGGNet, GoogleNet, and ResNet, in that order. In object detection, the CNN can identify and locate several things inside an image, a task that is more complex than classification tasks. YOLO, an acronym for "You Only Look Once," is a widely utilized object identification technique developed by Joseph Redmon [38]. This can efficiently detect many items. YOLO performs this by utilizing a complex grid framework in each layer. The technique involves partitioning the image into a narrow window of size $N \times N$ and employing an algorithm to anticipate the object. These procedures utilize deep learning. YOLO V2 (or YOLO9000) was created to improve on YOLO V1, which identified objects in real-time. Darknet19 serves as the foundation for YOLO V2 [39].

YOLO V3 has developed from various architectures, including ResNet and feature-pyramid network (FPN). The feature extraction technique in YOLO V3 utilizes Darknet53, a deep neural network comprising 53 layers. Initially, Darknet53 was trained using the Imagenet dataset. The 106 layers and features were

integrated into the YOLO V3 architecture. This method can enhance model performance in detecting complex features and improve detection efficacy. The FPN and U-Net architectures exhibit similarities. These are utilized for the prediction of several variables across various boxes, which is time-intensive. The data can transition from bigger to smaller scales and, alternately, function effectively in high semantic content inside high-resolution data [40]. Following the development of YOLO V3, the latest iteration is YOLO V4. YOLO V4 is derived on YOLO V3.

The architecture of YOLO V4 comprises a head section that delineates the bounding box and forecasts the object classes. The foundation of YOLO V4 is CSPDarknet53, which serves as the primary layer for feature extraction. The neck component incorporates spatial pyramid pooling (SPP) and Path aggregation network (PAN) modules. This component utilizes pooled geographic data and computes features across various scales and tiers of the network to enhance model performance. YOLO V4 has enhanced efficiency in GPU processing utilization. This renders the process time-consuming while enhancing accuracy, which is the benefit of YOLO V4. The model employs parallel computations to execute suitable tasks for object detection [41]. The YOLO V4 method demonstrates the performance of small item detection, suitability for big inputs, and the incorporation of additional layers. The model can detect objects of various sizes due to the numerous characteristics of YOLO V4.

YOLO V5 is a CNN architecture derived from YOLO V3 and implemented in PyTorch. This version modified the input sizes, depth, and width of the network. YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x represent distinct variants within this architecture. The primary benefit of YOLO V5 is its rapid image detection and operational efficiency. The image is segmented into multiple small grids, with each grid cell undergoing analysis. The foundation of YOLO V5 is CSPDarknet, functioning as a feature extractor. The neck component is PSNet, which is tasked with feature fusion. The network's head use YOLO to manage object outcomes. The output comprises class, score, location, and size data. A deep learning model often comprises multiple components: input, backbone, back, neck, head, and output. The backbone executes feature extraction, yielding a feature map. The neck portion is utilized in scenarios necessitating intricate feature extraction. The head produces the final output.

YOLO V6, created by the Meituan Vision AI Department in China and launched in 2022, enhances the original YOLO architecture, which is a single-stage YOLO algorithm. This iteration of YOLO was explicitly developed to tackle actual challenges in industrial applications. The EfficientRep functions as the backbone of the model. The Rep-PAN Neck enhances hardware performance, leading to enhanced overall efficacy. The model's backbone and neck are constructed based on the RepVGG architecture, which serves as its foundation [42]. Furthermore, a decoupled head is utilized, which streamlines the design of the head section for enhanced implementation ease. YOLO V6 surpasses YOLO V5 in accuracy and speed.

YOLO V7, created by the YOLOR team, is an advancement of YOLO V4. This iteration of YOLO exists in multiple variants, including YOLOv7, YOLOv7-tiny, and YOLOv7-W6, each presenting unique attributes and features [43]. YOLO V7's design integrates the Extended Efficient Layer Aggregation Network (E-ELAN) and utilizes Model Scaling for concatenation-based models. Additionally, YOLO V7 employs the Freebies feature, commonly referred to as bat-of-freebies. The anchor box configuration in YOLO V7 has been revised to include nine anchor boxes. This modification allows the model to proficiently identify items of varying forms and sizes. These modifications substantially enhance the speed and accuracy attained by YOLO V7 [44].

In 2023, the Ultralytics team, recognized for YOLO V5, significantly improved YOLO V3, resulting in YOLO V8. This modern model improves performance and adaptability, allowing it to manage many tasks, including classification, detection, pose estimation, segmentation, and tracking. YOLO V8 is applicable in various fields and can be employed in numerous applications. YOLO V8 utilizes the C2f module, similar to YOLO V5. The incorporation of advanced characteristics and contextual information in this module designates it as an essential element of the system, augmenting precision. Feature fusion is utilized to amalgamate features from many layers, facilitating a more thorough depiction. In the head section, YOLO V8 employs either anchor-based or anchor-free techniques to forecast bounding boxes and class probabilities. The model utilizes complete intersection over union (CIoU) and distribution focal loss (DFL) loss functions to measure the differences between predicted and actual values, facilitating training and optimization operations [43].

The object detection procedure involves both predicted and ground truth data. The correctness of a bounding box is determined by the intersection over union (IoU) metric. The precision and recall metrics are assessed using IoU. Object detection use the mean average precision (mAP) metric for evaluation measurement. The precision is computed by adjusting the threshold value until the optimal result is attained. When the model threshold yields several precision values, the mAP provides a singular score that reflects the model's overall performance. Figure 11 illustrates the predicted and ground truth bounding boxes, with the IoU represented by the yellow bounding box and the area indicated by the orange arrow.



Figure 11. An example of a prediction and ground truth bounding box. The ground truth box is represented in blue, while the prediction bounding box is depicted in red

The mAP calculation is obtained from the evaluation of model prediction scores. The scores are subsequently converted into class labels, and a confusion matrix is generated, consisting of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Precision and recall are then calculated. Subsequently, the actual values are employed to calculate the area under the curve for average precision (AP). The mAP is ultimately computed utilizing (1).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (1)$$

mAP incorporates FP and FN alongside precision and recall, rendering it an appropriate indicator for performance evaluation in this investigation.

4. RESULT AND DISCUSSION

We performed a comparative analysis of object detection models developed with conventional CNN architectures. Specifically, evaluated the efficacy of YOLO V3, YOLO V4, YOLO V5, YOLO V6, YOLO V7, and YOLO V8 in identifying road conditions. The aim was to determine the classification and location of objects inside the images. The evaluation criteria for model comparison were speed and mAP (mean Average Precision) over four different thresholds. They utilized a computer equipped with an Intel(R) Core(TM) i5-9300H CPU operating at 2.40 GHz and 12 GB of RAM to do the research. The software stack comprised Miniconda3 and Python 3.6. Typically, during model training, the issue of overfitting reduces the model's performance. Numerous strategies exist to address the overfitting issue, including augmentation, dropout, weight decay, and batch normalization, among others. The methodologies employed to replicate the model in this study are dropout regularization and Batch normalization. This study proposes three experiments. The YOLO version is presented in subsection 4.1. Batch normalization is addressed in subsection 4.2. Dropout regularization is illustrated in subsection 4.3.

4.1. YOLO version

Among the evaluated models, greater performance was indicated. Comprehensive information about the image dataset, including size and other related information, is addressed in subsubsection 3.3.1. The experimental configuration utilized original images derived from videos, which were then resized. The images utilized for testing possessed a resolution of 1280x720 pixels. The outcomes of YOLO V5 to YOLO V8 with a batch size of 8 and 100 epochs indicate that the mAP for the models was 74.50%, 86.19%, 61.30%, and 73.82%, respectively. Table 3 indicates that the highest mAP value is 86.19% for YOLO V6. YOLO V5 exhibits a mAP value of 74.50%, YOLO V8 has a mAP value of 73.82%, and YOLO V7 demonstrates a mAP value of 61.30%. The mAP and loss values for each of the 10 epochs are illustrated in Figures 12 and 13.

The mAP values of YOLO V5 and YOLO V8 are similar. YOLO V6 has superior performance, whereas YOLO V7 ranks below other YOLO versions. The loss graph indicates that YOLO V5 and YOLO V7 are identical. YOLO V8 represents the top of the graph, followed sequentially by YOLO V6, YOLO V5, and YOLO V7. Figure 14 illustrates the successful detection of three objects in the video.

Table 3. The result of three objects detection

Architecture	mAP (%)	Loss
YOLO V5	74.50	0.0463
YOLO V6	86.19	0.6113
YOLO V7	61.30	0.0241
YOLO V8	73.82	1.0036

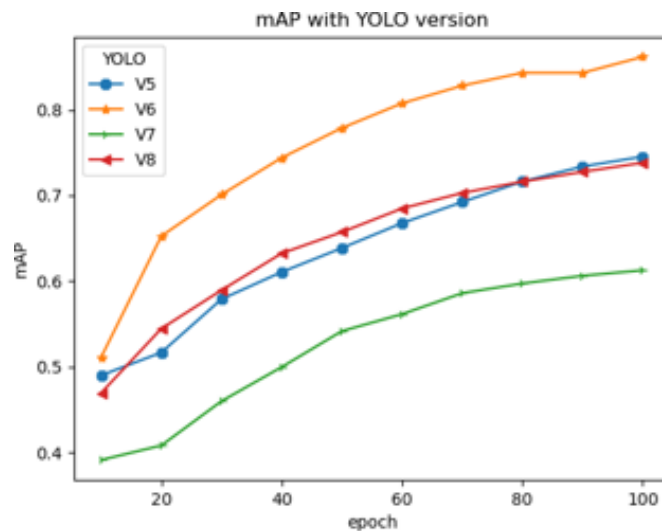


Figure 12. mAP value obtained during training with each YOLO version

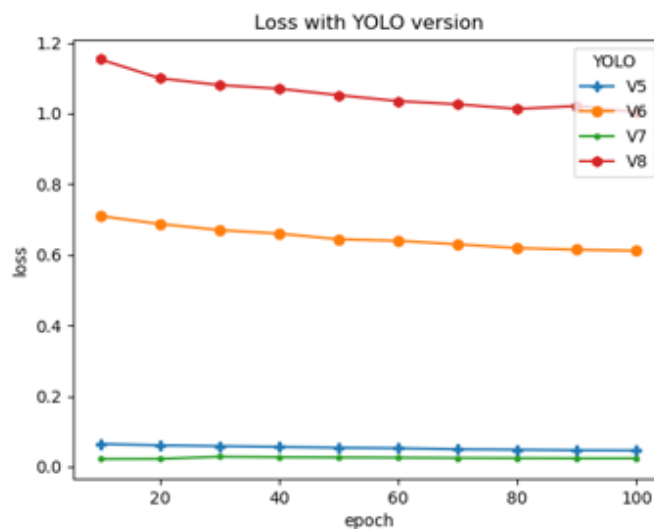


Figure 13. Loss value obtained during training with each YOLO version

4.2. Batch normalization

Batch normalization is a method employed to enhance the efficacy of models in deep learning tasks. This method could reduce the issue of overfitting. In this task, modify the batch to include four values. The image dimensions are 480x480, processed with YOLO V6 over 100 epochs. The outcome is presented in Table 4. Three objects identified by the YOLO V6 model with a batch size of 32 achieved a mAP of 87.38%, the highest recorded value. The subsequent batch has mAP values of 86.98%, 86.19%, and 82.70% for 16, 8, and 4, respectively. Batch 16 exhibits a loss value of 0.6075, which is lower than the loss value of 0.6091 for batch 32. Figure 15 illustrates the graph of the mAP value during training with YOLO V6 and different batch sizes. Figure 16 illustrates the graph of loss values during training with YOLO V6 at various batch sizes.

As illustrated in Figure 15, the various batches exhibit similar results. YOLO V6 batch 4 displays a decreased performance (shown by the blue line) compared to the other three lines. Batch 32 exhibits the greatest mAP value. Batch 4 exhibits the highest loss value (blue line). Additionally, the other three lines are likely as illustrated in Figure 16. Subsection 7.3 discusses the implementation of dropout regularization in YOLO V6 with a batch size of 32.

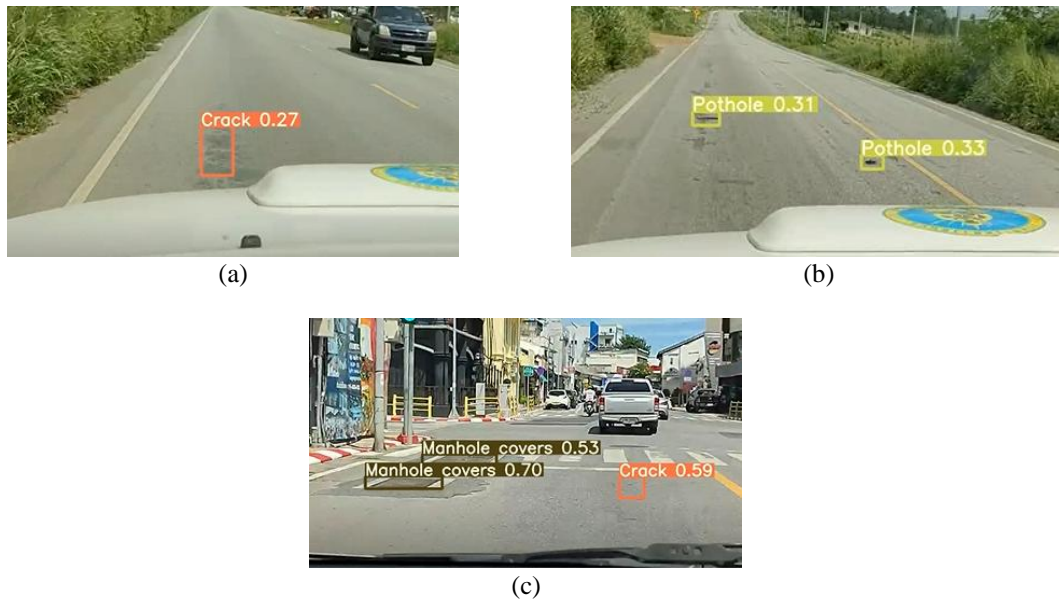


Figure 14. YOLO V6 successfully detects three objects in a new video; (a) crack detected, (b) pothole detected, and (c) manhole and crack detected

Table 4. The result of batch normalization on YOLO V6

Batch size	Loss	mAP (%)
4	0.6254	82.70
8	0.6113	86.19
16	0.6075	86.98
32	0.6091	87.38

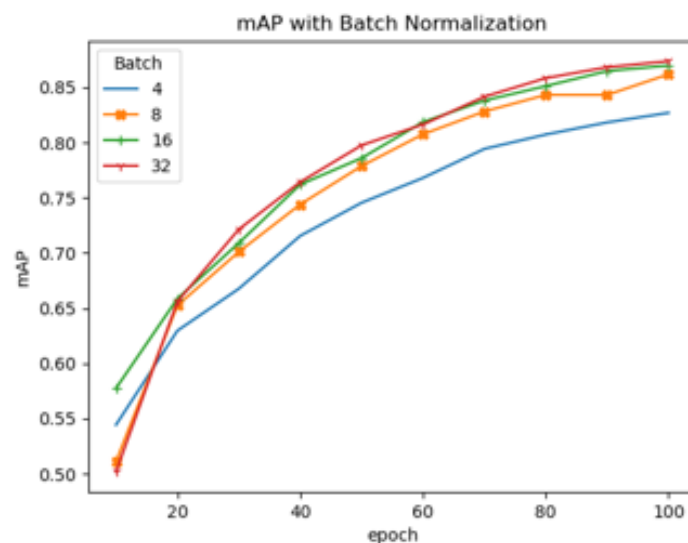


Figure 15. mAP value obtained during training with YOLO V6 with different batch size

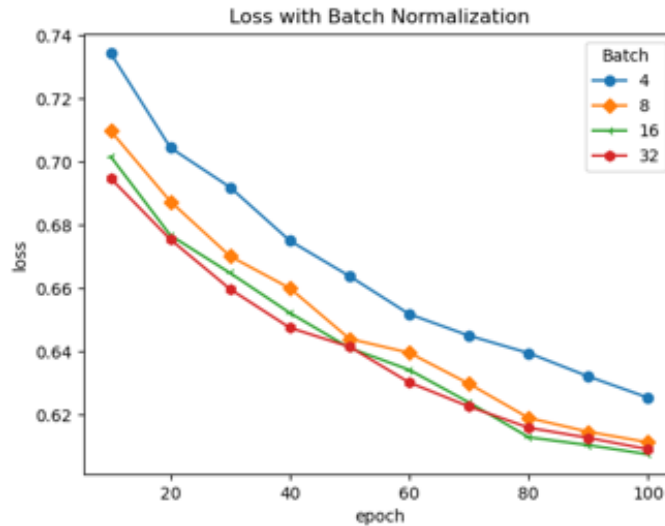


Figure 16. Loss value obtained during training with YOLO V6 with different batch size

4.3. Dropout regularization

Dropout regularization functions within the model by randomly deactivating some nodes during the training phase. This technology is time-consuming, saves resources, and requires no maintenance. This model was built using YOLO V6 with a batch size of 32, 100 epochs, and an image resolution of 480×480 . Table 5 presents the experimental results with dropout rates of 0.25, 0.50, 0.75, and 1.00.

Table 5 presents similar experimental results for dropout rates of 0.25, 0.50, and 0.75. The mAP values are 85.40% at a dropout of 0.25, 85.73% at a dropout of 0.50, and 85.15% with a dropout of 0.75. For dropout, 1.00 exhibits the lowest mAP value of 39.24%. The loss values for dropout rates of 1.00, 0.75, 0.50, and 0.25 are 0.6159, 0.6146, 0.6127, and 0.6114, respectively. Figure 17 illustrates a graph displaying the mAP value during training with YOLO V6 and different dropout rates. Figure 18 illustrates a graph showing the loss value during training with YOLO V6 and different dropouts.

Table 5. The result of Dropout Regularization on YOLO V6

Dropout	Loss	mAP (%)
0.25	0.6114	85.40
0.50	0.6127	85.73
0.75	0.6146	85.15
1.00	0.6159	39.24

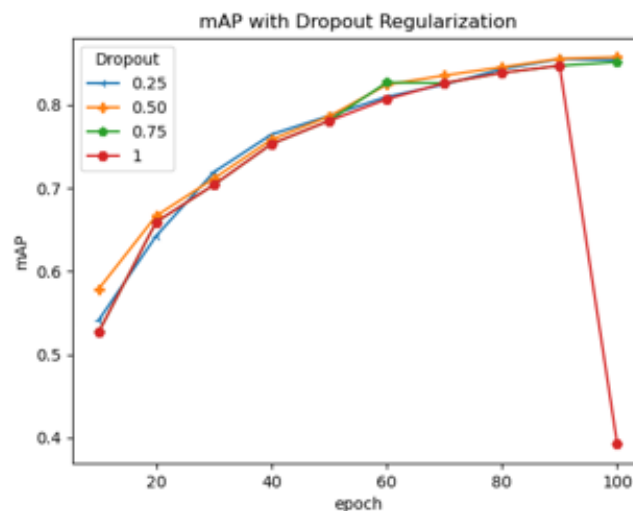


Figure 17. mAP value obtained during training with YOLO V6 with different dropout

The result illustrates the varying mAP value in dropout 1 (red line). The performance of dropout rates 0.25, 0.50, and 0.75 is similar. Figure 18 illustrates a loss graph containing four lines representing various dropout. Dropout 0.25 has a lower loss value compared to dropout rates of 0.50, 0.75, and 1.00.

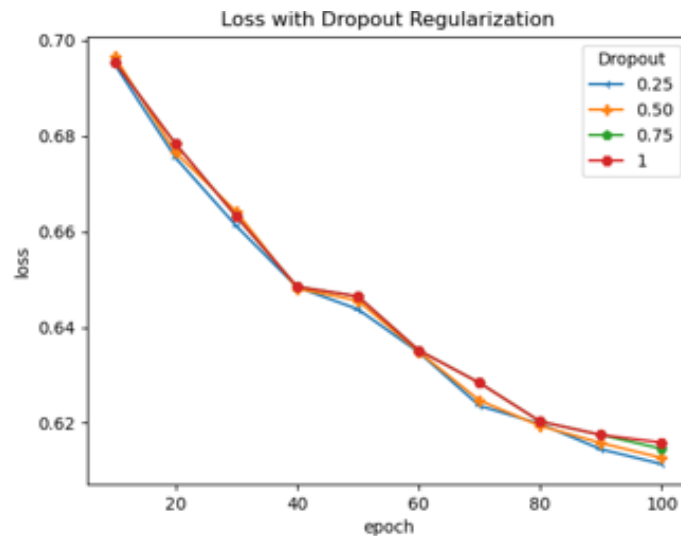


Figure 18. Loss value obtained during training with YOLO V6 with different dropout

5. CONCLUSION

The paper presents a comparison for detecting cracks, potholes, and manhole covers with video data collected via smartphones. This study presents various prior studies, the methodology, details of CNN, evaluation, and the experiments conducted. The experimental object detection comparison indicates that YOLO V8 is compared with YOLO V5, but YOLO V6 exhibits the highest mAP. Subsequently, compare the hyperparameters of YOLO V6, specifically emphasizing batch size and dropout. The performance of the proposed method improves the mAP. In conclusion, the paper presents a comparison of four different versions of YOLO, including details on batch size and dropout parameters. Acknowledgements reveal the supporters of this project.

ACKNOWLEDGMENT

This work was supported by the Digital Science for Economy, Society, Human Resources Innovative Development and Environment project funded by Reinventing Universities & Research Institutes no. 3674774, Ministry of Higher Education, Science, Research and Innovation, Thailand.

FUNDING INFORMATION

This work was supported in part by the Artificial Intelligence Innovation Laboratory, College of Computing, Prince of Songkla University, Phuket Campus, Thailand.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sujittra Sa-ngiem		✓	✓	✓	✓			✓	✓	✓	✓		✓	
Kwankamon Dittakan	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓		
Sarocho Boonsiripant	✓					✓	✓						✓	✓

C : Conceptualization	I : Investigation	Vi : Visualization
M : Methodology	R : Resources	Su : Supervision
So : Software	D : Data Curation	P : Project administration
Va : Validation	O : Writing - Original Draft	Fu : Funding acquisition
Fo : Formal analysis	E : Writing - Review & Editing	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [Sujittra Sa-ngiem], upon reasonable request.



REFERENCES

- [1] H. E. Rosen *et al.*, "Global road safety 2010–18: An analysis of global status reports," *Injury*, Jul. 2022, doi: 10.1016/j.injury.2022.07.030.
- [2] S. B. Kharat and S. D. Sawant, "Travolution: system for road safety," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2017, pp. 1123–1125, doi: 10.1109/ICCONS.2017.8250641.
- [3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [4] P. Srisurin and S. Chalermpong, "Analyzing human, roadway, vehicular and environmental factors contributing to fatal road traffic crashes in Thailand," *Engineering Journal*, vol. 25, no. 10, pp. 27–38, Oct. 2021, doi: 10.4186/ej.2021.25.10.27.
- [5] R. Mkwata and E. E. M. Chong, "Effect of pavement surface conditions on road traffic accident - A Review," *E3S Web of Conferences*, vol. 347, p. 01017, 2022, doi: 10.1051/e3sconf/202234701017.
- [6] T. Huang, P. Chakraborty, and A. Sharma, "Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images," *arXiv (Cornell University)*, Jan. 2020, doi: 10.48550/arxiv.2005.04188.
- [7] A. Basavaraju, J. Du, F. Zhou and J. Ji, "A machine learning approach to road surface anomaly assessment using smartphone sensors," in *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2635–2647, 2020, doi: 10.1109/JSEN.2019.2952857.
- [8] A. Shtayat, S. Moridpour, B. Best, and M. Abuhassan, "Using supervised machine learning algorithms in pavement degradation monitoring," *International Journal of Transportation Science and Technology*, Oct. 2022, doi: 10.1016/j.ijst.2022.10.001.
- [9] Q. Shi and X. Li, "Road detection from remote sensing images by generative adversarial networks," *IEEE Access*, vol. 6, pp. 25486–25494, Jan. 2018, doi: 10.1109/access.2017.2773142.
- [10] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang, and X. Huang, "Road detection and centerline extraction via deep recurrent convolutional neural network U-Net," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 7209–7220, Sep. 2019, doi: 10.1109/TGRS.2019.2912301.
- [11] Z. Zhang, Q. Liu and Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, May 2018, doi: 10.1109/LGRS.2018.2802944.
- [12] Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional neural networks-based potholes detection using thermal imaging," *Journal of King Saud University - Computer and Information Sciences*, Feb. 2019, doi: 10.1016/j.jksuci.2019.02.004.
- [13] A. D. Wiratmoko *et al.*, "Design of potholes detection as road's feasibility data information using convolutional neural network (CNN)," in *2019 International Symposium on Electronics and Smart Devices (ISESD)*, Badung, Indonesia, 2019, pp. 1–5, doi: 10.1109/ISESD.2019.8909461.
- [14] J. Li, C. Nguyen and S. You, "Temporal 3D fully connected network for water-hazard detection," *2019 Digital Image Computing: Techniques and Applications (DICTA)*, Perth, WA, Australia, 2019, pp. 1–5, doi: 10.1109/DICTA47822.2019.8945849.
- [15] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiya, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, Jun. 2018, doi: 10.1111/mice.12387.
- [16] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, pp. 9243–9275, Aug. 2022, doi: 10.1007/s11042-022-13644-y.
- [17] G. Lanzaro and M. Andrade, "A fuzzy expert system for setting Brazilian highway speed limits," *International Journal of Transportation Science and Technology*, vol. 12, no. 2, pp. 505–524, Jun. 2023, doi: 10.1016/j.ijst.2022.05.003.
- [18] G. Pan, L. Fu, and L. Thakali, "Development of a global road safety performance function using deep neural networks," *International Journal of Transportation Science and Technology*, vol. 6, no. 3, pp. 159–173, Sep. 2017, doi: 10.1016/j.ijst.2017.07.004.
- [19] R. K. Megalingam, K. Thanigundala, S. R. Musani, H. Nidamanuru, and L. Gadde, "Indian traffic sign detection and recognition using deep learning," *International Journal of Transportation Science and Technology*, Jun. 2022, doi: 10.1016/j.ijst.2022.06.002.
- [20] K. Shaaban, A. Hamdi, M. Ghanim, and K. B. Shaban, "Machine learning-based multi-target regression to effectively predict turning movements at signalized intersections," *International Journal of Transportation Science and Technology*, Mar. 2022, doi: 10.1016/j.ijst.2022.02.003.
- [21] S. Bhandari, X. Luo, and F. Wang, "Understanding the effects of structural factors and traffic loading on flexible pavement performance," *International Journal of Transportation Science and Technology*, vol. 12, no. 1, Mar. 2022, doi: 10.1016/j.ijst.2022.02.004.
- [22] K. A. Vinodhini and K. R. A. Sidhaarth, "Pothole detection in bituminous road using CNN with transfer learning," *Measurement: Sensors*, vol. 31, p. 100940, Feb. 2024, doi: 10.1016/j.measen.2023.100940.




- [23] M. Mokhtarzade and M. J. V. Zoej, "Road detection from high-resolution satellite images using artificial neural networks," *International Journal of Applied Earth Observation and Geoinformation*, vol. 9, no. 1, pp. 32–40, Feb. 2007, doi: 10.1016/j.jag.2006.05.001.
- [24] A. Laddha, M. K. Kocamaz, L. E. Navarro-Serment and M. Hebert, "Map-supervised road detection," *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, Sweden, 2016, pp. 118–123, doi: 10.1109/IVS.2016.7535374.
- [25] Q. Wang, J. Gao and Y. Yuan, "Embedding Structured Contour and Location Prior in Siamese Fully Convolutional Networks for Road Detection," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 230–241, Jan. 2018, doi: 10.1109/TITS.2017.2749964.
- [26] J. M. A. Alvarez and A. M. López, "Road detection based on illuminant invariance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 184–193, Mar. 2011, doi: 10.1109/TITS.2010.2076349.
- [27] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, June 2017, doi: 10.1109/TGRS.2017.2669341.
- [28] S. Saito and Y. Aoki, "Building and road detection from large aerial imagery," *Image Processing: Machine Vision Applications VIII*, Feb. 2015, doi: 10.1117/12.2083273.
- [29] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297–309, Feb. 2015, doi: 10.1109/tits.2014.2331353.
- [30] J. Fritsch, T. Kühnl and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands, 2013, pp. 1693–1700, doi: 10.1109/ITSC.2013.6728473.
- [31] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang and X. Huang, "Road detection and centerline extraction via deep recurrent convolutional neural network U-Net," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 7209–7220, Sept. 2019, doi: 10.1109/TGRS.2019.2912301.
- [32] H. Zhang, Y. Yang and H. Shen, "Detection of curvilinear structure in images by a multi-centered hough forest method," *IEEE Access*, vol. 6, pp. 22684–22694, 2018, doi: 10.1109/ACCESS.2018.2823726.
- [33] B. Wang, V. Frémont and S. A. Rodríguez, "Color-based road detection and its evaluation on the KITTI road benchmark," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Dearborn, MI, USA, 2014, pp. 31–36, doi: 10.1109/IVS.2014.6856619.
- [34] S. Zhou, J. Gong, G. Xiong, H. Chen and K. Iagnemma, "Road detection using support vector machine based on online learning and evaluation," in *2010 IEEE Intelligent Vehicles Symposium*, La Jolla, CA, USA, 2010, pp. 256–261, doi: 10.1109/IVS.2010.5548086.
- [35] Yucong Lin and S. Saripalli, "Road detection from aerial imagery," in *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, USA, 2012, pp. 3588–3593, doi: 10.1109/ICRA.2012.6225112.
- [36] D. Chaudhuri, N. K. Kushwaha and A. Samal, "Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 5, pp. 1538–1544, Oct. 2012, doi: 10.1109/JSTARS.2012.2199085.
- [37] B. Varona, A. Montserin, and A. Teyseyre, "A deep learning approach to automatic road surface monitoring and pothole detection," *Personal and Ubiquitous Computing*, vol. 24, no. 4, pp. 519–534, May 2019, doi: 10.1007/s00779-019-01234-z.
- [38] W. Zhiqiang and L. Jun, "A review of object detection based on convolutional neural network," in *2017 36th Chinese Control Conference (CCC)*, Jul. 2017, doi: 10.23919/chicc.2017.8029130.
- [39] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, pp. 9243–9275, Aug. 2022, doi: 10.1007/s11042-022-13644-y.
- [40] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arxiv.org*, Apr. 2018, doi: 10.48550/arXiv.1804.02767.
- [41] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," *arXiv.org*, Apr. 22, 2020. <https://www.arxiv.org/abs/2004.10934v1>
- [42] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding and J. Sun, "RepVGG: making VGG-style ConvNets great again" in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 13728–13737, doi: 10.1109/CVPR46437.2021.01352.
- [43] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: from YOLOv1 to YOLOv8 and beyond," *arXiv (Cornell University)*, Apr. 2023, doi: 10.48550/arxiv.2304.00501.
- [44] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, doi: 10.1109/cvpr52729.2023.00721.

BIOGRAPHIES OF AUTHORS






Sujittra Sa-ngiem   obtained her B.E. degree in Science in Information Technology (IT) from the Faculty of Information and Environment at Prince of Songkla University, Phuket Campus, Thailand in 2009. Later, she completed her M.E. degree in Science in Information Technology (MSIT) from the College of Computing at the same university in 2019. Currently, she holds the position of research assistant and is pursuing her PhD in the College of Digital Science Establishment Project at Prince of Songkla University, located in Songkhla, Thailand. Her research interests primarily lie in Artificial Intelligence (AI), Machine Learning, and Deep Learning. Specifically, she is engaged in research related to deep learning techniques in the field of object detection, with a focus on smooth road surfaces. She can be contacted at email: sujittra.sangiem@gmail.com.



Kwankamon Dittakan    obtained a Doctorate in Computer Science from the University of Liverpool, United Kingdom. She specializes in artificial intelligence, data science, and machine learning, with a particular focus on processing unstructured data such as images, videos, texts, or signals. She currently holds a position as a professor member at the College of Computing, Prince of Songkla University, Phuket Campus. In addition, she is in charge of leading the Artificial Intelligence Innovation Laboratory (AiiLAB). She can be contacted at email: kwankamon.d@phuket.psu.ac.th.



Saroch Boonsiripant    earned a Doctorate in Civil Engineering from the Georgia Institute of Technology in the United States. He is highly skilled in Intelligent Transportation Systems, Transportation Engineering, and smart mobility. He is currently an associate professor and faculty member in the Department of Civil Engineering, Faculty of Engineering, Kasetsart University. He can be contacted at email: saroch.b@ku.th.