RoadSCNet: Road Surface Condition Detection Network

Sujittra Sa-ngiem[1], Kwankamon Dittakan[2]*, Saroch Boonsiripant[3]
College of Digital Science, Prince of Songkla University, Songkhla, Thailand[1]
College of Computing, Prince of Songkla University, Phuket Campus, Phuket, Thailand[2]
Faculty of Engineering, Kasetsart University, Bangkok, Thailand[3]

***Abstract*—The quality of the road is an important issue that contributes to accidents, resulting in the loss of time, resources, and lives. To manually survey the road issue. This is very delayed and costly. Automatic detection of road conditions facilitates surveys more efficiently than human methods. This research identifies three objects: cracks, potholes, and manhole covers. This research shown the highest efficiency with YOLO V6 compared to YOLO V5, V7, and V8. This paper proposes RoadSCNet, designed for YOLOv6 implementations, has been developed for road research. A key part is the customized Horizon block, which enhances horizontal contextual feature extraction efficacy and reduces the limitations of traditional YOLO architecture in identifying road surface condition by long and low light variation, such as cracks and potholes.**

Keywords*—RoadSCNet; road surface; detect road; road condition; deep learning; crack; pothole; manhole cover; image analysis; convolutional neural network; CNN

I. Introduction

Road safety is a critical issue. Road accidents are the primary cause of loss of life and property. Improving monitoring for safety and driver assistance is important. The performance of pavement, covering surface friction and roughness, has been statistically associated with crash frequency and severity, highlighting the necessity for maintenance-based infrastructure design [1],[2]. Image Processing and Artificial Intelligence (AI) are methodologies employed to solve this issue. Object detection is crucial for effectively understanding the road environment. Road object detection is a critical and challenging area of research in computer vision, essential for Intelligent Transportation Systems (ITS), Advanced Driver-Assistance Systems (ADAS), and autonomous driving. The objective of many studies is to ascertain the position and boundaries of objects. For instance, automobiles, bicycles, and people. On the other hand, the inspection of roads and infrastructure is included. An example includes detecting potholes and evaluating road performance, which enhances safety and reduces accidents. Road object detection has numerous problems in real-life situations. For instance, poor brightness, cloudy conditions, and blocking by other objects. The previous method was limited by limitations in two-dimensional image processing techniques. Under situations where the roadway is missing of distinct markers or significantly obstructed. In recent years, deep learning has played a significant role in computer vision tasks. AI is a technology capable of learning, planning, problem-solving, and performing activities similar to those of humans. AI is widely utilized across various fields which enhancing labor efficiency. A dataset is essential for generating the model for each task. For instance, video, image, text, signal, etc. A convolutional neural network (CNN) is a component of deep learning that produces the suitable model for tasks. Typically encompassing categorization and object detection tasks. Examples of categorization in medical applications include generating models to diagnose diabetic retinopathy, identify cerebral microbleeds from MRI scans, and recognize abnormalities in lung X-ray images, among others. For object detection tasks, CNNs can automatically identify vehicles on the road, demonstrating superior speed and accuracy compared to manual methods. In the last ten years, the application of Deep Learning methods, especially CNN architectures, has significantly transformed the area of object detection. Two phase detection models for instance, Faster R-CNN provides superior accuracy, while single-stage models such as YOLO, SSD, and RetinaNet provide high processing rates for real-time applications, which are important for activities that depend on data from onboard cameras. However, though these high performance models, their real efficacy is based upon multiple factors. Including data set reliability, parameter selection, management of different situations, and performance on devices with limited processing resources. Other gap in road object detection research is the ability for domain adaptation to particular road issues, including urban, rural, or regionally distinct environmental features. Datasets frequently utilized, such as COCO, KITTI, or Waymo which are typically gathered from countries with particular environments, thus resulting in reduced efficiency when implemented in different situations. Moreover, specialized object detection tasks, such pothole recognition, road surface condition detection, and small object identification which remain underexplored because to missing of comprehensive, targeted datasets. Consequently, creating models capable of learning diversity from real data presents a significant difficulty in this domain. Current development trends in hardware systems focus both validation and real world testing to authenticate models that collect real data (real time inference) on limited hardware. For example, embedded systems or IoT devices in cars has the capability to process images from many data sources, such as RGB cameras, depth cameras, or LiDAR sensors. This strategy can substantially enhance detection accuracy. However, combining data from several sensors requires combining sensors using methodologies, which is increasing the system's complexity.

*Corresponding author.

In Thailand, road conditions significantly affect the performance of the system. Thailand features various road conditions, including heavily packed urban roads and narrow, uncertain community footpaths. This includes rural roads that often do not have suitable signs or lighting. The diversity and inconsistency of these road conditions present an enormous challenge for object detection programs. One common problem is the damaged road surfaces, including potholes, cracked asphalt, bridge joints, or subsidence resulting from flooding, which frequently occurs in multiple regions of the country. Particularly during the rainy season, these defects show different features and may be smaller in size, making object detection models trained on external datasets often ineffective in accurately identifying them. Consequently, detecting road surface degeneration is a critical research area within the Thai context, as it can mitigate accidents and facilitate preventive maintenance. The other issue is the unique features of road usage in Thailand, like motorcyclists mixing with cars and parking in unauthorized locations. Crossing the roadway at locations other than designated crosswalks, or in the path of temporary barriers such as barriers, traffic cones, or unregulated roadside vendors. These generate various and unexpected objects. Object detection under these conditions requires models capable of learning from real-world data supplied from Thailand for the highest level of accuracy. Moreover, the issue of traffic and warning sign systems on Thai roads is a barrier, as certain road signs might not be up to standards, may be hidden by trees, or may get old and faded. This complicates the detection of traffic signs using models trained on foreign datasets, particularly with temporary warning signs employed at construction sites. These often appear in many ways and lack standardization which important to driving safety so require detection. Moreover, the lighting conditions in various regions of Thailand, including rural roads missing of streetlights and urban roads light by vehicle headlights and visible advertising signs, result in issues of backlighting and reflection, which negatively affect the accuracy of object detection systems. Consequently, developing a model that can adapting different lighting conditions for application in Thailand. Traffic jams is a significant issue that results in time loss for individuals. An accident is an issue that results in the loss of lives and property. Accident causes include driver health issues, vehicle breakdowns, and road surface conditions. Examples of road surface abnormalities include potholes, road humps, irregular manhole covers, spaces between bridges and roads, bumpers, and cracks. An accident resulted in the loss of lives, time, and resources. However, the Department of Highways in Thailand often conducts manual road surveys and implements improvements. This method caused significant risks, expensive, and time-consuming. Deep learning can assist in resolving all issues. The government has opted for a survey of the road, which is efficient and cost-effective. Consumers can determine road surface quality and prevent accident risk. Entrepreneurs can minimize transportation costs through faster driving and a reduction in accidents. Current object detection CNNs, such as different versions of YOLO, may generate models for detecting road conditions [3],[4]. Nonetheless, the accuracy is not sufficient.

Therefore, research in road object detection is a significant domain with considerable potential for development. This research examines object detection methodologies employing deep learning techniques. Evaluation of the model's efficacy on actual road condition datasets, including an analysis of precision, speed and robustness to different environments. To develop a model that effectively matches the requirements of real road usage and can also provide a basis for future implementations in intelligent safety systems and autonomous cars. While YOLO-based detectors displayed strong efficacy in identifying general objects, they were not optimized for the detection of road surface and abnormalities. Cracks and potholes have long shapes, high horizontal continuity, and minimal class differentiation, frequently appearing segmented or partially ignored. Traditional convolution transform blocks are designed for small objects. This limitation necessitates structures that may more effectively identify horizontal data along road surfaces. This research is to customize the YOLO structure according to the data obtained in Phuket and Bangkok, Thailand. The novel CNN architecture is capable of accurately detecting three objects. The aim of this research is to create a model utilizing novel CNN for the detection of road objects, including cracks, potholes, and manhole covers. Subsequently, maintain it in the cloud. Subsequently, the pre-processing phase is executed. Ultimately, to develop the road surface condition detecting model utilizing CNN. The model is capable of identifying three items on the roadway from video footage. This research contributions include 1) a novel CNN architecture for detecting road conditions that provides empirical evidence that implementing horizontal contextual features is essential for delicate detection of road surface condition, 2) suggested an adaptive architectural change for YOLOv6, which improves robustness in under real-world road condition, and 3) Additionally, provide outcomes from experiments utilizing Thai road datasets, demonstrating the limitations of conventional object detectors in detecting surface condition.

The organization of this paper is as follows: Section II discusses the related work. Section III presents the research methodology, encompassing data collecting, frame extraction, and object labeling. Section IV explains the novel CNN architecture termed "RoadSCNet". The subsequent section, Section V, addresses the experiment results concerning three hyperparameters which are dropout, learning rate, and kernel. Section VI presents an experiment including three structures: the modified horizon block, vertical block, and pooling layer. Subsequently, analyze the structural experiment results. This part facilitates the implementation of a revolutionary CNN architecture. The conclusion is located in Section VII.

II. Related Works

This section described previous research. Beginning with image preprocessing, which is necessary for the preprocess of object detection. Cleaning datasets or managing data is the initial stage. Subsequently, the task involves separating or labelling data. The cleaned data is utilized in the model generation phase. The object detection Evolution is detailed in Section II(A). The next step is to generate the model. CNN is appropriate for this step, as numerous CNNs have been utilized in previous research, as detailed in Section II(B). The next

discussion relates to object detection on the road, as outlined in Section II(C). The last section for evaluating the theoretical framework of the approach is detailed in Section II(D).

A. Object Detection Evaluation

The preprocessing phase occurs before the generation of the object detection model, whether using traditional or deep learning methods. Examples of data collection equipment include cameras, smartphones, stereos, LiDAR, and others. This section discusses various techniques employed in the preprocessing phase, including lane detection and road surface analysis.

Before the development of deep learning, object recognition tasks mainly utilized classical image processing and computer vision methodologies, including edge detectors, corner detectors, and sliding windows. The Canny edge detector is employed to identify edges in images, providing the fundamental structure of objects. The Harris corner detector is another instance that identifies significant corners inside an image. Furthermore, a HOG (Histogram of Oriented Gradients) descriptor is employed, which defines the attributes of objects by aggregating the gradient directions in segmented regions of the image, for use in object classification and detection tasks. In classical learning, cascading classifiers, such as Viola-Jones, utilize Haar features and a cascade classifier for sliding window detection. These approaches display accuracy in controlled environments. For example, identifying faces in static photographs which although display several limitations—suboptimal performance on complex images, dependence on manually designed features, and limited generalization over a wide range of objects. Many research worked on image processing. image Quality Enhancement is the standard technique to help reducing noise and increasing an accuracy. The smoothing and filtering technique worked on Gaussian or Median filter [3]. Next technique is Brightness and Contrast Adjustment which working on Histogram Equalization (HE) and Recursively Separated and Weighted, Histogram Equalization (RSWHE) and Gamma Correction. (RSWHE) and Gamma Correction used to solve the Mean Shift problem [4]. Color transformation has to convert the image to grayscale to reduce complex calculations and decrease processing time. The HSV (Hue, Saturation, Value) or Lab transformation is employed in segmentation tasks that focus on color properties. This method employed in traffic sign recognition [5],[6]. Dark Channel Prior (DCP) or Temporal Correlations are employed to decrease fog and dehazing in the management of weather or illumination handling. Subsequently, Otsu's technique or Adaptive Thresholding is employed to automatically convert the image to binary before doing contrast adjustment [3]. The limiting of the Region of Interest (RoI) is useful in minimizing the quantity of complex data and computational complexity. In spatial division, the RoI will be separated from the background. One strategy applied to reduce computational complexity and improve accuracy in various contrast images is Adaptive Region of Interest (ARoI). Horizon line detection is a technique in ARoI utilized to divide the scene into road and sky regions. [3],[7]. Subsequently, edge detection and feature extraction are employed to identify the characteristics of geometry, road surfaces, or objects. This technique is necessary for traditional vision-based method. The Sobel and Canny operators are widely utilized approaches. Sobel is utilized more frequently than Canny and Prewitt due to both lower complexity and greater resistance to interference. Canny has incredible accuracy in identifying borders and positions [3], [8]. A method for edge detection is binarization, specifically Otsu's thresholding. This method aims to automatically identify parameters for separating background and foreground, hence reducing computation time and managing brightness [3]. One feature extraction technique is the Histogram of Oriented Gradients (HOG), which utilizes edge characteristics for feature extraction. This algorithm functions for detected traffic signs. [9],[10]. To improve binary image, Morphological processing is used to remove noise. An example are corrosion, expansion, open operation, and closed operation, etc. [9]. Image resizing is used to set up image dimensions for deep learning preparation. For instance, 32x32, 256x256, etc., this technique operates before training with CNN. The subsequent step involves identifying the RoI before cropping, concentrating only on the RoI area. The multi-channel input operates using transfer learning on the RGB model, example by ImageNet. The gray Computed Tomography (CT) picture is converted to three dimensions and utilized in the model [11],[12],[13].

Since the development of CNN, object detection has entered a new phase, utilizing CNNs as feature extractors. RCNN (Region-based CNN) is a critical advancement, employing a selective search technique for region proposals which is generating object region proposals and inputting each area into a CNN for classification and bounding box improvement. This was created by Ross Girshick and his crew. However, R-CNN is limited by slow progress due to the processing tens or hundreds of region proposals per image. Fast RCNN operates more efficiently by initially convolving the full image, subsequently extracting proposal points by RoI pooling to identify regions from the feature map, and also executing classification and regression, therefore minimizing the number of steps involved [14]. Subsequently, Faster RCNN incorporated a Region Proposal Network (RPN) into the CNN, facilitating expedited and more efficient region proposal creation by the utilization of convolutional characteristics shared between the RPN and the detection head [15]. The R-CNN Series approach, which consists of two steps: proposal and classification, achieves excellent accuracy. However, the limitations include significant delay and making real-time application challenging [16]. After the success of the R-CNN Series which remains limited in speed. Consequently, the notion of a "single shot detector" or "single-stage detector" which avoids the need to segregate suggestions and classify in a separate phase. An example is the Single Shot MultiBox Detector (SSD), which employs a singular network, segments the feature map at various scales, and simultaneously predicts bounding boxes and classifications for multiple default box configurations and aspect ratios, thereby allowing the SSD to identify objects with multiple sizes within a single image without requiring separate region proposals. SSDs have fast performance and excellent accuracy, make more appropriate for real-time applications compared to two-stage methods [17]. In 2015-2016, Redmon et al., introduced the You Only Look Once (YOLO) model as a novel approach which conceptualizes object detection as a regression problem rather

than generating proposals and classifying them separately. Both the bounding box prediction and class probability are executed within a singular network by processing all input images in one pass. YOLO divides the image into grids, with each grid predicting a bounding box and a class at once, resulting in high speed. The first model achieved approximately 45 frames per second. The benefits include low latency and comprehensive learning which including simultaneous training of bounding box and classification. A disadvantage is might have lower localization accuracy compared to proposal-based models, particularly for small objects [18]. YOLOv2 or YOLO9000 improves YOLOv1 with the addition of batch normalization, an increase in input resolution, and the utilization of anchor boxes to improve bounding accuracy. This includes joint classification and detection utilizing ImageNet and COCO datasets for training, allowing multi-category detection over more than 9,000 classes [19]. YOLOv3 enhances the backbone network, employs multi-scale prediction (three levels) to more effectively detect small, medium, and large objects, and utilizes a logistic classifier instead of softmax for class prediction. After that, YOLOv4, YOLOv5, and others were introduced for improving both accuracy and speed. Each model has an improved backbone architecture, optimized loss function, and advanced data training/augmentation techniques to better accommodate real-world applications. YOLO changed object detection by proving the feasibility of real-time detection using a single-stage neural network, independent of external region proposals. The YOLO model is widely utilized in applications necessitating real-time processing, including surveillance systems, driverless vehicles, and CCTV. Furthermore, the YOLO (single network, end-toend) concept has inspired the development of other models that enhance the speed and accuracy trade-off such as SSD, RetinaNet, and several YOLO variants. The deep learning evolution as shown in Fig. 1.

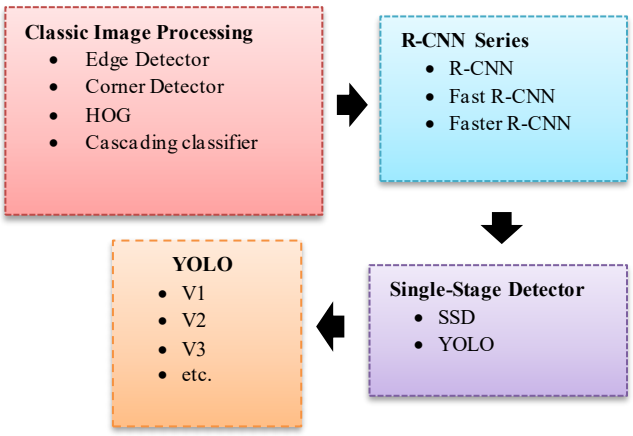

Fig. 1. Evaluation of Object Detection, begin classification image processing until YOLO object detection.

Object detection initiates with traditional image processing techniques, including edge detection, corner detection, Histogram of Oriented Gradients (HOG), and cascade classifiers. Continuing with the R-CNN series, which covers R-CNN, Fast R-CNN, and Faster R-CNN. Thus, YOLO is a single-stage detector, similar to SSD. Several versions of YOLO are currently operational.

B. CNN Object Detection

Object detection is an important component of computer vision that has gained popularity in recent years. This can analyze video, recognize images, and be applied in everyday life. For instance, autonomous driving, drone and robotics analysis, and safety surveillance, among others. The aim of object detection is to identify the localization of objects (object localization) and to classify object types (object classification). Object identification is a subset of deep learning that enhances the accuracy and speed of computer learning. Machine Learning is a method for creating models that operate autonomously. The handcrafted features and shallow trainable architectures were utilized. Initially, the selection of informative regions is conducted by scanning images with multiple sizes of sliding windows. Subsequently, feature extraction is conducted using SIFT, HOG, Haar-like features, among others. The classification is enhanced by methods such as SVM and AdaBoost. Deep Learning (DL) is a more advanced technique than machine learning. Neural networks can enhance model performance [20], [21]. Convolutional Neural Networks (CNNs) are utilized to create models with various architectures. The CNN architecture combines One-stage and Two-stage Detectors. The two-stage detector employs Regions with CNN features (RNN) as the principal model, demonstrating the efficiency of CNN in object detection. RNN employs a bottom-up approach for locating positions and objects within images. Subsequently, obtain the region suggestion by selective search and compute the features using a convolutional neural network (CNN). Subsequently, categorize the region type utilizing Support Vector Machine (SVM) methodology. This procedure is time-consuming and presents a bottleneck issue. This study achieves a mAP of 53.30% utilizing the PASCAL VOC 2012 dataset. The model generated by VGG16 achieved a mAP of 66.00% on the VOC Convolutional Networks (SPP-net) addresses the bottleneck issue of R-CNN by employing a Spatial Pyramid Pooling layer (SPP layer) in its final layer. Consequently, CNN can process input and output images of a fixed dimension. SPP-net (ZF) achieves a mAP of 60.9% on the VOC 2007 dataset [23]. Fast R-CNN enhances R-CNN and SPP-net by end-to-end training that integrates area selection, feature extraction, and classification. The VOC 2007 test set achieved a mAP of 70.00% [15]. Subsequently, Faster R-CNN is executed on Fast R-CNN utilizing the Region Proposal Network (RPN) instead selective search. This method generates region proposals from the features of a CNN RPN and predicts box regression (reg) and box classification (cls) using a sliding window approach. The results indicate a mAP of 83.80% [24]. Mask R-CNN is the instance segmentation framework that extends Faster R-CNN. Mask R-CNN achieves an Average Precision (AP) of 39.8% on the COCO test set [25]. The R-FCN, or Region-based Fully Convolutional Networks, achieves a mean Average Precision (mAP) of 85.00% on the VOC 2012 test set [26].

One-stage detectors, such as the Single Shot MultiBox Detector (SSD), perform classification and bounding box regression immediately. SSD300 has a mAP of 79.60%, whereas SSD512 demonstrates a mAP of 81.60% when evaluated on the VOC 2007 test set. SSD512 exhibits a mean mAP of 81.60%. Feature Pyramid Networks (FPN) are utilized for the extraction of features across diverse data sizes [17].

YOLO (You Only Look Once) is a famous real-time object detection model known for its speed and accuracy. YOLO V1 is the first end-to-end real-time object detection model [18]. YOLO V2 (YOLO9000) operates on the Darknet-19 backbone, employs batch normalization, utilizes anchor boxes for predictions, and includes a passthrough layer for feature extraction [27]. YOLO V3 employs the Darknet-53 backbone which is larger than the backbone utilized in YOLO V2. Bounding box prediction employs logistic regression and three sizes of multi-scale predictions. All operations involving the SPP block [28]. YOLO V4 was created for fast and accurate performance. The Bag of Freebies and the Bag of Specials are utilized to enhance performance and combine a receptive field. The backbone is CSPDarknet53. The SPP module, PANet path aggregation neck, and YOLO V3 head are processed [29]. YOLO V5 utilizes a modified CSPDarknet53 as backbone commencing with a Stem and SPPF (Spatial Pyramid Pooling Fast) layer. This facilitated rapid accuracy. The auto anchor verifies and enhances anchor boxes [30]. YOLO V6 is the suitable framework for industry applications. The EfficientRep backbone (RepBlock/RepConv) has been processed. The quantization employs GloU (Generalized IoU), Clou (Complete-IoU Loss), Slou loss, self-distillation, and channel-wise distillation [31]. YOLO V7 utilized the Extended Efficient Layer Aggregation Network (E-ELAN) and RepConvN, which lacks identity connections. YOLO V8 modified C3 to C2f and implemented 3x3 convolution [31] - [33]. In addition to YOLO, Pelee is a real-time object detection framework for mobile devices utilizing PeleeNet. PeeleeNet is produced using traditional convolution rather than depthwise separable convolution with SSD Pelee [34].

C. Road Object Detection

Numerous studies on road surface conditions utilize video and imagery. The model identified a road traffic sign utilizing Support Vector Machine (SVM) technology. This is classified and identified in the video. The Histogram of Oriented Gradients (HOG) is utilized to extract features previous to model generation [9]. Traffic light recognition employs image processing techniques based on color segmentation and the Hough transform for circles. Machine learning is utilized to generate the model. This model was designed to identify the traffic light in the autonomous vehicle [6]. Utilization of Helmets Detection on motorcycles is an important part of their widespread usage. YOLO V3 and multi-task learning are employed for helmet detection. Positional Encoding (PE) resolved the issue of class imbalance in this research, resulting in PE is better than the class-balanced loss [35],[36],[37]. Advanced Driver-Assistance Systems (ADAS) contain a collection of sensors, cameras, and other technologies within the vehicle that enhance driving safety and convenience. ADAS and safety alerts contribute to accident reduction studies. The Collision Avoidance System utilized the Google TensorFlow Object Detection (GTOD) API to build a driving safety alert and real-time vehicle detection, trained using the Microsoft Common Objects in Context (COCO) dataset [38]. Then the motorcycle curve warning presented the curve alert system including the intra-lane localization and roll angle estimation using CNN. The industry standard maps is used for improve an accuracy [39]. The pothole was detected using thermal imaging captured by a FLIR ONE camera. The principle that the temperature of the pothole is lower than that of the surrounding road because of water retention. This research performed effectively at night, during rain, and in fog, remaining unaffected by light, exhibiting a high response time, lower energy consumption, and reduced costs compared to laser-based techniques. The training with the ResNet101 model achieved an accuracy of 97.08% [40]. A pothole detecting system was developed for road surfaces to help blind people utilizing Convolutional Neural Networks (CNN) and the KITTI dataset. This system achieved an accuracy of 97.12% [11]. Numerous CNN models have been trained to detect potholes, including Inception v4, Inception ResNet v2, ResNet v2 152, and MobileNet v1. An accuracy above 96%. The grayscale image can be utilized in this research [14]. Crack detection is performed automatically using CNN on images captured by smartphones, employing YOLO and SSD methodologies [41],[42]. Research to present a 3D map as the outcome. The SSD can develop a model to detect potholes and humps by utilizing data collected from a Raspberry Pi, GPS module, and camera [43]. The video data is utilized to identify cars, trucks, and buses using Faster R-CNN in comparison to a mixture of Gaussian (MoG) background subtraction and SVM vehicle classification. The results indicated that Faster RCNN exhibited superior performance [44]. Furthermore, the accelerometer, gyroscope, and GPS in smartphones gather data that is utilized with machine learning to classify multiple categories, including smooth roads, potholes, and deep transverse cracks. The three feature axes of the sensor provide more accuracy, precision, and recall compared to a single axis [45]. In addition, the fog and darkness decreased the driver's visibility, presenting a problem that must be resolved for safety. Recursively Separated and Weighted Histogram Equalization (RSWHE) and Gamma Correction techniques were worked to increase the image and video performance. Detecting road surfaces at night can be achieved using the headlights of a vehicle [4], [46]. The summary table of techniques for road condition detection is presented in Table I.

TABLE I. SUMMARY OF THE ROAD CONDITION DETECTION TECHNIQUES

Method	Techniques
Data Collector	Smartphone, Video, Camera, Raspberry Pi, GPSmodule
Data Preparation	Frame extraction, Lane detection, Labelling, Image processing
Model Generation	YOLO, Faster R-CNN, ResNet, Inception, MobileNet, SSD, SVM, HOG

D. Evaluation

In object detection and classification, evaluation is crucial to evaluating performance during the development and maintenance phases. The object detection task needs to allow for both localization accuracy and classification accuracy. The confusion matrix, Intersection over Union (IoU), and mean Average Precision (mAP) are widely utilized metrics.

The Confusion Matrix is a fundamental instrument to evaluate the accuracy of a classification model by comparing the actual class with the predicted class. There are four values in a confusion matrix. A True Positive (TP) occurs when a prediction is positive and the actual outcome is also positive. A True Negative (TN) occurs when a prediction is negative and

the actual outcome is also negative. A False Positive (FP) occurs when a prediction indicates a positive result when the actual outcome is negative. A False Negative (FN) occurs when a prediction is negative while the actual outcome is positive. Subsequently, these four variables are utilized to compute additional values below.

Accuracy is the proportion of correct predictions, which may be calculated using Eq. (1).

$$\text{Accuracy} = \frac{TP+TN}{\text{TP+TN+FP+FN}} \quad (1)$$

Precision refers to the quality of a model concerning positive precision. The precision is illustrated in Eq. (2).

$$Precision = \frac{TP}{\text{TP+FP}} \quad (2)$$

Recall assesses the efficacy of a model in predicting positive results. The value of recall can be calculated using Eq. (3).

$$Recall = \frac{TP}{\text{TP+FN}} \quad (3)$$

The F1-score is the harmonic mean of precision and recall, serving as a singular metric to assess model performance. The F1-score equation is shown in Eq. (4).

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

In the object detection job, the Intersection over Union (IoU) serves as the localization measure that compares the area between the predicted bounding box and the ground truth bounding box. The IoU equation is illustrated in Eq. (5).

$$IoU = \frac{Area\ of\ Overlap}{\text{Area of Union}} \quad (5)$$

Then Area of Overlap is two boxes overlap area. And Area of Union is total area of both boxes.

The IoU value ranges from 0 to 1. If the IoU is greater than or equal to 0.5, determine it as a true detection (True Positive). The higher IoU mean predicts the bounding box's closer to the object's actual position.

Normally in object detection task has multiple object type and different confidence score. The average precision (AP) is measure model performance for each class and calculate by Precision-Recall Curve. After that calculate average AP of all class that is mean Average Precision (mAP) as calculate by Eq. (6).

$$mAP = \frac{1}{\text{N}} \sum_{i=1}^{N} AP_i \quad (6)$$

N represents the total number of classes, while APi is the average precision for class i. The mAP summarizes the model's performance in object detection and classification over all objects.

III. Research Methodology

This section described the methods employed in this research. Subsection A. presents the framework, which includes an overview of the research. The subsequent step is the preprocessing phase which involves data collection and frame extraction. Frame extraction can divide the video dataset into multiple images, which is the initial important phase as illustrated in subsection B. The following procedure is object labelling, which enables the model to recognize the item as illustrated in C.

A. Framework

This section described the research framework. Data was gathered from May 2020 to September 2021 in Phuket and Bangkok, Thailand. The smartphone is a collection instrument mounted to the vehicle's windshield. All data belongs in cloud storage. The video data is segmented into numerous frames with the frame extraction technique. The work involves image labelling, which encompasses cropping and identifying each object within all images. Laberu is an appropriate tool for this stage. This is a novel labelling tool created by our team. This study examines cracks, potholes, and manhole covers. 24,276 frames are utilized for labelling. The XML file provides the result of the labelling phase. The object detection model is built using a novel CNN customized on the backbone of YOLO V6. The research framework is illustrated in Fig. 2.

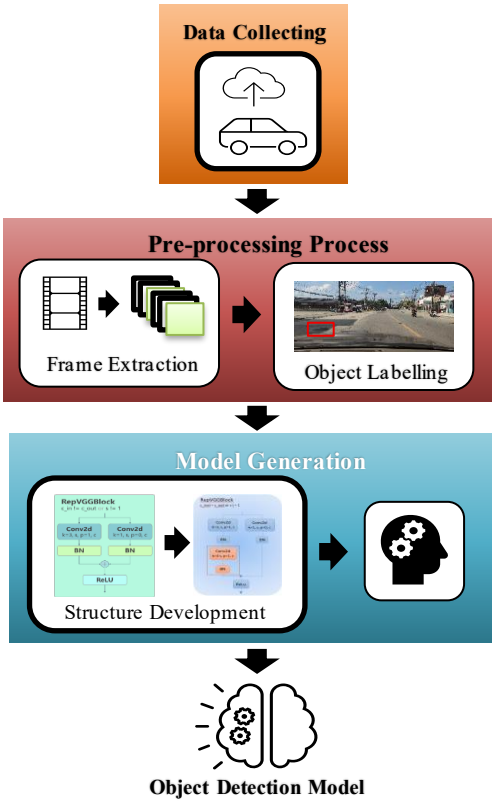

Fig. 2. Framework for the development of novel CNN architectures.

B. Data Collection and Frame Extraction

The initial phase of this research has involved the collection of video data. The equipment of this research is the smartphone. The smartphone was installed on the windshield of the vehicle operating on the roads of Phuket and Bangkok from May 2020 to September 2021. The total distance is 1,118.13 kilometers. 125 videos had collected a total duration of 134,176 seconds. Subsequently, the frame extraction goes into processing. This phase involves separating all of the video into multiple images. The quantity of data collected,

comprising 125 videos with a total duration of 134,176 seconds. Following the frame extraction process, a total of 1,341,760 frames were obtained. Selected 31,803 frames containing three objects. Ultimately, 24,276 frames were selected for the subsequent stage.

The data categories are identified as three objects inspected by the Department of Highways in Thailand. This investigation encompasses crack, pothole, and manhole cover. An example of manhole cover after extended from video as shown in Fig. 3.

C. Object Labelling

This subsection explained the process of object labelling. The step is to separate and select all three objects. This phase was successful when the video was captured and the frame was extracted. This process employs three tools. LabelMe and LabelImg are universal labelling tools. Labelru is the labeling tool developed by our research team. This research is designated by Labelru. This tool allows for multiple users immediately. Fig. 4 illustrates an example of each object post-labeling. The object's name and its position are indicated within the red rectangle.

Fig. 3. The image including Manhole cover extracted from the video.

```
<annotation>
  <folder>picture</folder>
  <filename>2-session-1610784850-7866.jpg</filename>
  <path>E:\select\211011\211011\picture\2-session-1610784850-7866.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Manhole covers</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>125</xmin>
      <ymin>762</ymin>
      <xmax>270</xmax>
      <ymax>801</ymax>
    </bndbox>
  </object>
</annotation>
```

Fig. 4. An example of a labelled XML file, red rectangle show object name and coordinates.

IV. Road Surface Condition Detection Network

Convolutional Neural Network (CNN) is the popular technique to generate a model to detect an object. At the time, there are many CNN structure work for classification and object detection work. The concept of CNN is learning similar the human brain, CNN is in the bio-inspire group. The object detection works base on classification and object detection [22].

This section presents Road Surface Condition Detection, or RoadSCNet. This section explores the novel framework for the object detection job. YOLO is a renowned convolutional neural network design utilized for object detection tasks. Numerous iterations of YOLO exist, ranging from YOLO V1 to YOLO V8, together with YOLONas, developed from 2016 to the present. This research focuses on YOLO V6 and a novel CNN called "RoadSCNet" which is based on this version. RoadSCNet is a novel convolutional neural network derived from YOLO V6. Typically, YOLO V6 is suitable for fast and accurate real-time object recognition. There are multiple versions of YOLO V6: YOLO V6-n, YOLO V6-S, YOLO V6-M, YOLO V6-L, and YOLO V6-tiny. YOLO V6-tiny is compact and suitable for IoT or mobile applications. The structure of YOLO V6 is illustrated in Fig. 5.

The architecture of the YOLO V6 backbone has five RepVGGBlock layers, four BepC3StageBlock layers, and concludes with an SPPFBottleneck. The kernel sizes 4x4. The BN layer refers to the batch normalization layer which enhances the mean and standard deviation of the activation output from the layer above. The appropriateness and stability of the mean and standard deviation facilitate the training of a more stable and effective model.

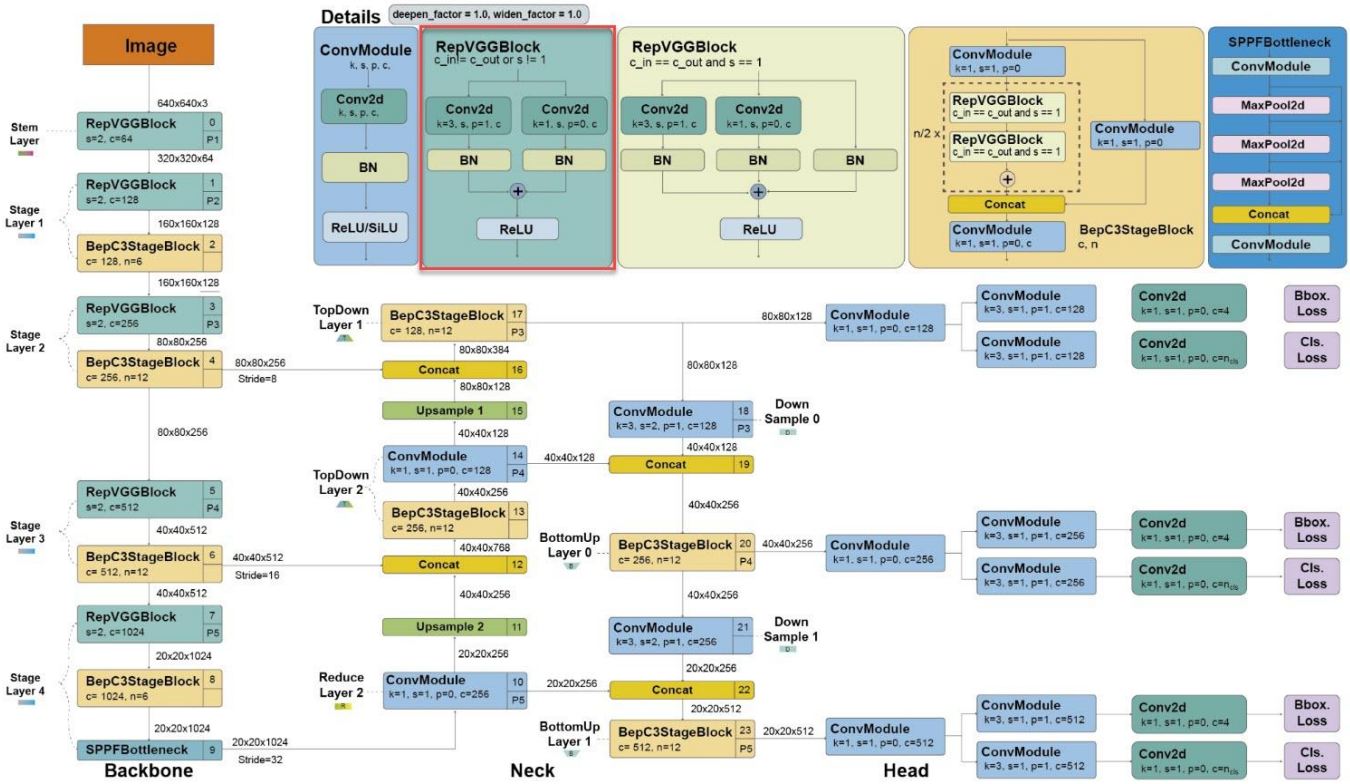

Fig. 5. The architecture of YOLO V6. The red rectangle on the RepVGGBlock indicates the position for block customization. Figure is based on study [47]

V. Hyperparameter Experiment

This part offered the experiment results and comments regarding hyperparameters. Video data was taken using a smartphone mounted on the windshield of the car. Data was obtained in Phuket and Bangkok, Thailand. The initial pre-processing stage involved frame extraction, which converted the video into several images. Subsequently, every three images were selected for the following step. The object labelling was subsequently processed. During the labelling phase, the novel labelling tool in this research, named "Labelru" received approval. This tool can be labelled by multiple users simultaneously. The produced model was enhanced. The experiment has dropout, learning rate, and kernel which are as mentioned in subsection A - C. The experiment setup utilizes a computer equipped with an i9-10900KF CPU, 32GB of RAM, and an RTX 3080 10GB GPU. Miniconda3 and Python 3.6 are software applications.

A. Dropout

Overfitting is a challenge encountered throughout the model training process. The method employed to reduce the overfitting issue is dropout. The dropout layer randomly deactivates certain neurons within that layer. This strategy enables the model to learn by distributing tasks among other neurons, hence enhancing its performance when predicting fresh datasets. The experiment with four dropouts is analyzed below using the mAP value. This research focused on dropout rates 0.2, 0.4, 0.6, and 0.8. The outcome with a dropout of 0.6 yielded a mAP value of 86.97. The subsequent dropout values are 0.2, 0.4, and 0.8 yielding mAP values of 86.41, 86.30, and 85.62, respectively. Subsequently, a dropout rate of 0.6 was employed in the subsequent experiment learning rate as delineated in subsection B.

B. Learning Rate

The learning rate is a hyperparameter that controls weights and biases of the model. In each iteration of the training process, the learning rate is multiplied by the gradient to adjust the weights and minimize loss. Particularly in Gradient Descent training. If the learning rate is excessively high, the model will set the weights and biases in an unstable manner during training. The low learning rate results in minimal weight and bias adjustments which are leading to protracted and time-intensive training. This section elucidates the learning rate experiment. The values are 0.1, 0.01, 0.001, and 0.00001. The outcome is presented in Table II.

According to Table II, the learning rate of 0.01 achieved the greatest mAP value of 86.97. The subsequent learning rates are 0.1, 0.001, 0.0001, and 0.00001, yielding mAP values of 85.78, 83.26, 82.33, and 62.54, respectively. The learning rate of 0.01 was utilized in the subsequent experiment.

TABLE II. Various Experiment Outcomes Learning Rate

Learning Rate	mAP	Time (hour)
0.1	85.78	13.95
0.01	86.97	13.34
0.001	83.26	13.10
0.0001	82.33	13.22
0.00001	62.54	13.20

C. Kernel

In CNN, a kernel or filter operates on the input image by performing convolution to process the data. This stage involves extracting features from the image. The convolution applies a stride kernel to a region of the image and computes the result based on the image pixels and weights in terms of the feature map. This subsection discusses the kernel experiment. The 3x3 and 4x4 kernels have been utilized. The 3x3 kernel demonstrates a superior mAP value compared to the 4x4 kernel. The mAP values are 86.97 and 86.29. Consequently, a 3x3 kernel is suitable for this research.

VI. Structure Experiment

After completing the hyperparameter experiments outlined in Section V. This section explains the experiment structure. Three experiments are being conducted on the RepVGGBlock based on YOLO V6. The system comprises a CPU i9-10900KF, 32GB of RAM, and an RTX 3080 10GB GPU. Miniconda3 and Python 3.6 are software applications. Subsection A. illustrates the various versions of YOLO when generating models utilizing the data. Subsections B. to D. analyzed three experiments.

A. Existing YOLO

Before developing the CNN structure, the model generated with the existing CNN is active. YOLO V5, YOLO V6, YOLO V7, and YOLO V8 are active. The image resolution of three objects is 1280x720 pixels, processed with a batch size of 8 within 100 epochs. The outcomes of the four existing YOLO models are presented in Table III.

TABLE III. Three Objects Detection Using Existing YOLO

Architecture	mAP
YOLO V5	74.50
YOLO V6	86.19
YOLO V7	61.30
YOLO V8	73.82

Table III presents the results for YOLO V5 to YOLO V8, displaying mAP values of 74.50, 86.19, 61.30, and 73.82, respectively. YOLO V6 has the greatest mAP value. YOLO V5 and YOLO V8 are similar. YOLO V7 displays a lower mAP compared to others. After that, the normalization process was compared with four different batch sizes: 4, 8, 16, and 32. The results gave mAP values of 82.70, 86.19, 86.98, and 87.38, respectively. Batch size 32 displays the highest mAP. The next batch size of 32 is employed in dropout regularization. There are four dropout rates in running: 0.25, 0.50, 0.75, and 1.00. The mAP values are 85.40, 85.73, 85.15, and 39.24, respectively. The no dropout showed better results [48]. While evaluating the results, the maximum mAP is 87.38, which is minimal for the collected data. The improvement of the mAP value in the backbone of YOLO V6 architecture is discussed in the next subsection.

B. Horizon Block

The initial experiment construction is incorporating Conv2d into the RepVGGBlock. In the YOLO V6 architecture, Conv2d operates before Batch Normalization (BN). The initial experiment involves executing Conv2d subsequent to the BN layer, as illustrated in Fig. 6. The outcome is presented in E.

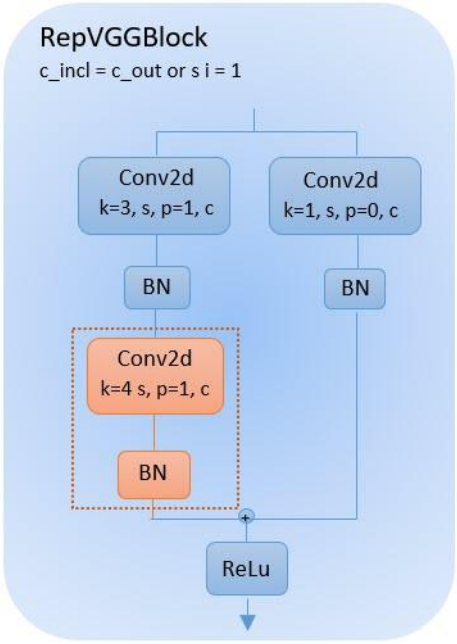

Fig. 6. The newly introduced horizon module within the RepVGGBlock.

C. Vertical Block

This subsection examines the new vertical block in RepVGGBlock following the initial experiment. Conv2d is already integrated with the Batch Normalization layer. The other parameter is k=4x4. This layer typically has two Conv2d and two BN layers; however, this experiment incorporated both Conv2d and BN layers. The revised structure is depicted in Fig. 7. The outcome as indicated in E.

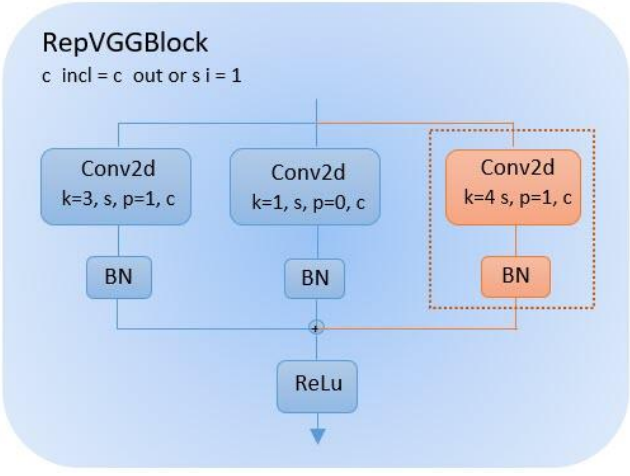

Fig. 7. The newly vertical module in RepVGGBlock.

D. Pooling Layer

The SPPFBottleneck operates in the final layer of the backbone. The YOLO V6 architecture operates on max pooling. The final experiment utilized average pooling. Compare the outcome as detailed in Table IV.

TABLE IV. THE MAP OF EXPERIMENT RESULTS FROM DIFFERENT POOLING METHODS

Experiment	Max Pooling	Average Pooling
Original	86.13	86.60
Width Multiple 0.6	88.07	88.25
Width Multiple 0.65	88.73	88.89
Width Multiple 0.7	89.96	89.09
Width Multiple 0.7 with Learning Rate		
Learning Rate 0.1	89.38	89.19
Learning Rate 0.01	89.96	89.09
Learning Rate 0.001	89.38	88.89
Learning Rate 0.0001	89.26	89.47

According to Table IV, a width multiple of 0.7 for max pooling yields the greatest mAP value of 89.96. The subsequent width multiples are 0.65 and 0.6, corresponding to mAP values of 88.73 and 88.07, respectively. The average pooling with a width multiplied by 0.7 achieves the maximum mAP value of 89.09. Multiply 0.65 and 0.6 by 88.89 and 88.25. The width is multiplied by 0.7 using max pooling for late experiment learning. When evaluating the Learning Rate using a width multiplier of 0.7 and max pooling, a learning rate of 0.01 yields the greatest mAP value of 89.96. The learning rates of 0.1 and 0.001 yield a mean Average Precision (mAP) value of 89.38, while a learning rate of 0.0001 results in a mAP value of 89.26. The learning rate of 0.0001 yields the highest mean Average Precision (mAP) value of 89.47 for average pooling. The subsequent learning rates are 0.1, 0.01, and 0.001, yielding mAP values of 89.19, 89.09, and 88.89, respectively. In this research, a maximum pooling width of 0.7 and a learning rate of 0.01 have been accepted as appropriate. Nevertheless, the comprehensive outcomes of all experiments are elucidated in subsection E.

E. Structure Experiment Result

This part examined all experiments in the Structure Experiment which including the results of the inserted horizon block, vertical block in RepVGGBlock, and the application of average pooling in SPPFBottleneck, as presented in sections VI.B. – VI.D. The experiment commenced with the original YOLO V6, utilizing a width multiplier of 0.5 and a learning rate of 0.01. The two hyperparameters modified in this study are presented in Table V. The experiment run consisted of 100 epochs using a machine equipped with an Intel i9-10900KF CPU, 32GB of RAM, and an RTX 3080 10GB GPU. Miniconda3 and Python 3.6 are software applications. Experiment 3 presents the results of maximum pooling.

According to Table V, Experiment 1 involves a modified horizon block within the RepVGGBlock. The outcome of a width multiplier of 0.7 yields the maximum mAP value of 89.91. The width multiples of 0.65 and 0.6 yield mAP values of 88.70 and 88.22, respectively. The width multiplied by 0.7 was effective for the learning rate experiment. The learning rate experiment with a width multiplier of 0.7 achieved the greatest mAP value of 90.01 at a learning rate of 0.001. The learning rates of 0.01, 0.0001, and 0.1 yield mAP values of 89.91, 89.51, and 89.27, respectively. The combination of a width multiplier of 0.7 and a learning rate of 0.001 yields the highest mAP value of 90.01.

TABLE V. THE MAP OUTCOMES OF MANY EXPERIMENTS

	Experiment 1	Experiment 2	Experiment 3
Original	86.97	86.12	86.60
Width Multiple 0.6	88.22	88.07	88.25
Width Multiple 0.65	88.70	88.73	88.89
Width Multiple 0.7	89.91	88.97	89.09
Width Multiple 0.7 with Learning Rate			
Learning Rate 0.1	89.27	89.38	89.19
Learning Rate 0.01	89.91	88.97	89.09
Learning Rate 0.001	90.01	89.69	88.89
Learning Rate 0.0001	89.51	89.26	89.47

Experiment 3 compared the average pooling layer in SPPFBottleneck with max pooling. The width multiple of 0.7 for max pooling and a learning rate of 0.0001 yielded the greatest mAP value of 89.47. Table V presents the results of Experiment 3 conducted with max pooling.

Upon evaluating the maximum mAP value from three experiments, Experiment 1, which utilized a width multiplier of 0.7 and a learning rate of 90.01, exhibited the highest mAP value. Experiment 2 achieved a mAP of 89.69 with a width multiplier of 0.7 and a learning rate of 0.001. Experiment 3 achieved a mAP of 89.47 when employing max pooling, a width multiplier of 0.7, and a learning rate of 0.0001. Ultimately, the modified horizon block in RepVGGBlock, after 200 epochs, achieved a mAP value of 96.03, which is conclusive for this research.

YOLO V6 employs RepVGGBlock for object detection, providing both high performance and speed. This study focuses on specific convolution. The result is positive for objects with specific boundaries. This is unsuitable given the road conditions characterized by cracks, potholes, and manhole covers. The three objects contain shapes, low sharpness, and irregular borders, primarily aligned along the horizontal axis of the road. Specifically, RepVGGBlock misses an element of context that clearly considers direction. This limits the ability to represent the horizontal continuity necessary for accurate identification of road conditions. Motivated by this limitation, RoadSCNet enhances the fundamental architecture of YOLOv6 by integrating a customized Horizon block with the RepVGGBlock to improve horizontal context learning efficacy while preserving computational efficiency. Then when run the new RoadSCNet model with 182 unseen data images which collected in Phuket, the result shown 91.60% accuracy. As displayed in Fig 8. The confusion matrix shows the model can identify 64 cracks, 3 potholes, and 122 manhole covers. All three objects include 72 cracks, 11 potholes, and 122 manhole covers. The accuracy is 91.60% and the precision is 94.50%. Then the model shows the performance to detect three objects. An example of three objects when detected using the model as shown in Fig. 9.

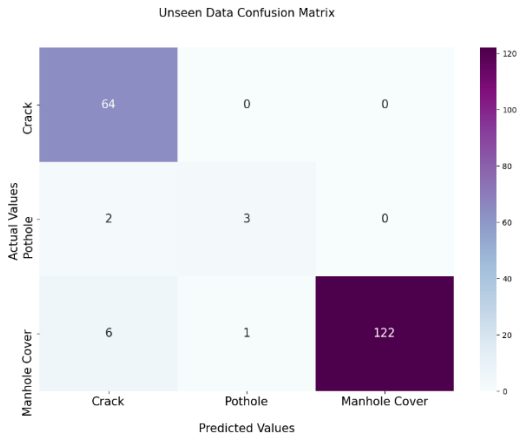

Fig. 8. The confusion matrix when test model with unseen data.

Fig. 9. Three examples when detected by RoadSCnet model.

The limitations of this investigation include window of the car which resulted in erroneous detection, the blurred images which made difficult the identification of the actual object, and the limitaion of equipment caused in reduced operational speed. Future work will focus on improving the mAP value and decreasing the loss value of the model.

VII. Conclusion

This research introduces RoadSCNet, an innovative CNN architecture that achieves a high mAP value for the dataset. The evaluation of three hyperparameters showed dropout 0.6, learning rate 0.01, and kernel 4x4 yielded the greatest mAP value. Subsequently, three distinct structures yield results for modeling under three road conditions, incorporating the new horizon block which demonstrates a superior mAP, followed by the integration of vertical adjustments and pooling modifications. The RoadSCNet enhances YOLO V6 by incorporating the new horizon block within the RepVGGBlock. RoadSCNet outperforms classic YOLO V6 due to a greater ability to gather horizontal contextual information, essential for identifying long cracks and rough road conditions, including cracks pothole and manhole covers.

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 under grant no. 2046735, Ministry of Higher Education, Science, Research and Innovation, Thailand.

References

[1] A. Choudhary, R. D. Garg, and S. S. Jain, "Estimating impact of pavement surface condition and geometrics design on two-wheeler run-off road crashes on horizontal curves," IATSS Research, vol. 48, no. 1, pp. 108–119, Apr. 2024.

[2] P. K. R. Lebaku, L. Gao, J. Sun, X. Wang, and X. Kang, "Assessing the influence of pavement performance on road safety through crash frequency and severity analysis," *arXiv preprint*, Jun. 19, 2025. [Online]. Available: https://arxiv.org/abs/2506.16485.

[3] M. Marzougui, A. Alasiry, Y. Kortli, and J. Baili, "A lane tracking method based on progressive probabilistic hough transform," IEEE access, vol. 8, pp. 84 893–84 905, 2020.

[4] D. Kaur and N. K. Garg, "Enhancement in foggy road scene videos using rswhe and gamma correction," in 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE). IEEE, 2016, pp. 276–281.

[5] A. De La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road traffic sign detection and classification," IEEE transactions on industrial electronics, vol. 44, no. 6, pp. 848–859, 1997.

[6] R. Kulkarni, S. Dhavalikar, and S. Bangar, "Traffic light detection and recognition for self driving cars using deep learning," in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018, pp. 1-4.

[7] H. Wang, Y. Wang, X. Zhao, G. Wang, H. Huang, and J. Zhang, "Lane detection of curving road for structural highway with straight-curve model on vision," IEEE Transactions on Vehicular Technology, vol. 68, no. 6, pp. 5321–5330, 2019.

[8] X. Nie, Y. Gao, F. Gao, Q. Li, and Z. Alam, "A novel vision based road detection algorithm for intelligent vehicle," in 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2019, pp. 504-507.

[9] J. Zhao, Z. Bai, and H. Chen, "Research on road traffic sign recognition based on video image," in 2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA). IEEE, 2017, pp. 110-113.

[10] H. M. Saddique, A. Raza, Z. U. Abideen, and S. N. Khan, "Exploring deep learning based object detection architectures: a review," in 2020 17th International Bhurban conference on applied sciences and technology (IBCAST). IEEE, 2020, pp. 255–259.

[11] M. M. Islam and 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). IEEE, 2018, pp. 268–273.

[12] K. E. An, S. W. Lee, S.-K. Ryu, and D. Seo, "Detecting a pothole using deep convolutional neural network models for an adaptive shock observing in a vehicle driving," in 2018 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2018, pp. 1-2.

[13] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016.

[14] R. Girshick, J. Donahue, T. Darrell, U. Berkeley, and J. Malik, "Rcnn: region-based convolutional neural networks," in Proc. Comput. Vis. Pattern Recognit, 2014, pp. 2–9.

[15] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in neural information processing systems, vol. 28, 2015.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European conference on computer vision. Springer, 2016, pp. 21–37.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE

conference on computer vision and pattern recognition, 2016, pp. 779–788.

[19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263-7271.

[20] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[21] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," IEEE access, vol. 7, pp. 128 837-128 868, 2019.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149, 2016.

[25] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[26] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via regionbased fully convolutional networks," Advances in neural information processing systems, vol. 29, 2016.

[27] K. Boudjit and N. Ramzan, "Human detection based on deep learning yolo-v2 for real-time uav applications," Journal of Experimental & Theoretical Artificial Intelligence, vol. 34, no. 3, pp. 527–544, 2022.

[28] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[29] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Deep convolutional neural network for enhancing traffic sign recognition developed on yolo v4," Multimedia Tools and Applications, vol. 81, no. 26, pp. 37 821–37 845, 2022.

[30] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-time vehicle detection based on improved yolo v5," Sustainability, vol. 14, no. 19, p. 12274, 2022.

[31] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie et al., "Yolov6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022.

[32] Y. Eği, "Yolo v7 and computer vision-based mask-wearing warning system for congested public areas," Journal of the Institute of Science and Technology, vol. 13, no. 1, pp. 22–32, 2023.

[33] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," Machines, vol. 11, no. 7, p. 677, 2023.

[34] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," Advances in neural information processing systems, vol. 31, 2018.

[35] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in 2013 28th International conference on image and vision computing New Zealand (IVCNZ 2013). IEEE, 2013, pp. 35–40.

[36] H. Lin, G. Chen, and F. W. Siebert, "Positional encoding: improving class-imbalanced motorcycle helmet use classification," in 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021, pp. 1194–1198.

[37] D. S. S. Sarma, D. M. Varun, and A. M. Posonia, "Helmet detection and number plate extraction using machine learning," in 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2021, pp. 947–951.

[38] C.-H. Hsieh, D.-C. Lin, C.-J. Wang, Z.-T. Chen, and J.-J. Liaw, "Real-time car detection and driving safety alarm system with google tensorflow object detection api," in 2019 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, 2019, pp. 1–4.

[39] S. Hecker, A. Liniger, H. Maurenbrecher, D. Dai, and L. Van Gool, "Learning a curve guardian for motorcycles," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019, pp. 2984–2991.

[40] Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, A. Akula et al. "Convolutional neural networks based potholes detection using thermal imaging," Journal of King Saud University-Computer and Infomation Sciences, vol. 34, no. 3, pp. 578–588, 2022.

[41] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated road crack detection using deep convolutional neural networks," in 2018 IEEE international conference on big data (big data). IEEE, 2018, pp. 5212–5215.

[42] H. Kim, Y. Lee, B. Yim, E. Park, and H. Kim, "On-road object detection using deep neural network," in 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia). IEEE, 2016, pp. 1–4.

[43] K. A. Govada, H. P. Jonnalagadda, P. Kapavarapu, S. Alavala, and K. S. Vani, "Road deformation detection," in 2020 7th International Conference on Smart Structures and Systems (ICSSS). IEEE, 2020, pp. 1–5.

[44] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, "Detection and classification of vehicles for traffic video analytics," Procedia computer science, vol. 144, pp. 259–268, 2018.

[45] A. Basavaraju, J. Du, F. Zhou, and J. Ji, "A machine learning approach to road surface anomaly assessment using smartphone sensors," IEEE Sensors Journal, vol. 20, no. 5, pp. 2635–2647, 2019.

[46] Y. Horita, S. Kawai, T. Furukane, and K. Shibata, "Efficient distinction of road surface conditions using surveillance camera images in night time," in 2012 19th IEEE International Conference on Image Processing. IEEE, 2012, pp. 485–488.

[47] R. Hasan, R. Hassoon, and I. Salman, "Yolo versions architecture: Review," International Journal of Advances in Scientific Research and Engineering, vol. 09, pp. 73–92, 11 2023.

[48] S. Sa-ngiem, K. Dittakan, and S. Boonsiripant, "The road conditions detection using the convolutional neural network," Indonesian Journal of Electrical Engineering and Computer Science, vol. 40, no. 1, pp. 327–345, Oct. 2025.