Exploring Automated Object Detection Methods for Manholes using Classical Computer Vision and Deep Learning for Autonomous Vehicles

Shika Rao[1], Nitya Mitnala[1,2]
[1] *Birla Institute of Technology and Science, Pilani – Hyderabad Campus, India*
[2] *Texas Instruments (India): Bengaluru, Karnataka, India*

Abstract. Open, broken, and improperly closed manholes can pose problems for autonomous vehicles and thus need to be included in obstacle avoidance and lane-changing algorithms. In this work, we propose and compare multiple approaches for manhole localization and classification like classical computer vision, convolutional neural networks like YOLOv3 and YOLOv3-Tiny, and vision transformers like YOLOS and ViT. These are analyzed for speed, computational complexity, and accuracy in order to determine the model that can be used with autonomous vehicles. In addition, we propose a size detection pipeline using classical computer vision to determine the size of the hole in an improperly closed manhole with respect to the manhole itself. The evaluation of the data showed that convolutional neural networks are currently better for this task, but vision transformers seem promising.

Key words: computer vision, object detection, size detection, Convolutional Neural Networks, Vision Transformers, autonomous vehicles

1. Introduction

The development of manholes is an ongoing trend spurred by urban growth. Though the construction of manholes is such an integral part of road development in the country, we see many incidents where the improper construction and maintenance of manholes have led to many road accidents, even resulting in the loss of lives. In India, at least two people die every day due to open manholes, according to [22]. These accidents occur due to broken manhole lids, improperly placed lids, or disproportional manhole lids which do not cover the manhole entirely. This problem is especially worsened during heavy rains wherein waterlogged roads can hinder even human driving.

Autonomous vehicles are not a reality yet in developing countries like India for many reasons as seen in [5, 27]. However, even seemingly trivial problems to human-driving could pose major obstacles to autonomous vehicles. One such problem that has not been accounted for in prior research is the existence of open or broken manholes which can lead to costly damages to the vehicle and pose a hindrance to public safety. Potholes are also a major problem, however the cost of damage to the vehicle would not be as high as that for open manholes. Thus, the focus of our research is manholes. Autonomous vehicles will soon be a reality even in developing countries [27] so assuming that the roads will have no problems with manholes could be catastrophic. Therefore, the inclusion of detection of open/broken manholes is imperative in obstacle avoidance and lane-changing

Machine GRAPHICS & VISION 32(1):25–53, 2023. DOI: 10.22630/MGV.2023.32.1.2 .

algorithms for self-driving vehicles. Humans can identify open manholes easily, however for autonomous vehicles this devolves into a computer vision problem.

In this paper, we focus on manhole detection and classification using computer vision. Autonomous vehicles have a camera and GPS for path planning and localization respectively. Using a camera, open or broken manholes can be localized and classified within an image, and using a GPS, the location of the manhole can be recorded to send to the concerned authorities. We collected a dataset of road-surface images and a few aerial images from Google Street View, Google Images, and [39]. A lot of the images in the dataset are from [39] and these images were collected using a moving vehicle and a GoPro HERO6 Black RGB camera with resolutions of 1280×720 pixels. Since the road-surface images were collected from a camera attached to a moving vehicle, it makes it very close to what an autonomous vehicle would encounter. The images include varied lighting conditions and can hence be used for real-time detection and classification of manholes on roads by self-driving vehicles.

The training images of the deep learning vision models we have trained, can also be used for detection on video streams. We also take this research one step ahead to further classify open and broken (improperly closed) manholes into high and low importance for determining the priority order of fixing the manhole. This can be used by private organizations or government authorities to easily find the faulty manholes, thus preventing unfortunate accidents.

In the past, image segmentation and classification based approaches have proven useful for crack detection [28] and pothole detection [21]. Considering this, and the fact that autonomous vehicles have cameras for driving, we focus on an object detection approach to manhole classification and localization. In this paper we evaluate and compare the different object detection models on the same dataset based on the computational complexity. The first approach we tested was classical computer vision for localization of the manhole in the images followed by a simple neural network for classification. This method is favorable for autonomous vehicles as it does not require a GPU or much computational power. In addition, our dataset was minimal with only 1032 images, thus we expected this method to be advantageous. However, it was not close to the expected benchmark, so we attempted simultaneous object localization and classification (object detection) using Convolutional Neural Networks (CNNs). Specifically, we first attempted YOLOv3-Tiny [1] which we hypothesized would give decent results without a heavy dependence on computational power. To compare the performance and the tradeoff between time and computational complexity, we applied YOLOv3 [37]. Vision Transformers (ViTs) [9] have gained a lot of popularity in object detection tasks in recent research (see [7]), thus we used YOLOS [12]. The results were similar to the classical computer vision approach, thus we implemented a simple, non-hierarchical vision transformer for just classification without localization, to evaluate its accuracy especially after recent research reported in [24] highlighting its promising applications as a backbone for object detection tasks.

In this paper, we go a step further than just object detection for localizing and classifying manholes. We propose a classical computer vision pipeline to measure the size of the hole in a broken manhole with respect to the manhole itself to analyze the level of damage done to the manhole. We take the images classified as open and improperly closed using the object detection model and crop them around the bounding box drawn by the object detection model. We then filter and find the maximum contour in the image and compare it with the smaller contour found. If any of the smaller contours are greater than 50% of the total manhole area, we determine that the vehicle should stop or definitely avoid the manhole. This is proposed because although broken and open manholes pose a risk of damage to autonomous vehicles, all of them cannot be avoided, especially when the traffic is heavy and speed of the vehicle is high. This pipeline acts as a form of direction to classify the level of damage into high and low importance, which would be useful for obstacle avoidance in autonomous vehicles, and it could even be useful for the respective authorities to determine which manholes are in need of immediate attention to prevent major accidents. As autonomous vehicles are a reality these days (though a minority), automation in reporting the improper manholes to the authorities in charge using the vehicles' camera and GPS itself is possible.

Thus, the main contributions and novelty of this research work are as follows:

1. In prior research work on autonomous vehicles, the problems that manholes pose, especially in developing countries, are not taken into account. Our work highlights this problem and proposes solutions for it. Open manholes are a tangible risk to public safety, hence, we focus on improperly closed and open manholes in this paper.
2. We conduct a thorough literature review on previous work in manhole detection in road surface images. We also include a theoretical review of object detection algorithms.
3. We propose and evaluate multiple approaches to manhole localization and classification considering the trade-off between computational complexity and accuracy.
4. We propose a novel pipeline for elliptical object localization, classification, and bounding box prediction using classical computer vision.
5. In prior art, only Convolutional Neural Networks (CNNs) have been considered till now for the purpose of manhole detection. We test Vision Transformers specifically for manhole detection in self-driving vehicles.
6. Using classical computer vision techniques, we propose a pipeline to determine the size of the hole in a broken manhole with respect to the manhole image itself (size detection).
7. The results of our research show that Vision Transformers are promising as backbones for object detection, however currently Convolutional Neural Networks can be integrated in obstacle avoidance techniques of autonomous vehicles.

The rest of this manuscript is structured as follows: Section 2 reviews the state-of-the-art research about manhole detection using computer vision approaches. This section discusses the primary research gaps identified and describes the role of our research. Our decisions regarding the approaches chosen for the methodology are also elaborated upon in this section. Section 3 investigates the methodology of the various object detection methods we attempted for manhole object detection and size detection. Section 4 discusses the results of the paper and summarizes the outcomes of the training and testing process. The results are analyzed based on computational complexity, accuracy, and speed. Section 5 reiterates the main results of this work and concludes the manuscript by identifying the future avenues of work.

2. Review of literature

Research on manhole covers has been done over the years, and the methods of research are ever-changing with the emergence of new technologies. The commonality between existing and upcoming research on manhole cover detection is that all the methods are based on digital imagery. Though traditionally, broken and open manhole covers are detected and fixed through manual surveys and crowd reporting, this method is laborious, time-consuming, and ill-planned. For this reason, there is a demand for methods of automation like classical computer vision and deep learning. The papers are organized by the date of publication to understand the flow of research methodologies.

2.1. Classical computer vision approaches for manhole detection in road-surface images

A morphological method (dependent on the structure) was developed in [42] (2000) for detecting round-shaped manhole covers. It involved a *black top-hat transform* for feature extraction designed with disc-shaped structuring elements. A masking operation with a thresholded input image was then done on the extracted round components. The small regions and the areas without any holes were eliminated from the final resulting manhole image.

Detection of obscure and textured circular objects were challenges faced by conventional methods of object detection. In [30] (2009), this drawback was overcome without the cost of learning patterns. This method was valid for even images with inhomogeneous contrast and noise as it analyzed the separability and uniformity of intensity distributions using the Bhattacharyya coefficient filter, rather than the conventional method at the time, i.e., analyzing the difference in intensity levels of the object interior and surroundings. Separability in this paper is defined such that it can handle image feature distributions that are not normal distributions.

In [17] (2014), manhole cover detection using vehicle-based multi-sensor data combining multi-view matching and feature extraction was developed. Close range images using GPS/IMU and LIDAR data were obtained. It involved two main steps – edge detection and texture recognition. Scene segmentation to eliminate cars and pedestrians was done, and on the segmented data, a custom edge detection algorithm based on Canny edge detection [6] (1986), which was sensitive to arcs and ellipses, was used. Arc-containing regions were fitted to an ellipse.

An algorithm for automatic recognition of manhole covers based on images from the Mobile Mapping System (MMS) was proposed in [8] (2016). The images were collected using the MMS and preprocessed by image enhancement using gray-scale transformation and filtering technology, followed by the double threshold method in Canny operator edge detection. Hough transform based on the rough localization of ellipse geometry was used to locate the accurate positions of manhole covers in the preprocessed images.

In [50] (2020) the author concentrated on the detection of manhole covers using texture-based image segmentation and elliptical fitting. To extract textural features, the Laplacian of Gaussian filter's performance was contrasted with that of the Gabor filter. To divide the pixels in the image into distinct regions, the K-means technique with sum of square error was employed, and the least-squares approach was utilized to process the ellipse fitting.

In all of the above papers, only the localization of a manhole in the image is provided. Additionally, classification of manholes or differentiating between the various objects within the image is not attempted. The true sense of object detection as we know it now, is classification+localization. The above papers do not focus on this and thus, our paper attempts to fill this research gap by implementing a classical computer vision approach to manhole *detection* in road surface images. Our approach models conventional deep learning based approaches by providing a bounding box and coordinates for localization, and classifies the manholes too. Multiple bounding boxes are predicted in the localization step as in deep learning models, but the mean of the best fits is taken for the final classification. Additionally, like in all of the papers reviewed above, we extract the shape of the manhole from the image using ellipse fitting.

2.2. Machine and deep learning approaches for manhole detection in road-surface images

In [43] (2011), a multi-view method implementing 2-D and 3-D techniques for manhole mapping based on vision and GPS was presented. The position and inclination of the ground plane were estimated to generate front-to-parallel 2-D views. Single-view processing involved the application of a cascaded framework composed of mean-shift color segmented area, aspect ratio, intensity variance, radial symmetry, and texture-based filters to the 2-D views. The object detection system used for identifying manholes in single-view processing was Local Binary Pattern feature vectors [31] and Discriminatively

Trained Part-Based Model [14]. Multi-view processing involved fusing and grouping the results of single-view processing into 3-D hypotheses which were then fed into a graph-cut segmentation filter, and finally used for accurate localization of the manholes.

Automated detection of manholes using Mobile Laser Scanning (MLS) data was put forward in [52] (2015). The road surface images were segmented by detecting curbs in the images and using them as reference points for segmentation. These images were converted to raster images with georeferencing and intensity information using Inverse Distance Weighted (IDW) Interpolation. The high-order features of images were depicted by a multilayered feature generation model which was built on a vision based deep learning model. A random forest model was then trained to learn how these features are mapped to the probability of existence of manhole covers at specific locations. The manhole covers were then detected in the previously rasterized images using both models.

In [51] (2019), deep learning was suggested for the autonomous extraction of tiny objects in urban environments. A Mobile Mapped System was used to gather a dataset for Urban Element Detection (UED) that included manholes, milestones, and license plates. The faster R-CNN framework was tuned for small object identification, and a feature extraction CNN network named SlimNet with six convolutional layers and three max-pooling layers was developed. The performance of these networks on the collected dataset was compared to other existing deep networks. The findings of this paper concluded that the SlimNet model had the highest accuracy.

In [4] (2019), the Automated Localization of urban drainage infrastructure from public-access street-level images was done. A dataset of manhole and storm drain images was captured using the Google Street View API and annotated. The Faster R-CNN deep learning meta-architecture with Resnet 101 as the feature extractor backbone was tested on this dataset. Localization was done to project the coordinates from image space to geographical coordinates.

Mapping manholes using the deep learning method RetinaNet in road-level RGB images was done in [39] (2020). ResNet-50 and ResNet-101, being the two different feature extractor networks for the RetinaNet method, were used to experimentally test the method. The results of this test were then compared with the Faster R-CNN method. However, the findings of the paper concluded that the RetinaNet method was far more effective than the Faster R-CNN method for mapping manholes.

A method was proposed in [10] (2020) to convert RGB image data and extracted contours using an object detection algorithm. Pavement distress was classified using the YOLOv3 (You Only Look Once) algorithm which is a one-stage detection algorithm that does not need the region proposal phase. A large-scale dataset was prepared, containing images taken in various weather and illumination conditions. The image data was analyzed effectively by using Average Precision (AP) as the indicator on the data.

In [15] (2021), two deep learning techniques were implemented for automated pavement distress detection and classification, namely Faster Region-based Convolutional Neural Networks (R-CNN) and YOLOv3. These deep learning frameworks were trained on a dataset the authors collected and validation accuracy was indicated using Average Precision (AP) and Receiver Operating Characteristic (ROC) curves. By contrasting the suggested model with manual quality assurance and quality control (QA/QC) results received on automated pavement data, the models were assessed.

In all of the above papers, only convolutional neural network based architectures have been researched for the task of manhole detection. In our paper, we also focus on manhole detection with vision transformers as they have gained popularity for object detection tasks. Additionally, the networks modeled in the above papers have not been tested for the detection speed. Since in this paper we attempted the task of manhole detection for autonomous vehicles, the computational power requirement and speed is a significant factor we took into account. This is the main reason why we attempted YOLOv3 and YOLOv3-Tiny object detection networks in our paper.

2.3. Theoretical review of object detection algorithms

Since we adopt deep learning based object detection approaches, we also include a review of the state of the art object detection models. The three main approaches towards object detection are:

1. Classical Computer Vision,
2. Convolutional Neural Networks,
3. Vision Transformers.

We attempt networks from all three of these categories in our paper.

Localization algorithms using classical computer vision have been explored to a large extent as seen from all the previous papers reviewed. The following papers give examples of classical computer vision being used for object localization and classification including bounding box calculation. In [26], the authors use a variant of Hough Transform for localization and Machine Learning for classification to perform object detection. Max Margin Hough Transform was used for localization and bounding box prediction, and classification was done using an SVM based classifier. In [20], classical computer vision was combined with machine learning to create a fully automated hybrid cell-detection model named CIRCLE. The images were first processed to extract various tiles from them. MaskRCNN was then applied to them to detect the cells. These works encouraged us to test classical computer vision approaches especially to focus on the detection speed for autonomous vehicles. However, we do not follow the same methodology for object detection as the above two papers; instead we propose a novel pipeline.

Convolutional Neural Networks have been popular in object detection for a long time now. The YOLO algorithm [35] is popular and YOLOv3 is the model we chose for our research. YOLOv1 [35] is a one-stage detector which uses the Darknet framework [2] and is trained on the ImageNet-1000 dataset [13]. It splits a given image to a grid of $S \times S$ cells. For every cell in the grid, it computes confidence for n bounding boxes. The predicted result is encoded into a tensor of dimensions $S \times S \times (5n + p)$, wherein the input image is divided into $S \times S$ sub-images, the $5n$ term corresponds to five attributes of the bounding box that must be detected (center coordinates, height, weight, and confidence score). The p term represents the probability of the object in the image belonging to a particular class. YOLOv1 had difficulty with detecting small objects and when the dimensions of the testing images varied when compared to the training images [1]. YOLOv2 [36] improves upon this by introducing batch normalization in every convolutional layer. YOLOv3 [37] further improves upon this by using independent logistic classifiers for multilabel classification instead of multiclass classification when using softmax. This improves the model as using softmax imposes the assumption that each box has exactly one class which is often not the case. The key novelty of the YOLOv3 algorithm is that it makes its detections at three different scales. YOLOv4 [3] furthers upon YOLOv3 to introduce a new architecture with a backbone, neck, dense prediction, and sparse prediction. The backbone and dense prediction networks are similar to that of YOLOv3 (the backbone is changed to Cross Stage Partial Network (CSPNet) [47, 48]), and the neck is a novel idea to add layers in between the backbone and dense prediction block. The layers added to the neck are a modified Path Aggregation Network (PANet) [25], a modified spatial attention module, and a modified spatial pyramid pooling, which are all utilized to combine the data in order to increase accuracy. The CSPDarknet53 backbone [47] eliminates repetitive gradient information in big backbones and incorporates gradient change into a feature map that speeds up inference, improves accuracy, and shrinks the size of the model by reducing the number of parameters. YOLOv5 [18,19] utilizes the same CSPDarknet53 as backbone. The Path Aggregation Network in YOLOv5 is different and adopts a new feature pyramid network (FPN) that includes several bottom up and top down layers. The model's low level feature propagation and localization precision are both enhanced by this PANet. The localization accuracy of the object is increased because of PANet's improved localization in lower levels. In YOLOv7 [46], the PAnet is replaced by Extended Efficient Layer Aggregation Network (EELAN) which uses group convolution to enhance the features learned by different feature maps and improve the use of parameters and calculations. In addition, compound model scaling is used. This is scaling the width (number of channels) and depth (number of layers) in coherence for concatenation based models. A summary of the architecture of these models can be found in Table 1.

Transformers were first proposed for Natural Language Processing tasks in [45]. In [9], the authors propose a Vision Transformer called ViT which closely models the original transformer architecture of [45] as closely as possible for image classification tasks. Images are first flattened into 2D patches to be passed to the transformer's encoder network.

Tab. 1. Table from [29] with an additional column added for YOLOv7.

	YOLOv3	**YOLOv4**	**YOLOv5**	**YOLOv7**
Neural Network	FCNN	FCNN	FCNN	FCNN
Backbone Feature Extractor	Darknet-53	CSPDarknet53	CSPDarknet53	CSPDarknet53
Loss Function	Binary Cross Entropy Loss	Binary Cross Entropy Loss	Binary Cross Entropy Loss and Logits Loss Function	Binary Cross Entropy Loss
Neck	FPN	SSP and PAnet	PAnet	EELAN
Head	YOLO Layer	YOLO Layer	YOLO Layer	YOLO Layer

The transformer's encoder consists of alternating layers of Multi-headed Self Attention (MSA) and Multi Layer Perceptron (MLP) blocks. Layernorm (LN) and residual connections are applied respectively before and after every block. The MLP contains two layers with a GELU non-linearity. Due to its accuracy on ImageNet, in [7], a Vision Transformer was used for object detection to develop the DETR model. The DETR model uses the feature maps extracted by a CNN backbone as input for the transformer encoder-decoder architecture for transforming feature maps to features, followed by a single feed forward neural network for prediction. It uses a bipartite matching loss function for matching between the predicted tokens and ground-truth objects. In [24], instead of using a CNN backbone for object detection as done in [7], the authors explore using a plain Vision Transformer like vanilla ViT as a backbone for object detection. In [12], the authors propose YOLOS, an object detection model which uses plain Vision Transformers ViT and DeiT [44] as the backbone. While YOLOS chose a Transformer with an encoder-only architecture similar to ViT, DETR used a Transformer encoder-decoder architecture. For each encoder layer, YOLOS always examines a single sequence without making a distinction between the tokens in terms of operations, where the tokens are the learnable embeddings in the image.

For our work, we chose YOLOv3 and YOLOv3-Tiny as the networks to compare to YOLOS. We wanted to compare the performance of a *tiny* object detection model as it can be used for autonomous vehicle research. YOLOv4 and YOLOv3 have *tiny* object detection networks and as seen in the above figure, the authors of YOLOS also released a YOLOS-Tiny model. That is why these networks were chosen for our research work. Also, YOLOS is a completely transformer based architecture with a transformer backbone, and YOLOv3 is a completely CNN based architecture which makes it a good point of comparison.

3. Methods

3.1. Functional Block Diagram

Fig. 1 and Fig. 2 describe the proposed methodology's general flow. A visual explanation of some of the methodology's key decision points is provided by the flowchart. This modular structure makes it simple to plan for future changes to functionality and flow that will boost efficiency and allow integration with additional methods.

3.2. Dataset

We mainly used road surface images for this study due to its application in autonomous vehicles. However, few close-up aerial images were added to the dataset as well. Also in support of road-surface images, in [4] it was indicated that street-level imagery could provide useful information to identify manholes that could not be detected in aerial images. In this dataset, we focus only on round manhole covers. This is as most manhole covers are round, for the purpose that orientation when placing the cover is not an issue. At the same time since manholes weigh around 250 pounds [11], it is easy to roll them in case of replacement.

We have used the dataset publicly provided in [38] and described in [39].

A total of 1032 images were collected and annotated. Data augmentation in the form of 90° rotation, and saturation value change were performed in the dataset to obtain a dataset containing 2673 images. This was carried out to balance the number of images per class, avoid overfitting, and enhance the deep neural networks' performance during model training. The images were split into three different classes in an unbalanced fashion: closed manhole improperly closed manhole open manhole

Since we used the same dataset for classification and object detection, it is available in two different formats. In the format for classification, all the images have been divided into folders based on the class label without any annotation file as required in the classical computer vision object detection method. The drawback of this dataset is that images which have multiple objects each of different classes, the objects cannot be classified separately in the same image. All of the images were manually annotated in the format for deep learning-based object detection by marking rectangles (bounding boxes) around the manholes and categorising each rectangle by the corresponding class. This was done using the `labelImg` tool [23]. Images which have multiple objects, each of different classes, can be classified separately in the same image.

For training, these images were divided into three groups for training, testing, and validation. The train-test-validation split is done randomly with no manual intervention. As the dataset is minimal already, we increased the number of training samples to contribute to a more robust evaluation. The number of images in each of the sets are as follows:

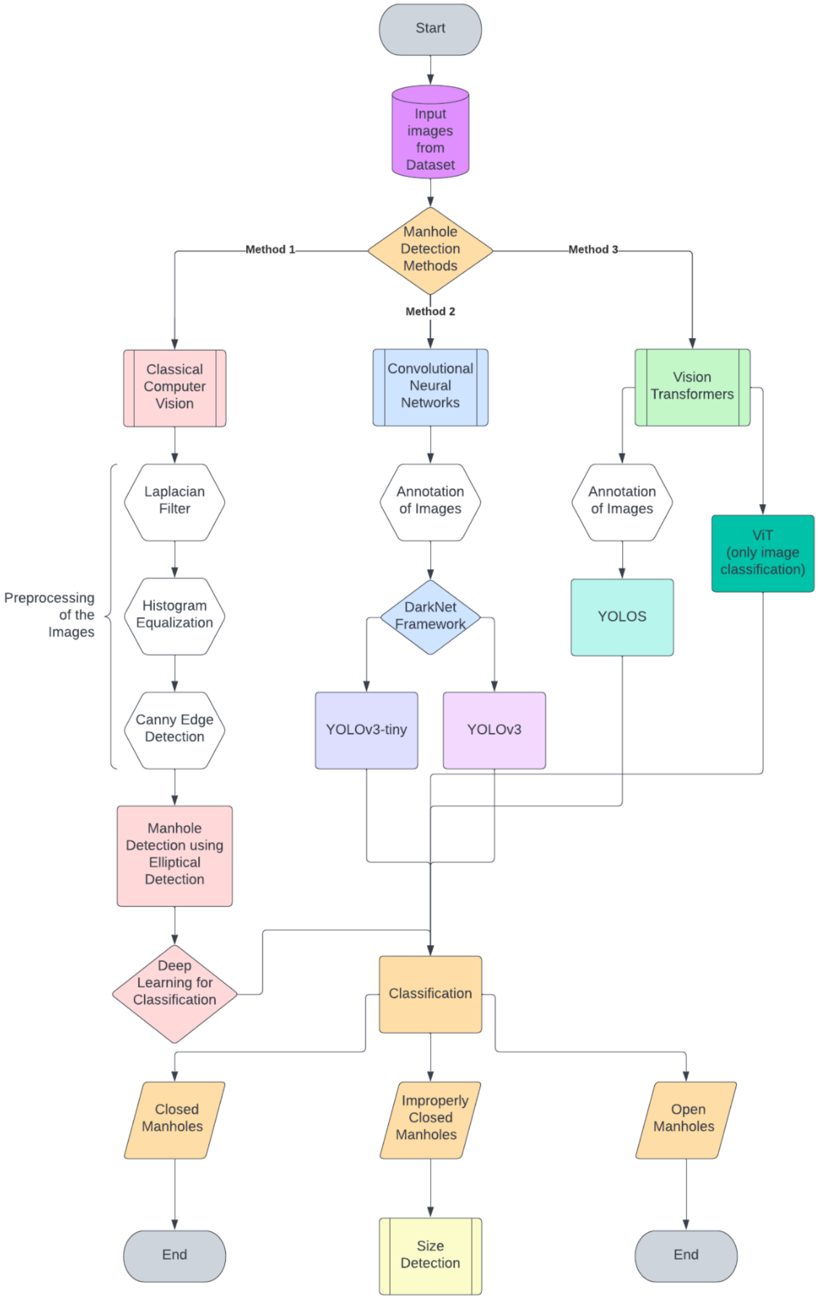

Start
Input images from Dataset
Manhole Detection Methods
Method 1
Method 2
Method 3
Classical Computer Vision
Convolutional Neural Networks
Vision Transformers
Preprocessing of the Images
Laplacian Filter
Histogram Equalization
Canny Edge Detection
Annotation of Images
DarkNet Framework
YOLOv3-tiny
YOLOv3
Annotation of Images
YOLOS
ViT (only image classification)
Manhole Detection using Elliptical Detection
Deep Learning for Classification
Classification
Closed Manholes
Improperly Closed Manholes
Open Manholes
End
Size Detection
End

Fig. 1. Block diagram of the methodology

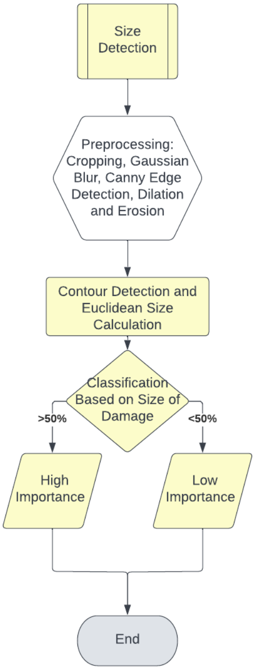

Size Detection
Preprocessing: Cropping, Gaussian Blur, Canny Edge Detection, Dilation and Erosion
Contour Detection and Euclidean Size Calculation
Classification Based on Size of Damage
>50%
<50%
High Importance
Low Importance
End

Fig. 2. Block diagram of the size detection algorithm

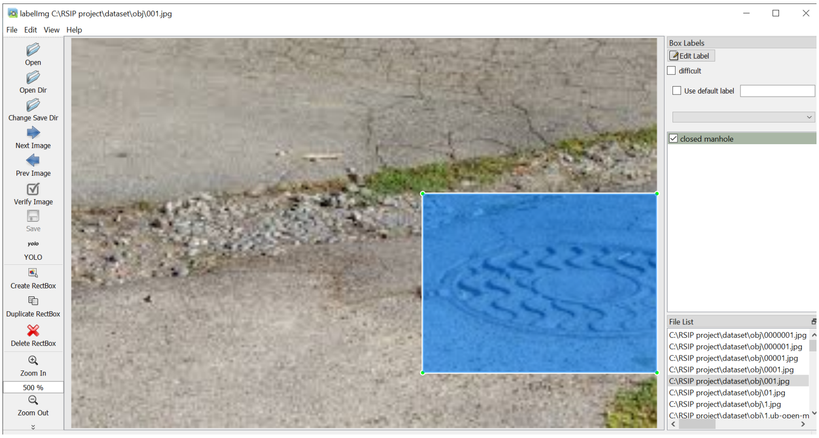
1. Training – 2469
2. Testing – 71
3. Validation – 133

Fig. 3 depicts an example of an image which has been annotated with a bounding box drawn around the manhole.

3.3. Methodology

3.3.1. Object Detection

In Section 2 in the Review of Literature, we elaborated on the architectural differences between the three main approaches towards object detection. As described in the review of literature, we implemented classical computer vision approaches, CNN-based YOLOv3 and YOLOv3-Tiny, and Vision Transformer based YOLOS. We evaluated the performance of each of these models on our dataset to determine the best for manhole detection and classification specifically for autonomous vehicles. A summary and comparison of the different methods used in this paper is available in Table 2.


labelImg C:\RSIP project\dataset\obj\001.jpg

☑ closed manhole

File List

C:\RSIP project\dataset\obj\0000001.jpg
C:\RSIP project\dataset\obj\000001.jpg
C:\RSIP project\dataset\obj\00001.jpg
C:\RSIP project\dataset\obj\0001.jpg
C:\RSIP project\dataset\obj\001.jpg
C:\RSIP project\dataset\obj\01.jpg
C:\RSIP project\dataset\obj\1.jpg
C:\RSIP project\dataset\obj\1.ub-open-m

Fig. 3. Example of an annotated image.

Classical Computer Vision approach to manhole detection

In this method, we propose a novel pipeline for manhole detection in road-surface images. This method was attempted as it does not require specialized hardware like a GPU and the computational complexity of the algorithms is not high especially in comparison to deep learning approaches. Since we are evaluating the performance of the algorithms for self-driving vehicles, we expected this method to be advantageous. In addition, the dataset collected is minimal for deep learning tasks so we proceeded with this approach. The code for this approach was implemented with MATLAB.

1. **Pre-processing**

   In classical computer vision approaches, the filtration and preprocessing stage is an essential part of object detection. We combined multiple approaches in order to get a high level of accuracy. For filtering, we employed the Laplacian filter [16]. This filter was chosen since it preserved the edges while reducing the noise. This filtering was followed by histogram equalization to improve the overall contrast of the image as low contrast regions are brought closer to the average contrast value. Next, Canny Edge Detection [6] was done to detect the edges of objects in the images. We applied this technique to extract the morphological information from the images while also reducing the data to be processed before performing manhole localization.

   Fig. 4 shows an example of how an image looks after preprocessing.

2. **Manhole Localization**

   The geometrical circular detection filter is based on the approach proposed in [30] and adapted in [32]. However, since the manholes in road surface images have a mostly

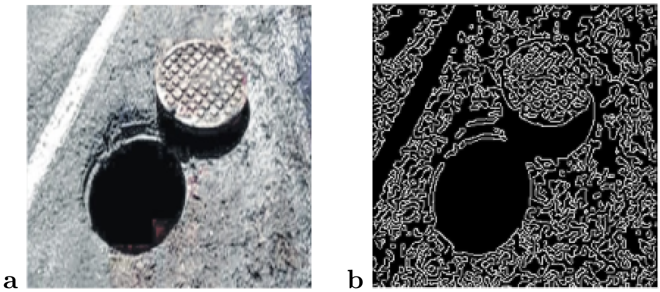


Fig. 4. An example of the effect of preprocessing. (**a**) Original image; (**b**) pre-processed image.

elliptical shape, we used an elliptical filter. The code for the ellipse detection is adapted from [41]. It can be used for circle detection also. The theory behind this approach is that for an ellipse, there are five unknown parameters, (x_0, y_0) for the center, (a, b) for the major and minor axes, and α for the orientation. In [49], the author proposes a method using only a 1D accumulator array to accumulate the length of the minor axis of the ellipse. Using this method, the four coordinates of the major and minor axes of the ellipse were calculated.

The shapes that were the best fit as per the definition of an ellipse were taken. In the case of more than one best fit, the mean of the best fits was taken to be the final ellipse. Once the ellipse was identified, two rectangles were identified taking each of the major and minor axes of the ellipse as diagonals. In order to make sure that no part of the ellipse is cut out, the final rectangle was taken so that it bounds both of the rectangles created. The coordinates of the rectangular bounding box drawn around the manhole in the images are obtained. Multiple manholes in the same image can be localized this way.

A mask was created such that the pixels in the area covered by the final bounding rectangle all had a value of 1, and those not covered by the rectangle all had a value of 0. The original image and the mask were combined, finally showing only the manhole. There were some cases in which the model failed to identify a rectangle showing only the desired part of the image. To ensure that this was not a problem, we allowed the original preprocessed image itself to be taken as the final product in such a case wherein the image was already cropped enough and the manhole was in the region of interest. Fig. 5 shows examples of ellipses drawn around manholes in each case: closed manhole, open manhole and broken manhole.

In our attempt to ensure that as much as possible of the manhole is retained while the unnecessary information is deleted, we had chosen to take the mean of the best fits (as many possible ellipses were found) as the final ellipse, and used these coordinates to draw a bounding box. Using the mean of the best fits may have resulted in the

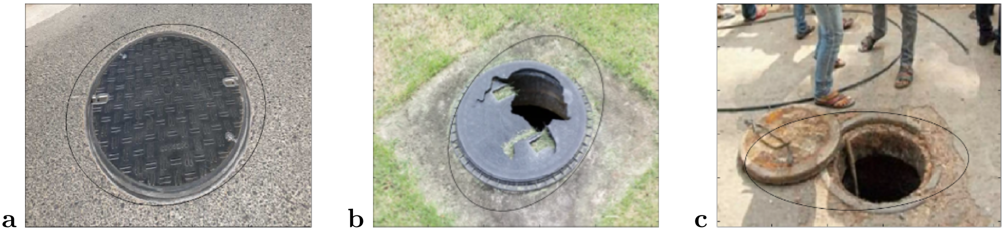


Fig. 5. Samples of the ellipse drawn by the algorithm around each class: (**a**) closed, (**b**) broken, and (**c**) open manhole.

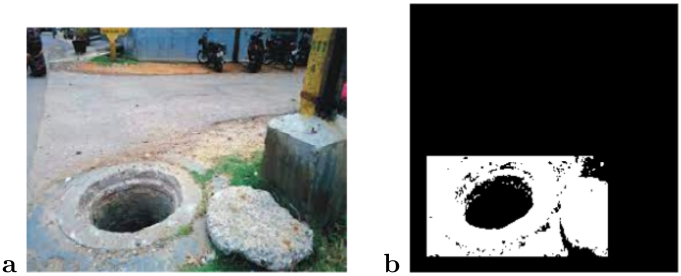


Fig. 6. Bounding box drawn around the manhole using the ellipse coordinates. (**a**) Original image; (**b**) manhole detected (the bounding box is the region with white background).

inclusion of some extra pixels, as seen in Fig. 5; the ellipses drawn were not perfectly accurate to the edge. However, this does ensure that none of the necessary information for training the deep learning model was deleted, while a majority of the unwanted material was effectively removed.

Instead of drawing an ellipse around the manhole as done in Fig. 5, we can use the coordinates of the ellipse directly to obtain the coordinates for the bounding box. In the algorithm we used, the pixels inside the bounding box are left unchanged while the values of the pixels outside the bounding box are all equated to zero.

The resulting image after the bounding box is defined is shown in Fig. 6 for another manhole from our dataset. The left and right most x-coordinates and top and bottom most y-coordinates were calculated from the major and minor axes coordinates of the ellipse defined. As seen in the image, the pixels inside the box are left as they were while all the other pixels in the image were made black. The time taken for localization of all of the images was around 6 hours on a CPU.

3. **Manhole Classification**

The images obtained after localization were taken for the deep learning model. Since it was for simple classification, we used a basic Convolutional Neural Network architecture. We used 3 convolutional layers, 2 max pooling layers followed by a fully connected layer. ReLu was taken as the activation function for the convolutional layer and softmax for the final classification. Batch Normalization is used after every Convolutional Layer as used in YOLOv3 [37]. The number of epochs was set to 100 and the number of iterations per epoch was set to 6. The training time was 68 minutes.

Convolutional Neural Networks approach to manhole detection

In this method, we used YOLOv3 and YOLOv3-Tiny for manhole detection. As the performance of the classical computer vision algorithm could be improved greatly, we implemented a convolutional neural network based approach notwithstanding the computational intensity. There are many other state-of-the-art object detection models, however we chose to analyze the performance of YOLOv3 as it has a YOLOV3-Tiny model in conjunction. We hypothesized that the YOLOv3-Tiny model would especially prove useful for self-driving technology as it has a higher FPS (frames per second) rate as per [35]. In addition, YOLOv3-Tiny was attempted as it is not computation intensive and is a small model meant for constrained environments. The code for this method was implemented in Python.

YOLO is a deep Convolutional Neural Network based one-stage object detector. YOLO only looks at an image once to predict whether the object is present and where it is located in the image. YOLO implicitly encodes contextual and visual information about classes as it views the entire image during training and test. Object detection in YOLOv3 is framed as a logistic regression problem to separate the bounding boxes in the image and associate class probabilities with each bounding box [35]. A single neural network predicts the coordinates of the bounding box using dimension clusters as anchor boxes. YOLOv3 uses K-means clustering on the dataset to determine bounding box priors automatically rather than manually. The probability of the object class and the confidence of the object in the bounding box is captured in one evaluation itself. The probabilities for each class are calculated using independent logistic classifiers rather than the softmax function; thus, each bounding box can belong to several classes. Thus, as the detection pipeline is a single network and end-to-end optimizations is performed to improve detection, the unified architecture of YOLOv3 is extremely fast and robust running at 100 FPS (frames per second). YOLO v3 uses upsample, downsample, and fusion methods to independently detect objects on multiple scales of fusion feature maps. This method is especially effective in detecting small and near-distance target objects; thus, it's suitable for manhole detection.

We adopted the DarkNet-53 (53 convolutional layers) neural network framework as the backbone for YOLOv3. We adopted transfer learning, so our models' weights were initialized with weights from the network trained on the MS COCO dataset [37]. We adopted this approach in our project as transfer learning enables training of neural networks with lesser data, the accuracy is high, and the training time is reduced. We used the source code available in [2] for our implementation. The model was trained and tested on Google Colaboratory with an NVIDIA Tesla T4 Graphics Card (2560 Compute

Unified Device Architecture (CUDA) cores and 16 GB graphics memory). The number of iterations was set to 6,000 (as specified in [2]) for both YOLOv3 and YOLOv3-Tiny. The classes, number of filters, steps, jitter, etc., were also customized to our requirements in the config file. As it's an open-source framework, we tuned a few hyperparameters to customize the network to our task as documented in [2]. However, we adopted *early-stopping* to prevent overfitting. Also, due to the time it takes to run the program, and because of the desirable mAP value obtained, we stopped training the model at 3000 iterations for both networks. The only difference in training between the two models YOLOv3 and YOLOv3-Tiny was the configuration file used. The YOLOv3 model ran for 3035 iterations. YOLOv3 took 5 hours and 15 mins for it to train, even with a GPU that supports fast computation. Thus, it's a very time-consuming process to train. YOLOv3-Tiny ran for 3020 iterations.

Vision Transformer based approach to manhole detection

In this method, we used YOLOS [12], a Vision Transformer based model for object detection. Vision Transformers have recently gained a lot of popularity in object detection tasks, according to [7]. Out of all of the Transformer based object detection models, we chose YOLOS as it uses a Transformer as the backbone. Also since it is a single sequence based architecture, we wanted to compare it to the YOLO which is also a one-shot object detector. The code for this method was implemented in Python.

YOLOS uses a *plain vanilla* Vision Transformer (ViT) as its backbone [12]. The difference between ViT and the backbone for YOLOS is that YOLOS drops the CLS token used for image classification and adds 100 randomly initialized detection tokens [DET] to the input patch embedding sequence. The CLS token indicates that the training task is classification. The [DET] token is a learnable embedding for object binding. Position embeddings are added to all of these input tokens to retain positional information. Another difference between ViT and YOLOS' backbone is that the image classification loss used in ViT is replaced with a bipartite matching loss to perform object detection similar to DETR [7]. During training, YOLOS produces an optimal bipartite matching between predictions from the one hundred [DET] tokens and the ground truth objects. During inference, YOLOS directly outputs the final set of predictions in parallel.

The randomly initialized detection [DET] tokens are used as substitutes for object representation. This is done to avoid inductive bias and any prior knowledge of the task that can be introduced during label assignment. When YOLOS models are fine-tuned on the COCO dataset, an optimal bipartite matching between predictions generated by [DET] tokens and the ground truth is established for each forward pass. This serves the same purpose as label assignment but is completely unaware of the input 2D structure, or even that it is 2D in nature. These steps to reduce the inductive bias are imperative as some of the YOLO algorithms' (CNN based YOLO family) performance reduces when the dimensions of the testing images vary when compared to the training images, as stated in [1].

We trained YOLOS for 150 epochs for batches of 8 and monitored the validation loss when training. The images were tested for a box confidence score of 0.2. The model was trained and tested on Google Colaboratory with an NVIDIA Tesla T4 Graphics Card and it took around 9 hours to train. The accuracy of YOLOS is not very accurate, as stated in [12], and we noticed this too. Thus, we attempted plain Vision Transformer ViT with no hierarchical backbone as the authors of [24] pointed out that vanilla ViT could be used as a backbone for classification in object detection networks with good accuracy. We tried ViT on our dataset to check if Visual Transformers were promising for manhole classification as YOLOS uses this as its transformer backbone. We trained the ViT model for 30 epochs in a batch size of 20. The total time taken for training was around 15 minutes with a Tesla T4 GPU.

3.3.2. Size Detection

The manholes which were detected as improperly closed through the object detection model were run through a size detection algorithm. The algorithm detects the size of the hole in a broken manhole with respect to the size of the manhole itself in an image. We used classical computer vision using python and OpenCV for size detection.

1. **Preprocessing**

   The manually annotated images from the dataset were first converted from the YOLO to the COCO annotation format. Using those coordinates, we automated the process of cropping the manholes identified as improperly closed around their bounding box to avoid unnecessary visual information.

   The cropped images were then Gaussian blurred. This step was done to reduce the extraneous visual information in the image. The image was then converted to grayscale to apply Canny edge detection which enabled us to outline the manhole and the broken hole. After this, the image was *opened* (dilation+erosion).

2. **Algorithm**

   Canny edge detector detects all of the edges in an image, but we need only the manhole and the broken hole. Hence, we applied contour detection methods. This method was preferred over ellipse detection methods like the Hough Transform as a broken manhole is not elliptical in shape anymore. Additionally, the broken part of the manhole is also not necessarily elliptical in shape. The manhole was identified by detecting the maximum contour in the image. We draw a bounding box around the contour of the manhole. Using the coordinates of the ordered bounding box, we compute the midpoint of the top left and top right coordinates and the midpoint of the bottom left and bottom right coordinates. We calculate the Euclidean distance between the midpoints using the distance formula in terms of pixels. We follow the same steps for all of the contours found in the image except the max contour. If any of the contours found in the image are greater than 50% of the max contour (i.e. the entire manhole), we classify that manhole as a *High Importance* manhole in need of

Tab. 2. Summary and comparison of the object detection methods used in this paper.

Approach	**Automated Method**	**Localization Method**	**Classification Method**	**Pros of the Approach**	**Cons of the Approach**
Classical CV	Automatic Annotation (just classified images into classes)	Classical CV	Custom Neural Network	– Localization is done in the shape of the manhole before drawing the conventional bounding box. – The images are pre-classified into the classes. Automated annotation of the dataset. – Time taken was around 7 hours for all of the images.	– Not very accurate. – Localization works for multiple manholes in the same image. Cannot classify each of the manholes in the localized image with the deep Neural Network however.
YOLOv3 and YOLOv3-Tiny (CNN)	Manual Annotation	Simultaneous localization and classification (CNN)	Simultaneous localization and classification (CNN)	– Accurate. – Can localize and classify multiple instances of manholes in the same image. – Gives the probability of the object class in the box. – Time taken to train was 3, 6, 9 hours respectively to train the 3 models.	– Requires specialized hardware (GPU). – Annotation is manual.
YOLOS	Manual Annotation	Simultaneous localization and classification (Vision Transformer)	Simultaneous localization and classification (Vision Transformer)		

immediate attention by the concerned authorities. The rest of the improperly closed manholes are classified as low importance. Thus the method of ranking according to safety is:

Open manholes > Improperly closed manholes whose broken parts are greater than 50% of the entire manholes > Rest of the Improperly closed manholes.

Size detection did not take a lot of time. Contour detection methods are fast and have already proved useful in developing computer vision algorithms for real-time autonomous vehicles [34].

4. Results

4.1. Object Detection

Classical Computer Vision approach to manhole detection

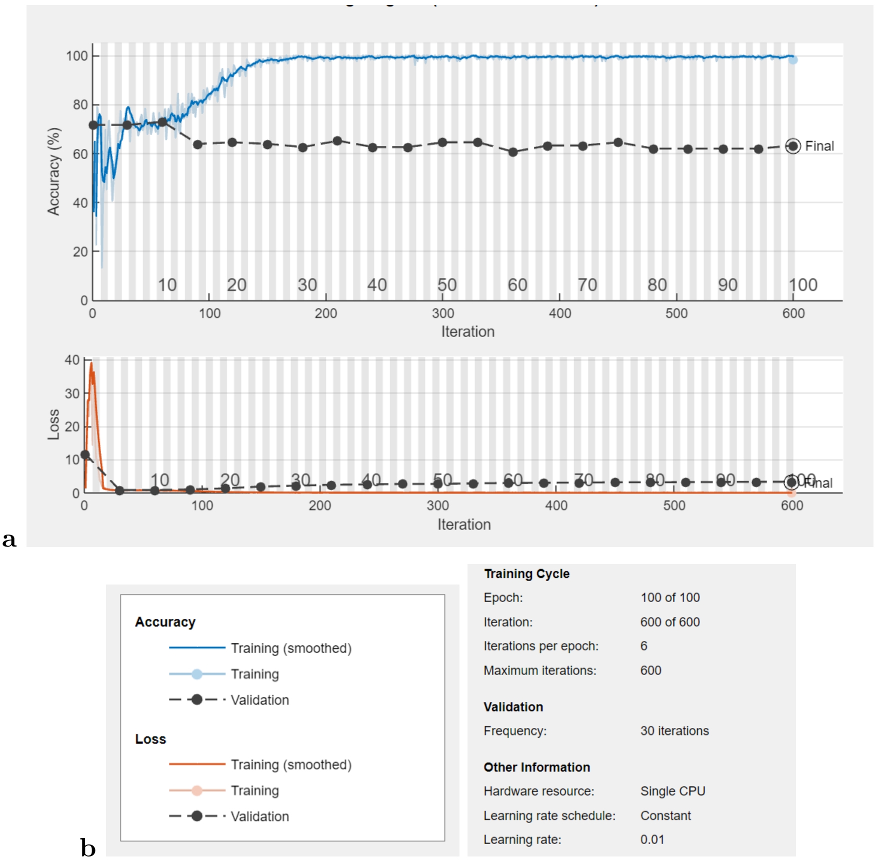
The metrics used to evaluate the classification accuracy of the neural network were Testing/Validation Accuracy and Loss. An image of the graph at the end of 100 epochs and 6 interactions per epoch is shown in Fig. 7. According to the graph, Testing/Validation Accuracy is 63.23%. The images chosen for evaluating the accuracy were randomly selected 150 images from the validation and test datasets. As there was no hyperparameter tuning and the the images from the validation dataset were not shown to the model more than one time, this served as the Testing Accuracy though the images from the validation dataset were used. The model was trained as the accuracy percentage increased and the loss percentage decreased. Training accuracy finally reached 100%.

Convolutional Neural Networks and Vision Transformer approach to Manhole Detection

The performance of DarkNet YOLOv3, YOLOv3-Tiny, YOLOS (for object detection), and ViT (for just classification) were assessed by precision–recall curves and the average precision (AP). The Intersection over Union (IoU) was calculated in order to evaluate the precision and recall. According to well-known competitions in object detection, a true positive (TP) is for IoU ≥ 0.5 and a false positive (FP) for IoU < 0.5. Based on the above metrics, precision (P) and recall (R) are estimated. The average precision (AP) is estimated by the area under the precision–recall curve. The Mean Average Precision (mAP) was used as the primary indicator to analyze the accuracy of the models as seen in Fig. 8. The AP is calculated for each of the classes and it is averaged over all of the categories to get the mAP. We trained the models while the mAP percentage increased and the validation loss decreased. The mAP percentage value was calculated during training. The loss curves also indicate accuracy, however, mAP is considered to be a better indicator [37].

The weights files were generated for every 1000 iterations. The best weights obtained were saved and can be used for further testing and manhole classification of all 3 object detection models. At some points in the graph even though the mAP percentage reduces, we still continued training as the recall values were still increasing [35]. The average precision (AP [%]) and its mean values (mAP [%]) obtained from the area under the curve for both of the models are illustrated in the above figure. Table 4 displays the results at an IoU cutoff at 0.5 (AP50). The FPS rate was calculated as given in [40] with a batch size of 1. The size of the images in the image stream was taken as 256×256 for testing purposes.

Thus we realized that the YOLOv3 model had the best accuracy but YOLOv3-Tiny



Training Cycle

Epoch:	100 of 100
Iteration:	600 of 600
Iterations per epoch:	6
Maximum iterations:	600

Validation

Frequency:	30 iterations

Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.01

Fig. 7. (**a**) Training and validation accuracy and loss curves for the classification network of the classical CV approach. (**b**) Key for (a) and other information.

seems like the best model for manhole detection from the autonomous vehicles standpoint. Though the accuracy of the YOLOS model is not large, the ViT classification network gives good results. This indicates that further research into object detection models using transformers could prove promising. In terms of the computational requirement and speed too, CNNs and classical computer vision could be used.

Fig. 11 and Fig. 12 show examples of detected and classified manholes.

Since the accuracy of YOLOS was not high, we trained ViT on the same dataset just to evaluate the classification accuracy. The results in the form of the confusion matrix are shown in Fig. 13.

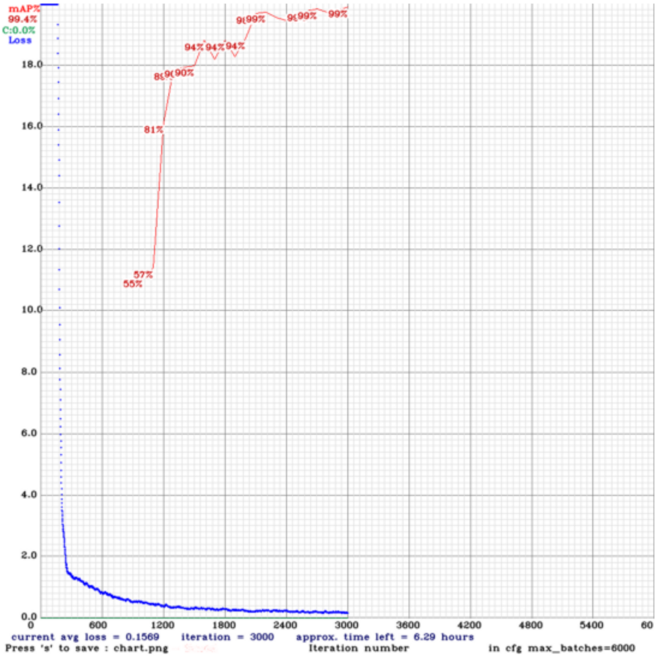


Fig. 8. mAP graph for YOLOv3.

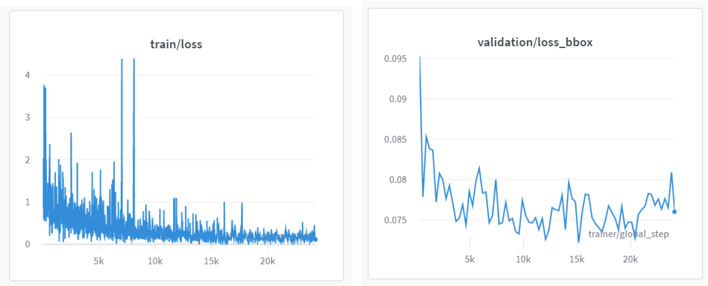


Fig. 9. Loss Graph for YOLOS.

4.2. Size Detection

We tested the size detection model on 26 improperly closed manholes which contained partially open manholes and broken manholes. We also included 2 open manholes and 2 closed manholes for testing purposes. The evaluation of this system model was done by manual classification as in, we cross verified if what we determined as high importance matched the size detection model's output. We determined that out of the 26 improperly

Tab. 3. Results of object detection models tested.

Network	**Confidence Threshold**	**Precision**	**Recall**	**F1 score**	**mAP@0.5 or Accuracy**	**FPS rate with GPU**	**FPS rate with CPU**
Classical CV	-	-	-	-	0.63	-	30
YOLOv3	0.25	0.98	0.98	0.98	0.994	33	-
YOLOv3 -Tiny	0.25	0.82	0.82	0.82	0.92	62	7
YOLOS	0.2	0.59	0.77	0.6	0.67	5	-
ViT*	0.2	0.84	0.81	0.83	0.91	-	-

* Not an object detection network, just tested for classification accuracy, thus does not have an FPS rate.

closed manhole images, 20 of them could be classified as high importance. Out of the 30 images the accuracy percentage of the method is as follows in Table 4.

Figs. 14, 15 and 16 depict the various predictions of the open manhole. If even one of the predictions was $>50\%$ of the area, then it's a high importance manhole that needs to be fixed.

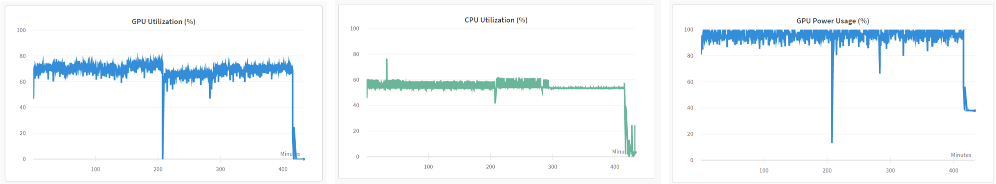


Fig. 10. GPU, CPU utilization and power consumption of YOLOS model.

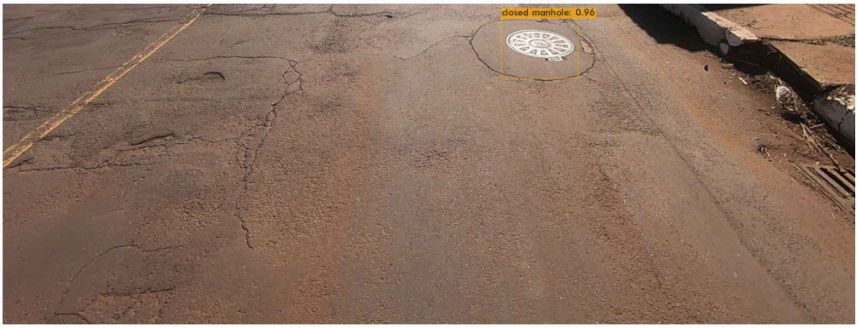


Fig. 11. Test result of YOLOv3.

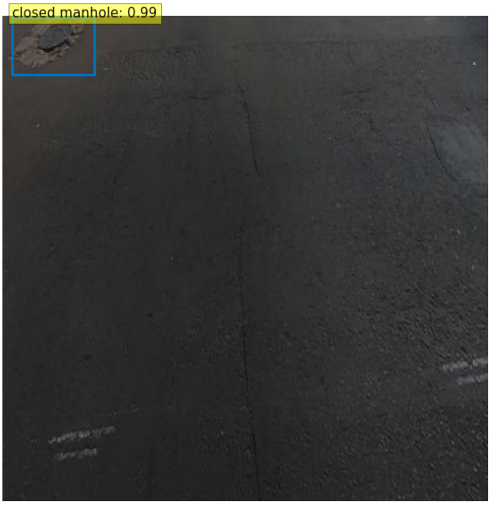


Fig. 12. Test result of YOLOS.

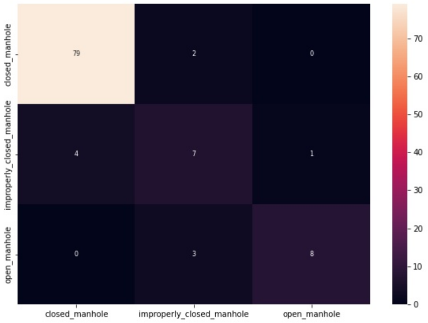


Fig. 13. Confusion matrix for ViT model.

Tab. 4. Results of size detection algorithm tested.

Class	Size Detection Algorithm Accuracy	FPS Rate tested with CPU
Open manholes	100%	30 fps
Closed manholes	100%	30 fps
Improperly closed manholes	90% *	30 fps

* There were 20 high importance manhole images from the 26 images of the improperly closed manhole class. 22/26 were classified as high importance.

5. Conclusion and Future Work

Including manhole detection models in obstacle avoidance and lane changing algorithms could improve the suitability of autonomous vehicles for use on the roads of developing countries. This research presented three different ways that autonomous vehicles may use for detecting manholes. We tested a classical computer vision approach involving image processing algorithms like Canny edge detection and ellipse detection. We also tested YOLOv3 and YOLOv3-Tiny to try Convolutional Neural Networks, and tested YOLOS and ViT to attempt vision transformer based approaches. The results of our research point to the conclusion that YOLOv3-Tiny would be the most suitable model for deploying on autonomous vehicles. Since self-driving vehicles have a drive-by-wire

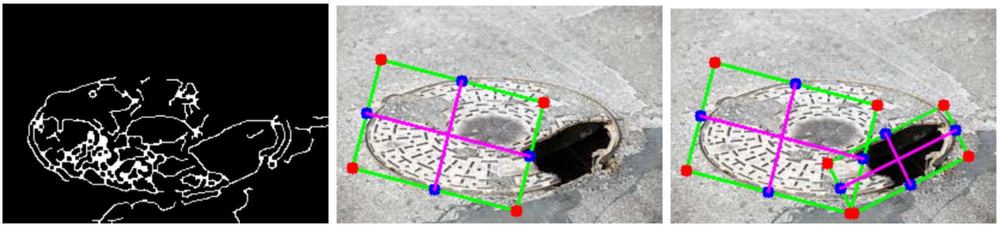


Fig. 14. Sample of a broken manhole.

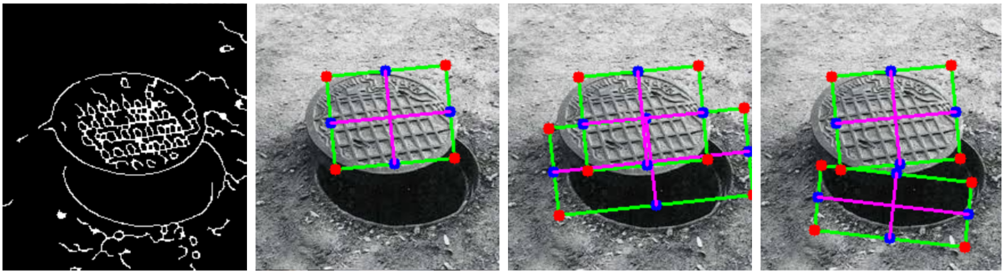


Fig. 15. Sample of a partially open manhole.

system and are equipped with cameras for computer vision, then the speed, computational complexity, and accuracy are the metrics that are taken into consideration when evaluating the three approaches to manhole detection. The GPU power consumption is low and FPS (frames per second) rate is very high for YOLOv3-Tiny, further reinforcing our conclusion. Vision Transformers also seem promising for this purpose with future research advancements.

In addition to manhole detection, our research presents a pipeline for size detection using filters and a contour detection method. The size of the hole in the broken/improperly closed manhole with respect to the manhole itself is determined and

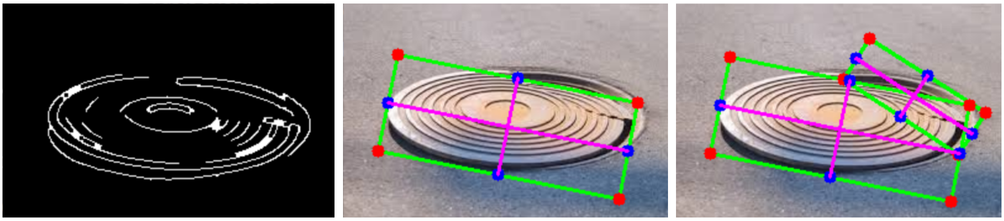


Fig. 16. Sample of a broken manhole that looks closed.

classified as high or low importance to determine a priority order for fixing them. Technology is designed to meet the requirements of different locations and the aim of science and development is to improve the standard of living in the places where they are used. This size detection model provides a method to not only detect manholes to avoid pavement distress and improve the standard of driving, but also enables the vehicle to aid in solving the problem of fixing an open, broken, or improperly closed manhole. With this in mind, we propose a future course of action. Autonomous vehicles are equipped with GPS and this system can be used to record the coordinates of the open or broken manhole along with its priority order from the size detection model. This information can then be used to alert the authorities for damage control.

In the future, we intend to work on aspects of research on autonomous vehicles that bring us one step closer to making self-driving vehicles a reality even in developing countries.

Acknowledgement

The authors would like to acknowledge BITS Pilani Hyderabad Campus for providing us an opportunity to carry out this project as part of Remote Sensing and Image Processing, under the guidance of Dr. K. Rajitha of the Civil Engineering Department.

Data accessibility statement

The data used in our work is accessible as open source at [33]. In our study many images from the dataset [38] described in [39] were used.

References

[1] P. Adarsh, P. Rathi, and M. Kumar. YOLO v3-Tiny: object detection and recognition using one stage improved model. In *6th Int. Conf. Advanced Computing and Communication Systems (ICACCS 2020)*, pages 687–694, Coimbatore, India, 6-7 Mar 2020. doi:10.1109/ICACCS48705.2020.9074315.

[2] A. Bochkovskiy. darknet. GitHub, 30 Oct 2021. `https://github.com/AlexeyAB/darknet`. [Accessed 17 Oct, 2022.

[3] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao. YOLOv4: Optimal speed and accuracy of object detection, 2020. arXiv:2004.10934. doi:10.48550/arXiv.2004.10934.

[4] D. Boller, M. M. Vitry, J. D. Wegner, and J. P. Leitão. Automated localization of urban drainage infrastructure from public-access street-level images. *Urban Water Journal*, 16(7):480–493, 2019. doi:10.1080/1573062X.2019.1687743.

[5] Can self-driving cars run on Indian roads? *moneycontrol*. [Accessed 17 Oct, 2022]. `https://www.moneycontrol.com/news/driverless-cars/`.

[6] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. doi:10.1109/TPAMI.1986.4767851.

[7] N. Carion, F. Massa, G. Synnaeve, et al. End-to-end object detection with transformers. In *Proc. European Conf. Computer Vision (ECCV 2020)*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229, Glasgow, UK, 23-28 Aug 2020. doi:10.1007/978-3-030-58452-8_13.

[8] Z. Chong and L. Yang. An algorithm for automatic recognition of manhole covers based on MMS images. In *Proc. 11th Chinese Conf. Advances in Image and Graphics Technologies (IGTA 2016)*, volume 634 of *Communications in Computer and Information Science*. Springer, Beijing, China, 8-9 Jul 2016. doi:10.1007/978-981-10-2260-9_4.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020. arXiv:2010.11929v2. doi:10.48550/arXiv.2010.11929.

[10] Y. Du, N. Pan, Z. Xu, et al. Pavement distress detection and classification based on YOLO network. *International Journal of Pavement Engineering*, 22(13):1659–1672, 2021. doi:10.1080/10298436.2020.1714047.

[11] L. Eliot. Manhole covers and self-driving cars. In: Self-Driving Cars. Podcasts by Dr. Lance Eliot, 9 Sep 2021. `https://ai-selfdriving-cars.libsyn.com/manhole-covers-and-self-driving-cars`. [Accessed 17 Oct, 2022].

[12] Y. Fang, B. Liao, X. Wang, et al. You Only Look at One Sequence: Rethinking transformer in vision through object detection. In *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, pages 26183–26197. 2021. `https://proceedings.neurips.cc/paper/2021/hash/dc912a253d1e9ba40e2c597ed2376640-Abstract.html`.

[13] L. Fei-Fei, J. Deng, O. Russakovsky, A. Berg, and K. Li, editors. *IMAGENET*. 2021. [Accessed December 2022]. `https://image-net.org`.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010-09. doi:10.1109/TPAMI.2009.167.

[15] R. Ghosh and O. Smadi. Automated detection and classification of pavement distresses using 3D pavement surface images and deep learning. *Transportation Research Record*, 2675(9):1359–1374, 2021. doi:10.1177/03611981211007481.

[16] E. Hildreth and D. Marr. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological sciences*, 207(1167):187–218, 1980. doi:10.1098/rspb.1980.0020.

[17] S. Ji, Y. Shi, and Z. Shi. Manhole cover detection using vehicle-based multi-sensor data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIX-B3:281–284, 2012. doi:10.5194/isprsarchives-XXXIX-B3-281-2012.

[18] G. Jocher, A. Chaurasia, A. Stoken, et al. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Zenodo, Nov 2022. doi:10.5281/zenodo.7347926.

[19] G. Jocher et al. YOLOv5 by Ultralytics. GitHub, 2020. `https://github.com/ultralytics/yolov5`. [Accessed 17 Oct, 2022].

[20] E. Karimi, M. Rezanejad, B. Fiset, et al. Machine learning meets classical computer vision for accurate cell identification., 28 Feb 2022. doi:10.1101/2022.02.27.482183.

[21] Y. M. Kim, Y. G. Kim, S. Y. Son, et al. Review of recent automated pothole-detection methods. *Applied Sciences*, 12(11):5320, 2022. doi:10.3390/app12115320.

[22] C. Kumar. Preventable deaths: In India, at least 2 die each day due to open pits & manholes. *The Times of India*, 25 Nov 2021. [Accessed 9 Oct, 2022]. `https://timesofindia.indiatimes.com/`

india/preventable-deaths-in-india-at-least-2-die-each-day-due-to-open-pits-manholes/articleshow/87917848.cms.

[23] Label Studio community (originally created by Tzutalin). LabelImg. GitHub, 23 Sep 2022. https://github.com/heartexlabs/labelImg. [Accessed 17 Oct, 2022].

[24] Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. In S. Avidan et al., editors, *Proc. European Conf. Computer Vision (ECCV 2022)*, volume 13669 of *Lecture Notes in Computer Science*, pages 280–296, Tel Aviv, Israel, 23-27 Oct 2022. doi:10.1007/978-3-031-20077-9_17.

[25] S. Liu, L. Qi, H. Qin, et al. Path Aggregation Network for instance segmentation. *arXiv*, 2018. arXiv:1803.01534v4. doi:10.48550/arXiv.1803.01534.

[26] S. Maji and J. Malik. Object detection using a max-margin Hough transform. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1038–1045, Miami, FL, USA, 20-25 Jun 2009. doi:10.1109/CVPR.2009.5206693.

[27] B. Mali, A. Shrestha, A. Chapagain, et al. Challenges in the penetration of electric vehicles in developing countries with a focus on Nepal. *Renewable Energy Focus*, 40:1–12, 2022. doi:10.1016/j.ref.2021.11.003.

[28] A. Mohan and S. Poobal. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 57(2):787–798, 2018. doi:10.1016/j.aej.2017.01.020.

[29] U. Nepal and H. Eslamiat. Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors*, 22(2):464, 2022. doi:10.3390/s22020464.

[30] H. Niigaki, J. Shimamura, and M. Morimoto. Circular object detection based on separability and uniformity of feature distributions using Bhattacharyya Coefficient. In *Proc. 21st Int. Conf. Pattern Recognition (ICPR2012)*, pages 2009–2012, Tsukuba, Japan, 11-15 Nov 2012. https://ieeexplore.ieee.org/abstract/document/6460553.

[31] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002-07. doi:10.1109/TPAMI.2002.1017623.

[32] J. Pasquet, T. Desert, O. Bartoli, et al. Detection of manhole covers in high-resolution aerial images of urban areas by combining two methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(5):1802–1807, 2016. doi:10.1109/JSTARS.2015.2504401.

[33] S. Rao and N. Mitnala. Manhole_Detection. GitHub, Dec 2022. https://github.com/sh-r/Manhole_Detection. [Accessed: Dec, 2022].

[34] S. Rao, A. Quezada, S. Rodriguez, et al. Developing, analyzing, and evaluating vehicular lane keeping algorithms using electric vehicles. *Vehicles*, 4(4):1012–1041, 2022. doi:10.3390/vehicles4040055.

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, real-time object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2016)*, pages 779–788, Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.91.

[36] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2017)*, pages 7263–7271, Honolulu, HI, USA, 21-26 Jul 2017. doi:10.1109/CVPR.2017.690.

[37] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv*, 2018. arXiv:1804.02767v1. doi:10.48550/arXiv.1804.02767.

[38] A. Santos, J. Marcato Junior, J. Andrade Silva, et al. Storm-drain and manhole detection. Geomatics and Computer Vision Datasets. https://sites.google.com/view/geomatics-and-computer-vision/home/datasets#h.sen6zve8r3ra. [Accessed Jan, 2022].

[39] A. Santos, J. Marcato Junior, J. Andrade Silva, et al. Storm-drain and manhole detection using the RetinaNet method. *Sensors*, 20(16):4450, 2020. doi:10.3390/s20164450.

[40] P. Saxena. Increase Frame Per Second (FPS) rate in the custom object detection step by step. Towards Data Science, 3 Sep 2020. `https://towardsdatascience.com/no-gpu-for-your-production-server-a20616bb04bd`. [Accessed 17 Oct, 2022].

[41] M. Simonovsky. Ellipse detection using 1D Hough transform. In *MATLAB Central File Exchange*. 2022. [Accessed 16 Oct, 2022]. `https://www.mathworks.com/matlabcentral/fileexchange/33970-ellipse-detection-using-1d-hough-transform`.

[42] N. Tanaka and M. Mouri. A detection method of cracks and structural objects of the road surface image. In *Proc. IAPR Workshop on Machine Vision Applications*, pages 387–390, Tokyo, Japan, 28-30 Nov 2000. `http://b2.cvl.iis.u-tokyo.ac.jp/mva/proceedings/CommemorativeDVD/2000/papers/2000387.pdf`.

[43] R. Timofte and L. Gool. Multi-view manhole detection, recognition, and 3D localisation. In *Proc. IEEE Int. Conf. Computer Vision Workshops (ICCV Workshops)*, pages 188–195, Barcelona, Spain, 16 Jan 2011. Workshops. doi:10.1109/ICCVW.2011.6130242.

[44] H. Touvron, M. Cord, M. Douze, et al. Training data-efficient image transformers & distillation through attention. In M. Meila and T. Zhang, editors, *Proc. 38th Int. Conf. Machine Learning (ICML 2021)*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357, Virtual Only, 18-24 Jul 2021. PMLR. `https://proceedings.mlr.press/v139/touvron21a.html`.

[45] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, volume 30. 2017. `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

[46] 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. *arXiv*, 2022. arXiv:2207.02696v1. doi:10.48550/arXiv.2207.02696.

[47] C.-Y. Wang, H.-Y. M Liao, Y.-H. Wu, et al. CSPNet: A new backbone that can enhance learning capability of CNN. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2020)*, pages 1571–1580, Seattle, WA, USA, 14-19 Jun 2020. doi:10.1109/CVPRW50498.2020.00203.

[48] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, et al. CSPNet: A new backbone that can enhance learning capability of CNN. *arXiv*, 2019. arXiv:1911.11929v1. doi:10.48550/arXiv.1911.11929.

[49] Y. Xie and Q. Ji. A new efficient ellipse detection method. In *Proc. 2002 Int. Conf. Pattern Recognition (ICPR 2002)*, volume 2, pages 957–960, Quebec City, QC, Canada, 11-15 Aug 2002. doi:10.1109/ICPR.2002.1048464.

[50] S. Yan. *Manhole Cover Detection from Natural Images.* PhD thesis, University of Dublin, Trinity College, Sep 2020. [Accessed 17 Oct, 2022]. `https://www.scss.tcd.ie/publications/theses/diss/2020/TCD-SCSS-DISSERTATION-2020-111.pdf`.

[51] Z. Yang, Y. Liu, L. Liu, X. Tang, J. Xie, and X. Gao. Detecting small objects in urban settings using SlimNet model. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):8445–8457, 2019. doi:10.1109/TGRS.2019.2921111.

[52] Y. Yu, H. Guan, and Z. Ji. Automated detection of urban road manhole covers using mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3258–3269, 2015. doi:10.1109/TITS.2015.2413812.