Detection of manholes

Shika Rao
Electrical and Electronics Engineering Department
Birla Institute of Technology and Science, Pilani
-Hyderabad Campus
Hyderabad, Telangana, India
f20191237@hyderabad.bits-pilani.ac.in

Nitya Mitnala
Electrical and Electronics Engineering Department
Birla Institute of Technology and Science, Pilani
- Hyderabad Campus
Hyderabad, Telangana, India
f20191216@hyderabad.bits-pilani.ac.in

***Abstract*—Manhole construction is an integral part of road development in the country. However, 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.**

Present methods of manhole detection are highly unreliable, especially in times of heavy rains and floods, wherein low visibility is a problem. The implementation of this project could lead to avoidance of the path with the open manhole, thus aiding in times of adversity and helping the authorities keep the roads in a clean and safe condition.

Our aim in this project is to identify manholes in a road, check if the manhole is correctly closed or not, and if there is a lid, whether or not it is at road level. We are attempting to achieve these aims using principles of Remote Sensing and Image Processing.

***Index Terms*—Manhole detection, image processing, MATLAB, Convolutional Neural Networks, YOLO object detection, Deep Learning**

I. Introduction

Urban growth is an ongoing trend, and one of its direct consequences is the development of underground utility networks. Manholes are vital components of the urban-drainage system that are essential to the hydrological modeling of urban basins. Accurate mapping of these objects can help improve the drain systems to prevent and mitigate urban floods. Furthermore, these manholes can be used as landmarks in photogrammetric applications [29,1997] or geotechnical works such as subsidence calculations [30, 2011]. [3,2015]

We put forward a methodology to detect small urban objects, namely manholes. Two methods are tested on the dataset. The first method is based on Canny edge detection and geometrical circular filter [3,2016], whereas the second one implements a Convolutional Neural Network (CNN) based on the YOLO algorithm. The results are then compared in order to benefit from the two approaches.

II. 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. Spatial position and attribute information of a manhole cover is required for digital city management. Though traditionally, manhole cover measurement is done through manual surveys, it is seen that this method is both laborious and time-consuming. For this reason, we have a requirement for methods of survey that are unmanned. However, a more efficient system of survey is object detection in remotely-sensed images. The categories of methods used are object-based image analysis, fusion of spatial information and machine learning, and deep learning approaches.

A. Road-surface images

A morphological method (dependent on the structure) was developed in [9, 2000] for detecting round-shaped manhole covers. It involved a 'black top-hat transform' operation which is conventionally used for removing uneven background illumination from disc-shaped structuring elements. This filtering was done for feature extraction and histogram equalization from road surface images. A masking operation was then done on the extracted round components.

Improved Hough transform was proposed in [13, 2006] for the detection of manhole covers. Contour tracking was performed on binary images, and false positives were eliminated using contour filtering.

Obscure and textured manhole covers were challenges faced by conventional methods of object detection. In [10, 2009], this drawback was overcome using separability filters. A method for circular object detection in noisy images with inhomogeneous contrast was developed. Instead of analyzing the difference in intensity levels of the object interior and surroundings, the separability and uniformity of intensity distributions were analyzed using the Bhattacharyya coefficient.

In [12, 2011], a multi-view method implementing 2-D and 3-D techniques 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 segmentation area, aspect ratio, variance, symmetry, and texture-based filters to the 2-D views. Multi-view processing involved grouping the results of single-view processing, and these were used in a graph-cut segmentation filter. These 3-D hypotheses were used for accurate localization of the manholes.

In [16, 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. Scene segmentation to eliminate cars and pedestrians was done, and Canny edge detection [4, 1986], which is sensitive to arcs and ellipses, was applied.

B. Deep Learning approach used on road surface images

Automated detection of manholes using Mobile Laser Scanning (MLS) data was put forward in [14, 2014]. The superiority of MLS-based systems over optical imaging Mobile Mapping Systems (MMS) was presented. Road surface points were segmented from a point cloud via a curb-based approach. Then, these were rasterized into a georeferenced intensity image through Inverse Distance Weighted (IDW) interpolation. A supervised deep learning model was trained to depict the high-order features in a multilayer feature generation model. A random forest model was then trained to map the high-order features to the probability of existence of manhole covers at specific locations. The georeferenced intensity images based on the multilayer feature generation model and random forest model thus enabled detection of urban road manholes. Circular-shaped and rectangular-shaped road manhole covers were detected and optimized based on the reversible jump Markov chain Monte Carlo (RJMCMC) algorithm.

In [26, 2019], deep learning algorithms were integrated with smartphones to detect potholes in real-time. A deep learning object detection algorithm called Single Shot Multibox Detector (SSD) was used for detecting potholes using a mobile camera in the background. SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for each object category's presence in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to handle objects of various sizes. [21, 2016] A Deep Feed Forward Neural Network model using SSD was trained in [26, 2019] and used to assess continuously taken accelerometer and gyroscope readings, to detect unregistered potholes. A method was then developed to directly update the coordinates of these unregistered potholes to a database in real-time.

Mapping manholes using the deep learning method RetinaNet in road-level RGB images is done in [22, 2020]. ResNet-50 and ResNet-101 were adopted in the experimental assessment as the two distinct feature extractor networks for the RetinaNet method. The results were then compared with the Faster R-CNN method. R-CNN (Regions with CNN features) series algorithms are two-staged detection algorithms based on region proposals. This process uses non-maximum suppression (NMS) to eliminate redundant proposals. After NMS screening, detected objects are obtained. R-CNN combines region proposals with CNNs and two key insights can be derived from it namely, high-capacity CNN can be applied to bottom-up region proposals in order to localize and segment objects and, when labeled training data is scarce, supervised pre-training for an auxiliary task followed by domain-specific fine-tuning, yields a significant performance boost. [17, 2014] Fast R-CNN efficiently classifies object proposals using deep convolutional networks and employs several innovations to improve training and testing speed while also increasing detection accuracy. [18, 2015] A Region Proposal Network(RPN) is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. RPN and Fast R-CNN are merged into a single network by sharing their convolutional features. The RPN component tells the unified network where to look, thus creating a new, more efficient network- Faster R-CNN.[19, 2017] However, the findings of this paper [22, 2020] concluded that the RetinaNet method was more effective than the Faster R-CNN method.

A method was proposed in [8, 2020] to convert RGB image data and contour extraction using an object detection algorithm. Pavement distress is classified using the YOLOv3 (You Only Look Once) algorithm which is a one-stage detection algorithm that does not need the region proposal phase. It is framed as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation, and it can be optimized end-to-end directly on detection performance. [20, 2016] In [8, 2020], a large-scale dataset was prepared, the containing images taken in various weather and illumination conditions. The image data was analyzed effectively by using AP as the indicator on the data.

C. Aerial images

A novel method for photogrammetric extraction of information from aerial images was done in [11, 1997]. This approach was based on a parametric intensity model. Orientation parameters were determined using spatial resection, and the reliability and accuracy depended on the selection as well as precision obtained for landmark localization. Manhole covers were used as a specific type of circular landmark, and localization was done by fitting the

model to the observed image intensities.

D. Deep Learning and Machine Learning Approach used on aerial images

In [3, 2016], a framework was proposed to automatically detect manhole covers in high-resolution aerial images by combining the method based on the geometrical filter with a machine-learning SVM-based approach. Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. In [25, 2017], it was proposed that the combination of the circular filter with deep-learning CNN instead of SVM could be envisaged to obtain better results than only using RGB channels.

Localization of manhole covers and automatic detection in Very High Resolution (VHR) aerial and remotely sensed images using Convolutional Neural Networks (CNN) was done in [23, 2017]. The entire image is processed without any prior segmentation due to the deep neural networks approach adopted. A customized AlexNet architecture [28, 2012] was trained and implemented on very high spatial resolution aerial images.

In [2, 2019], the challenge that manholes appear in different scales in remotely sensed images was tackled. DCNN (Deep Convolutional Neural Networks) have a fixed receptive field and cannot match the scale variability of these objects. DCNN has poor localization performance when applied to densely packed objects, which is how manhole covers are portrayed in remotely sensed images. To overcome this difficulty, Visual Group Geometry (VGG), which can increase the receptive field size, was adopted. A multi-scale network (MON) and a Multi-Level Convolutional Matching Network (M-CMN) were used to match receptive fields with object scales and fuse feature maps for a more robust response, respectively.

III. Methodology

A. Functional Block Diagram

Fig. 1 describes the overall flow of the proposed methodology. There are two approaches adopted while doing this study. The first method is the feature extraction method which involves pre-processing and then classification of the images using algorithms. Pre-processing consists of preparing the field for taking input of images and filtration of noise and unwanted detection. Manhole detection involves the use of geometrical methods for elliptical object detection. The second method uses deep learning, wherein feature extraction and classification are performed jointly (end-to-end learning). It involves the use of YOLOv3 object detection algorithm on the Darknet backbone. The flowchart gives a visual understanding of some of the key decision points made in the methodology. This modular approach to the methodology allows for ease in planning future changes in functionality and flow to increase efficiency and integration with other supplementary methods.

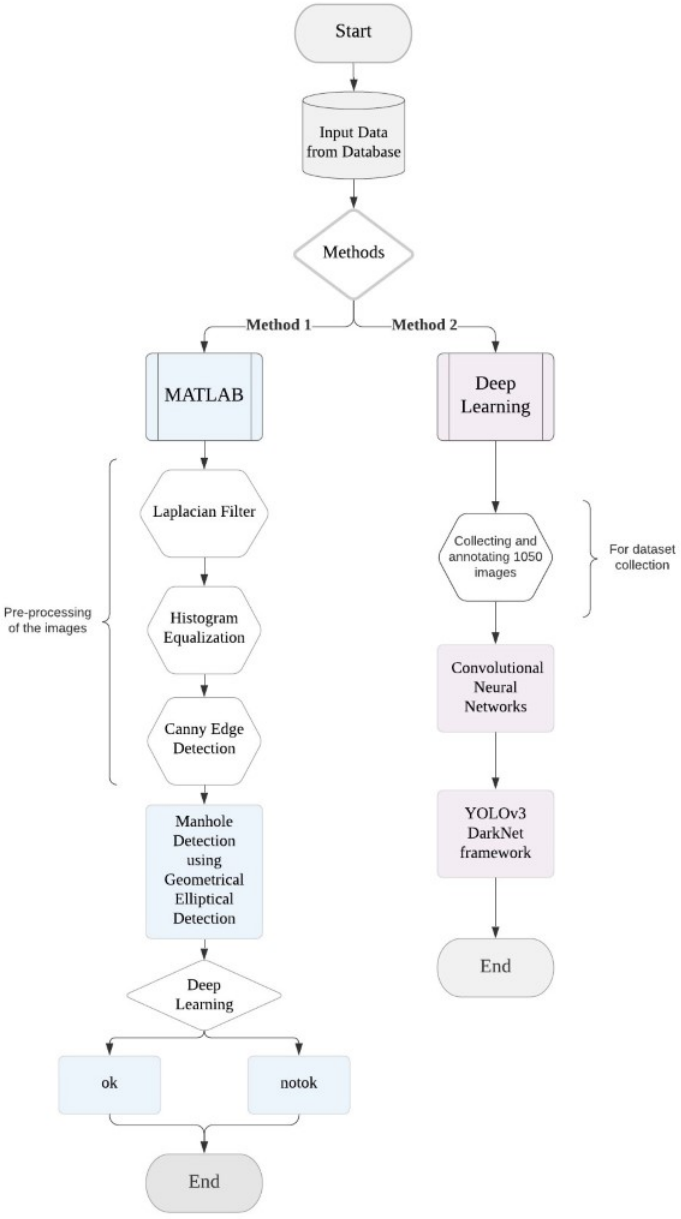

Fig. 1. Flowchart of the proposed methodology

B. Dataset

We used road surface images for this study. The ease of availability of these images and lack of requirement for specific equipment for obtaining the images was a contributing factor in the decision to use road surface images. In [35, 2019], it was demonstrated that street level imagery could provide useful information to identify manholes that could not be appropriately detected in aerial images. The dataset publicly provided for future investigations in [N, 2020] has

been used for our project.

https://sites.google.com/view/
geomatics-and-computer-vision/home/datasets

In the dataset we collected, 157 images were taken from the above pre-existing dataset and the rest of the images were acquired through Google Street View, Google Images, and Bing Images. A total of 1050 images were collected and annotated.

IV. Method 1

A. Methodology

a) Data Collection and Classification: A dataset containing 40 images was collected from Google images. Initial classification was done manually, as the dataset was small. The images were simply classified into different folders, and the paths to the folders were taken while writing the program. The data was divided into two classes:

1) ok (containing images with perfectly closed manholes)
2) notok (containing images of manholes with broken/partially closed/ open lids)

b) Pre-processing: The images are processed to obtain a high degree of accuracy when manhole detection is performed. A combination of several techniques has been used to achieve this goal. For filtering, spatial domain filtering, namely the Laplacian filter, was preferred. This edge detector filter was used to compute the second derivatives of the images. [5, 1980] This step was done to determine whether changes in adjacent pixel values were from an edge or continuous progression. The Laplacian filter kernels contain negative values in a cross pattern centered within the array, and the corners are either zero or positive values. This filter was preferred because this filter reduced the noise associated with the images while preserving the edges.

The second step is the use of equalized histogram method. The histogram modification is an image enhancement technique that adjusts the dynamic range of pixel values. Visual information can be delivered effectively to the human viewer, and the non-overlapped block adaptive technique is based on the pixel properties intensities in a neighborhood. [6, 1998] A histogram is a graphical representation of the total number of pixels varying with intensity values. [7, 1992] For a non-equalized image, there are some intensity values around which maximum pixels are clustered.[31, 2020] This step is done to improve the overall contrast of the image as low contrast regions are brought closer to the average contrast value.

Canny Edge Detection [4, 1986] was done to detect the edges of objects in the images. If the gradient magnitude of a pixel is larger compared to pixels on either side of it in the direction of maximum intensity change, the Canny edge detector classifies that pixel as an edge. This technique is performed to extract the morphological information from the images and reduce the amount of data to be processed before performing circular detection to identify manholes.

Fig. 2 shows an example of the original image taken before pr-processing.

Fig. 2.

Fig. 3 show an example of how an image looks after pre-processing.

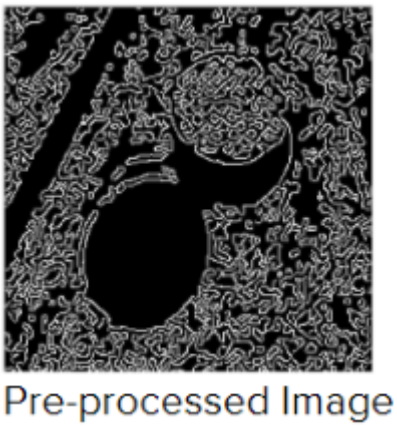

Fig. 3.

c) Manhole Detection: The geometrical circular detection filter is based on the approach proposed in [10, 2009] and adapted in [3, 2015]. However, since the manholes in our road surface images have a mostly elliptical shape, we used an elliptical filter. This filter is applied to the binary preprocessed images.

The shapes that were the best fit as per the definition of an ellipse were taken. In the case of more than one best fits,

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.

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 was converted to binary form, and the logical 'and' of the binary image and the mask were taken, 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. We assumed that this was because the manhole shape looked very circular or because there was no other disturbance surrounding the manhole. To make sure that this did not stop the model from continuing, we wrote a conditional to allow the original pre-processed image itself to be taken as the final product in such a case.

Fig.4 shows an example of the ellipse drawn around the manhole.

Fig. 4.

As seen in Fig. 4, the ellipse is drawn approximately around the manhole. The whole process of preprocessing and manhole detection was set into a loop, taking the images earlier classified into ok and notok as inputs. The final images were sorted into another two files: processed ok and processed notok, depending upon which source they had come from.

d) Improperly Closed/ Open Manhole Detection: Deep Learning: Training and Validation
The 40 images in the processed ok and processed notok files were taken as the input for the deep-learning code. Training options were set. MATLAB has set only one iteration per epoch. The number of iterations per epoch can not be changed, but the number of epochs can be changed. The number of epochs was set to 64. The more the number of epochs, the longer it takes to run. Since we ran the code many times, it was not suitable for us to use larger number of iterations. The number of images used for training was 34. The remaining 6 images were used for validation.

B. Results of Method 1

a) Manhole Detection: The manholes were successfully detected in the images. A picture of an original image and the final image after detection was done is shown in Fig. 5. As seen in the final image, a bounding box was drawn around the manhole.

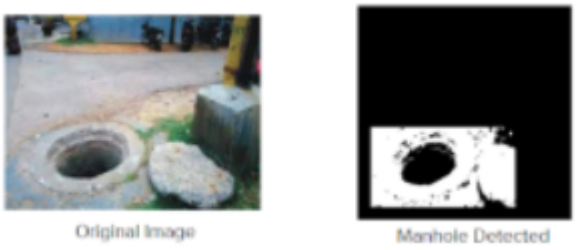

Fig. 5.

b) Result for Deep-Learning: An image of the graph at the end of 64 epochs is shown in Fig. 6. According to the graph, Validation Accuracy= 83.33% .The model was trained as the accuracy percentage increased and the loss percentage decreased. Training accuracy finally reached 100%.

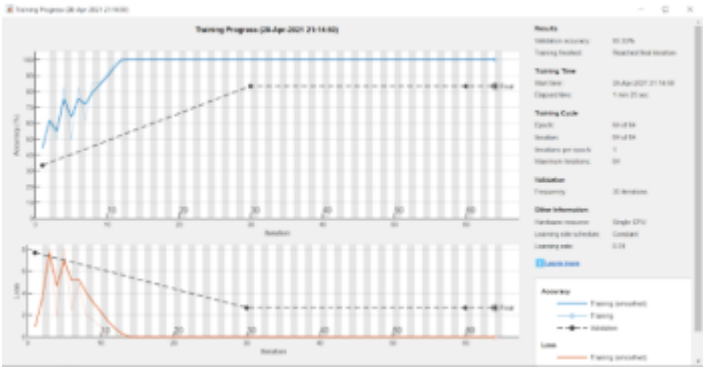

Fig. 6.

c) Testing: Though it is recommended that images not used for training and validation should be used for testing, as we are not using such a large dataset, the images from the notok folder were used to test the model. The model classified them perfectly, labeling them all as notok.

C. Discussion of Method 1

a) Scope for Improvement and Learning Outcomes: As it is clear from the results, this model has a fairly good accuracy for one that has such a small dataset and only 64 iterations. If the number of iterations are increased and

the number of images also are increased, we think that the accuracy can be increased.

We haven't increased the number of iterations in MATLAB due to the fact that the run time would greatly increase. Since this code was run on a normal CPU on a PC, there is not so much computational power and running it would take too long. We opted not to do it due to time constraints. There are a low number of images because the program we used requires us to input the whole list of images' file names manually. If we used a method to read all the file names in the directory automatically, we could increase the number of images used, thus improving performance. But, using such a method would only mean that we would have to take a lot more time to run the pre-processing and deep learning. Hence, we did not do it.

However, this method has shown the basic techniques of image processing that can be used in widely used Deep Learning networks. In order to see how well a Deep Learning network which has incorporated Image processing techniques works when presented with the same problem, we give the second method in our project.

b) Drawbacks: One issue may have been with the ellipse detection. Though we have used the mean of the best fits, perhaps using some other method of arriving at a final ellipse, maybe max or min of best fits could have created a more accurate ellipse.

Another drawback is that we can not detect manholes which are not seen as ellipses in the image. If the manhole was in the form of a square or rectangle, or if the image is taken from directly overhead, making it appear as a circle, then the model will fail.

The final drawback comes with our need to fall back to the original image in case of not being able to detect an ellipse. If we could find a better way for ellipse detection, this problem might be covered, increasing our accuracy.

V. Method 2

A. Methodology

Deep Learning-based methods have shown higher performance in tasks because they can extract features while jointly performing classification. Deep Learning methods have also been very successful in object detection in several applications, such as agriculture, urban infrastructure, and health analysis. However, there have been very few research works on manhole detection using Deep Learning.

a) Image Dataset: The images collected were manually annotated by marking the manhole objects with rectangles (bounding boxes) and labeling each rectangle to its corresponding class. This was done using the 'labelImg' tool. We divided the images into the following classes:

1) Open manhole
2) Closed manholes
3) Improperly closed manhole

Afterward, these images were divided into two groups, namely training and testing sets. The same set of images were used for training as well as validation as specified in [34]. There are 1033 images in the training and validation set and 17 images in the testing set. These sets were considered to assess the methods contributing to a more robust evaluation.

Fig. 7 shows a sample of an annotated image.

Fig. 7. Sample of annotated image

b) Object Detection Method: A Convolutional Neural Network is able to successfully capture the spatial and temporal dependencies in an image through the application of relevant filters. The objective of the convolution operation is to extract high-level features such as edges from the input image.

For this study, the YOLOv3 object detection method [20, 2016], [36, 2018] was adopted. YOLOv3 (You Only Look Once version 3) is an object detection algorithm for converting RGB image data. It is a one-stage detection algorithm, and it is framed as a regression problem. Instead of selecting only particular parts of an image, it predicts classes and bounding boxes for the whole image.

YOLO divides the input image into grid cells wherein each grid cell predicts only one object. Now each grid cell predicts multiple number of boundary boxes and assigns a box confidence score. This confidence score reflects how likely the box contains an object and the accuracy of the bounding box. It also predicts conditional class probabilities that is, the likeliness of an object belonging to a particular class.

To compute the loss for the true positive, only one of the bounding boxes out of the multiple bounding boxes predicted is to be responsible for the object. For this purpose, the one with the highest IoU (intersection over union) with the

bounding box we had provided when annotating the image is selected.

Region proposal methods limit the classifier to the specific region. YOLO accesses the whole image in predicting boundaries. With the additional context, YOLO demonstrates fewer false positives in background areas.YOLOv3 uses multi-label classification. YOLOv3 replaces the softmax function with independent logistic classifiers to calculate the likeliness of the input belonging to a specific label. [36, 2018]

$$\text{box confidence score} \equiv P_r(object) \cdot IoU$$
$$\text{conditional class probability} \equiv P_r(class_i|object)$$
$$\text{class confidence score} \equiv P_r(class_i) \cdot IoU$$
$$= \text{box confidence score} \times \text{conditional class probability}$$

where

$P_r(object)$ is the probability the box contains an object.
IoU is the IoU (intersection over union) between the predicted box and the ground truth.
$P_r(class_i|object)$ is the probability the object belongs to $class_i$ given an object is presence.
$P_r(class_i)$ is the probability the object belongs to $class_i$

Fig. 8. Mathematics behind YOLO object detection system

We adopted the DarkNet neural network framework as the backbone for YOLOv3. We adopted transfer learning, so the models' weights were initialized with weights from the same architecture pre-trained on the MS Coco dataset [36, 2018]. We adopted this approach in our project as transfer learning enables the training of deep neural networks with comparatively little data, accuracy is high, and the training time is reduced. We used the source code available in [34] 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 [34]). The classes, number of filters, steps, jitter, etc., were also customized to our requirements in the config file. Most of the instructions for training the model on a custom dataset were well documented in [34] as it's an open source framework and there are various troubleshooting resources easily available too.

c) Method Assessment: The performance of DarkNet YOLOv3 was assessed by precision–recall curves and the average precision (AP) as adopted in [36, 2018]. To estimate the precision and recall, the Intersection over Union (IoU) was calculated. This metric is given by overlapping the area between the predicted and the ground truth bounding boxes divided by the area of union between them. Following well-known competitions in the object detection scene, a correct detection (True Positive, TP) was considered for IoU 0.5, and a wrong detection (False Positive, FP) for IoU ¡ 0.5. A False Negative (FN) is assigned when no corresponding ground truth is detected. Based on the above metrics, precision (P) and recall (R) are estimated using below equations.

- True Positive- TP was considered for IoU >= 0.5
- False Positive- FP for IoU <0.5 (Indicates a wrong detection)
- False Negative- FN is assigned when no corresponding ground truth is detected.

- Precision (P)= $$\frac{TP}{TP + FP}$$
- Recall (R) = $$\frac{TP}{TP + FN}$$

The average precision (AP) is estimated by the area under the precision–recall curve.

B. Results of Method 2

a) Indicators used while training: The Mean Average Precision (mAP) was used as the primary indicator to analyse the accuracy of the model. We trained the model while the mAP percentage increased and the loss decreased. The mAP percentage value was calculated during training. The images as part of the validation set were used to calculate the mAP. The loss curves also indicate accuracy, however mAP is considered to be a better indicator.

b) Learning results of the object detection method: In Fig. 9, even though the mAP percentage reduces at some points on the graph , we still continued training as the recall values were still increasing. [20, 2016]

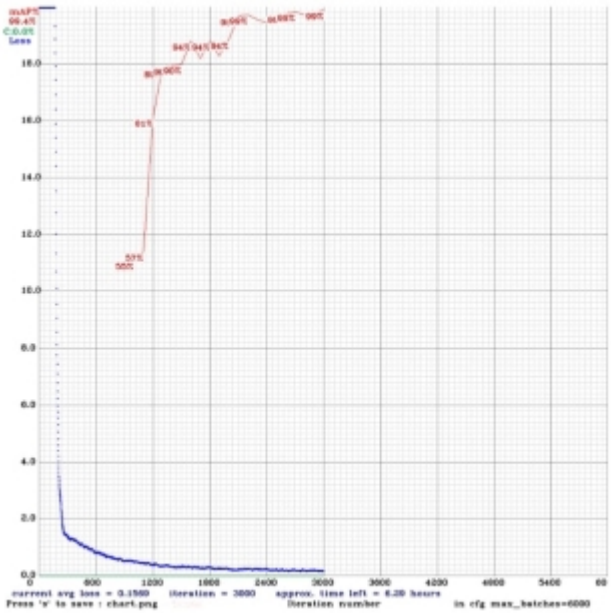

Fig. 9.

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 at 3000 iterations, i.e. half the maximum (6000 was set as the max-batch size as data was split into 3 classes). Thus the image data was analyzed effectively by reducing the loss rate on the data and increasing mAP.

The weights files were generated for every 1000 iterations thus there are 3 weights files. The best weights obtained were saved and can be used for further testing and manhole classification. As mAP calculation is every 100 iterations after 1000, and weights are saved every 1000 iterations, the weights which gave the best mAP percentage were saved separately in a different file for further use.

c) Inference results of the object detection method: The average precision (AP, %) and its mean values (mAP, %) obtained from the area under the curve are illustrated in the above figure and in the table Fig. 10. The results in the table display the IoU cutoff at 0.5 (AP50) and the AP values of each class.

Class ID	Class Name	TP	FP	AP
0	Open manhole	97	3	99.56%
1	Closed manhole	845	20	98.75%
2	Improperly closed Manhole	214	4	99.92%

Confidence Threshold	TP	FP	FN	Average IoU	Precision	Recall	F1 score	mAP@0.5
0.25	1156	27	29	78.99%	0.98	0.98	0.98	0.9941

Fig. 10. These values are for iteration 3000 which produced the best mAP value

Considering the information in Fig. 10, it becomes obvious that not all predictions were made correctly by DarkNet YOLOv3. This could be attributed to the fact that some images were noisy and the illumination conditions were dark, which thus made it challenging for the trained network to detect. Nevertheless, even in these conditions DarkNet YOLOv3 achieved an IoU value of 0.7899 and a mAP of 99.41%.

d) Testing: The model was tested on images which weren't a part of the training or validation set. The weights which contained the best mAP percentage were used for testing the images. Fig. 11 shows a sample of the image returned after testing by the model.

Fig. 11. Sample of result

C. Discussion of Method 2

The same images were used for training as well as validation. The mAP flag calculates the accuracy based on the validation dataset. Since the same images were used for training as well as validation, the accuracy turned out to be very high. This could mean that the model is slightly overfitting. However, early stopping was adopted to counteract this.

Deep Learning models perform better and give more accurate results when the dataset it is trained on is larger. So more images should have been collected and annotated. The images collected for the dataset (1033 images) had to be manually annotated, and the rectangular bounding box had to be drawn around each image manually. Since there are not many pre-existing datasets available with images of manholes, this was a labor-intensive task.

The program ran for 3035 iterations. It took 5 hours and 15 minutes for it to train, even with a GPU that supports fast computation. Thus, it's a very time-consuming process.

VI. Conclusion

A. Summary

The 2 methods are performed, and their results are evaluated.

The first method shows how the image pre-processing is done. Since this has been inculcated into the Deep learning networks, due to the ease of detection and minimal manual pre-processing of images involved, nowadays, the deep learning model is preferred. The imperfect ellipse detection in Method 1 is its major drawback since it reduces its accuracy. Also, method 1 uses a brand new deep learning network that was not previously trained on any other dataset. The model trained on YOLOv3 Darknet neural network framework covers these drawbacks as the bounding box is drawn manually and also because it uses transfer learning, it has a much better accuracy.

However, the number of images required in the deep learning approach for processing is very high, making the manual annotation very difficult. The computational power

required is also very high.

We have provided a basic approach for manhole detection and classification that has many applications for industries and the government.

B. Learning Outcomes and Applications

Two main methods of object detection were studied in detail. The principles thus discovered have applications in many different fields. In the Review of Literature, we have mentioned the ongoing research in the field of object detection in aerial images. Though our present project made use of only road surface images, real-time detection of manholes from aerial images is very important as well. Our study has shown yet another use of object detection, other uses being pothole detection, detection of faults in roads, uses in traffic control, detection of faults in buildings, etc. A similar project should be carried out using aerial images, collaborating with Google Maps to get satellite images to detect the location of faulty manholes in real-time. Reporting such cases to the concerned authorities would prevent many accidents and tragic loss of life.

Though some steps that need to be followed for manhole detection in aerial images are similar to our methods, many more steps must be performed as well. Further research can be done to implement this.

REFERENCES

[1] https://ieeexplore.ieee.org/document/7084661
[2] https://www.mdpi.com/2220-9964/8/1/49
[3] https://sci-hub.se/10.1109/JURSE.2015.7120524
[4] https://ieeexplore.ieee.org/document/4767851
[5] http://www.hms.harvard.edu/bss/neuro/bornlab/qmbc/beta/day4/marr-hildreth-edge-prsl1980.pdf
[6] https://sci-hub.do/https://ieeexplore.ieee.org/abstract/document/663733
[7] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley, 1992.
[8] Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., Kang, H. (2020). Pavement distress detection and classification based on YOLO network. International Journal of Pavement Engineering, 1-14
[9] N. Tanaka and M. Mouri, "A detection method of cracks and structural objects of the road surface image," in Proc. IAPR Workshop Mach. Vis. Appl., Tokyo, Japan, 2000, pp. 387–390.
[10] 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 Recog., Tsukuba, Japan, 2012, pp. 2009–2012.
[11] Drewniok and D. Rohr, "Model-based detection and localization of circular landmarks in aerial images," Int. J. Comput. Vis., vol. 24, no. 3, pp. 187–217, Sep. 1997.
[12] R. Timofte and L. van Gool, "Multi-view manhole detection, recognition, and 3D localization," in Proc. IEEE Int. Conf. Comput. Vis. Workshops, Barcelona, Spain, 2011, pp. 188–195
[13] Y. Cheng, Z. Xiong, and Y. H. Wang, "Improved classical Hough transform applied to the manhole cover's detection and location," Opt. Tech., vol. 32, no. S1, pp. 504–508, 2006.
[14] Y. Yu, J. Li, H. Guan, C. Wang, and J. Yu, "Automated detection of road manhole and sewer well covers from mobile LiDAR point clouds," IEEE Geosci. Remote Sens. Lett., vol. 11, no. 9, pp. 1549–1553, Sep. 2014
[15] Deng, Z.; Sun, H.; Zhou, S.; Zhao, J.; Lei, L.; Zou, H. Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 2018,145, 3–22
[16] S. Ji, Y. Shi, and Z. Shi, "Manhole cover detection using vehicle-based multi-sensor data," in Proc. Int. Archives Photogramm. Remote Sens. Spatial Inf. Sci., Melbourne, Australia, vol. 39-B3, 2012, pp. 281–284.
[17] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv 2014, arXiv:1311.2524v4.
[18] Girshick, R. Fast R-CNN. arXiv 2015, arXiv:1504.08083.
[19] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017,39, 1137–1149.
[20] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June2016; pp. 779–788.
[21] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
[22] https://sci-hub.do/https://www.mdpi.com/1424-8220/20/16/4450
[23] https://hal.archives-ouvertes.fr/hal-01556762/document
[24] https://www.mdpi.com/2076-3417/10/19/6662/htm
[25] https://hal.archives-ouvertes.fr/hal-01556762/document
[26] https://sci-hub.do/https://ieeexplore.ieee.org/document/9105737
[27] C. Drewniok and K. Rohr. Exterior orientation-an automatic approach based on fitting analytic landmark models. ISPRS Journal of Photogrammetry and Remote Sensing, 1997, 52: 132-145.
[28] Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. in Advances in neural information processing systems pp. 1097–1105
[29] C. Drewniok and K. Rohr. Exterior orientation-an automatic approach based on fitting analytic landmark models. ISPRS Journal of Photogrammetry and Remote Sensing, 1997, 52: 132-145
[30] G. Liu, H. Jia, R. Zhang, H. Zhang, H. Jia, B. Yu and M. Sang, "Exploration of subsidence estimation by persistent scatterer InSAR on time series of high resolution TerraSAR-X images." Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal, 2011, 4(1): 159-170.
[31] https://sci-hub.se/10.1109/ICSSIT48917.2020.9214219
[32] https://www.sciencedirect.com/science/article/pii/S1110016817300236
[33] https://ieeexplore.ieee.org/document/7065001
[34] https://github.com/AlexeyAB/darknet
[35] Boller, D.; de Vitry, M.M.; Wegner, J.D.; Leitão, J.P. Automated localization of urban drainage infrastructure from public-access street-level images. Urban Water J. 2019, 16, 480–493. [CrossRef]
[36] https://pjreddie.com/media/files/papers/YOLOv3.pdf
[37] https://in.mathworks.com/
[38] https://colab.research.google.com/drive
[39] https://vovaprivalov.medium.com/training-alekseyab-yolov3-on-own-dataset-in-google-colab-8f3de8105d86
[40] https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088