TABLE OF CONTENTS

Chapter 1

Introduction

Ensuring the integrity of pipelines is critical to preventing operational failures and safety hazards. Traditional inspection methods are often time-consuming and prone to errors. This project utilises the YOLOv5 (You Only Look Once) model to automate defect detection within pipes, specifically identifying multiple folds and the pipe's centre.

The project involves training the YOLOv5 model with annotated images of pipe interiors, where defects are marked with bounding boxes. After training, the model is tested on new, unseen videos to evaluate its performance. Using OpenCV (cv2), the model processes video frames, highlighting detected defects with bounding boxes.

The final output is a video that visually highlights the detected defects, providing an efficient and accurate method for pipeline inspection, thereby reducing time and resources required for manual inspections.

Chapter 2

Literature Survey

Sr No.	Title of the paper	Name of Authors	Description of Paper	Comparison Overview
1.	"Automatic Detection of Internal Corrosion Defects in Natural Gas Gathering Pipelines Using Improved YOLOv5 Model"	Bingjie Chen, Li Ma, Shan Liang	This paper presents an advanced YOLOv5 model designed to automatically detect internal corrosion defects in gas pipelines. It addresses the challenges of traditional pipeline endoscopy by using video footage from an endoscope robot. The paper details techniques like sample augmentation and defect segmentation to enhance data, and it introduces a region-based recursive localization method to improve the accuracy of defect detection.	Both our project and this paper use YOLOv5 for detecting defects, but in different contexts. Our project focuses on identifying multiple folds and the pipe's centre from internal footage, utilising bounding boxes for defect localization. In contrast, the paper deals with detecting internal corrosion in gas pipelines and employs techniques such as sample augmentation and advanced localization methods. While our work is on internal pipe defects, the methods and improvements discussed in the paper might provide valuable insights to enhance our detection approach.
2.	"A Surface Defect Detection Method for	Lili Wang, Chunhe Song,	The paper presents a new method for detecting surface defects in steel pipes using an enhanced YOLO framework.	Our project and this paper both use YOLO for defect detection but focus on different applications. We are using

	Steel Pipe Based on Improved YOLO"	Guangxi Wan, Shijie Cui	It addresses challenges like insufficient texture, high similarity among defects, and varying defect sizes by introducing a new backbone block, a refined neck block, and a novel regression loss function. These innovations improve the detection accuracy for both small and large defects, even in cases of high sample imbalance.	YOLOv5 to identify internal defects in pipes, such as multiple folds and the centre, with bounding boxes for localization. The paper, however, enhances YOLO for detecting surface defects in steel pipes, addressing challenges like texture and size variability. While we're dealing with internal pipe footage, the methods proposed in the paper could provide useful techniques for improving our defect detection, especially in handling various defect sizes and types.
3.	"Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets"	Trupti Mahendrakar, Andrew Ekblad, Nathan Fischer, Ryan T. White, Markus Wilde, Brian Kish, Isaac Silver	The paper evaluates how well YOLOv5 and Faster R-CNN perform in the context of autonomous navigation around space objects. It uses experimental data from formation flight simulations to test these algorithms under various conditions, such as different object motions and lighting scenarios. The study aims to assess how effectively these algorithms can identify and characterise objects for space missions, helping guide	Our project and the paper both employ YOLOv5 for object detection but in different applications. We use YOLOv5 to detect internal pipe defects like multiple folds and the centre of the pipe, utilising bounding boxes for localization. In contrast, the paper compares YOLOv5 with Faster R-CNN for autonomous space navigation, focusing on detecting and tracking non-cooperative targets under complex conditions. While our

			spacecraft without causing damage.	focus is on internal defect detection, the paper’s insights into YOLOv5's performance in challenging scenarios could offer valuable lessons for improving our defect detection methods and handling various operational challenges.
4.	"Object Detection Using YOLOv5"	L. Manikandan, Rapelli Malavika, Borlakunta Anjali, Sathuri Chandana	This paper delves into YOLOv5, a cutting-edge algorithm for detecting and locating objects in images and videos. It showcases YOLOv5's ability to identify people, vehicles, and animals quickly and accurately, offering a significant improvement over previous YOLO models. The paper tests YOLOv5 on the COCO dataset and reviews its strengths, limitations, and potential future enhancements.	Our project and this paper both use YOLOv5 for object detection, but for different purposes. We apply YOLOv5 to find internal pipe defects, such as multiple folds and the pipe's center, by training with bounding boxes and testing with new video footage. On the other hand, the paper focuses on YOLOv5’s performance in detecting objects like people and vehicles in various settings. While our work targets internal pipe defects, the insights from the paper could help us refine our methods and improve detection accuracy.

Chapter 3

Deep Neural Networks

R-CNN

1. **R-CNN (Region-based Convolutional Neural Networks)**

   R-CNN was one of the first models to successfully perform object detection by combining region proposals with Convolutional Neural Networks (CNNs).

   The key steps include:

   Selective Search: Propose candidate regions (region proposals) where objects might be located.

   Feature Extraction: Use a CNN to extract features from each proposed region.

   Classification and Bounding Box Regression: Classify each region and refine the bounding box coordinates using SVMs and linear regression.

   Advantages:

   - RCNN achieves high accuracy by combining region proposals with deep learning, allowing precise object localization and classification.
   - Leverages pre-trained CNNs for transfer learning, enhancing performance even with limited annotated data.

2. **Fast R-CNN**

   Fast RCNN improves upon RCNN by integrating the region proposal generation and feature extraction processes, significantly reducing computational time. It processes the entire image with a single CNN to extract a feature map, from which region proposals are classified and refined simultaneously. This makes Fast RCNN much faster and more efficient.

   Fast R-CNN improves upon R-CNN by sharing the convolutional computations:

   Single CNN Pass: The entire image is passed through a CNN once to produce a feature map.

   Region of Interest (RoI) Pooling: Extracts fixed-size feature vectors from the feature map for each region proposal.

   End-to-End Training: A single network that jointly optimises classification and bounding box regression.

Advantages:

- Faster than R-CNN due to shared computations.
- End-to-end training improves accuracy and simplifies the workflow.

3. **Faster R-CNN**

Faster RCNN further enhances Fast RCNN by introducing a Region Proposal Network (RPN) that generates region proposals directly from the feature maps, eliminating the need for an external proposal generation process. This integration allows for nearly real-time object detection with increased speed and accuracy. Faster R-CNN further optimises the process by introducing the Region Proposal Network (RPN):

RPN: Generates region proposals directly from the feature maps.

Shared Layers: Uses shared convolutional layers for both RPN and object detection tasks.

Advantages:

- Significantly faster due to the integrated RPN.
- High accuracy and efficiency.

4. **Mask R-CNN**

Mask R-CNN extends Faster RCNN by adding a branch for predicting segmentation masks on each region of interest (ROI). This allows Mask RCNN to perform pixel-level object segmentation in addition to object detection and classification, enabling more detailed image analysis. Mask R-CNN extends Faster R-CNN to perform instance segmentation:

Instance Segmentation: Adds a parallel branch for predicting segmentation masks for each region of interest.

Advantages:

- Can detect objects and provide pixel-level segmentation.
- Maintains high accuracy and flexibility.

YOLO (You Only Look Once)

Overview

YOLO (You Only Look Once) is a real-time object detection system that divides the input image into a grid and simultaneously predicts bounding boxes and class probabilities for each grid cell. Its single-shot approach allows for rapid and efficient object detection, making it suitable for applications requiring high-speed processing. YOLO is a family of object detection algorithms designed for real-time processing:

Single Forward Pass: Predicts bounding boxes and class probabilities directly from full images in one forward pass of the network.

Grid Division: Divides the image into a grid and predicts bounding boxes and probabilities for each grid cell.

YOLO Versions

1. **YOLOv1**: The original YOLO model introduced real-time object detection by framing detection as a single regression problem, predicting bounding boxes and class probabilities directly from full images in one evaluation.
2. **YOLOv2 (YOLO9000)**: Improved accuracy and speed over YOLOv1 by incorporating batch normalisation, high-resolution classifiers, and a new anchor box method, along with the ability to detect over 9000 object categories.
3. **YOLOv3**: Enhanced the detection capabilities with a deeper network, multi-scale predictions, and better performance on small objects using a more robust feature extractor called Darknet-53.
4. **YOLOv4**: Further optimised for speed and accuracy by integrating features like CSPDarknet53, Mish activation, and other advanced data augmentation and training techniques, making it more suitable for production environments.
5. **YOLOv5**: Focused on usability with improvements in speed and ease of deployment, incorporating features like auto-learning bounding box anchors and mosaic data augmentation, though it is not an official release by the original YOLO creators.

Advantages

- **Speed:** Real-time object detection, capable of processing images at high frame rates.
- **Unified Architecture:** A single network that directly predicts bounding boxes and class probabilities, simplifying the pipeline.
- **Good Accuracy:** Competitive accuracy with much faster inference times compared to traditional methods.

Why was YOLO used?

Reasons for Choosing YOLO:

1. **Simplicity**:
   Unified Architecture: YOLO (You Only Look Once) employs a unified architecture where the entire object detection pipeline is consolidated into a single neural network. This design contrasts with the more complex and multi-stage pipelines of models like R-CNN, which involve separate steps for region proposal, feature extraction, and classification.
   End-to-End Training: YOLO allows for end-to-end training, which simplifies the training process and minimises the need for multiple training phases and individual component optimization. This reduces the development time and makes it easier to integrate the model into existing systems.

2. **Efficiency**:
   One-Shot Detection: YOLO is a single-shot detection model, meaning it predicts bounding boxes and class probabilities in a single pass through the network. This approach is inherently more efficient than the region proposal methods used in R-CNN variants, which require multiple stages of processing.
   Speed: YOLO's single-stage nature leads to significantly faster inference times compared to R-CNN-based models. Faster R-CNN, for example, involves a region proposal network (RPN) that generates potential bounding boxes, followed by a second stage that classifies these proposals and refines their locations. This two-stage process

is computationally intensive and time-consuming, making YOLO a more suitable choice for applications needing quick processing.

3. **Versatility**:
   Handling Multiple Objects: YOLO's architecture is designed to handle multiple objects within a single frame effectively. It divides the input image into a grid and predicts bounding boxes and probabilities for each cell, enabling it to detect various objects simultaneously. This capability is crucial for tasks such as pipe inspection, where multiple defects may need to be identified within a single video frame.
   Robustness to Variations: YOLO can adapt to variations in object sizes, shapes, and orientations due to its grid-based prediction approach. This versatility is particularly beneficial in detecting diverse types of defects within pipes, ensuring comprehensive coverage and accurate localization.

Technical Comparison with R-CNN

1. **Single-Shot vs. Multi-Stage Detection:** YOLO's single-shot detection approach allows it to process images in one go, predicting bounding boxes and class probabilities together. In contrast, R-CNN models follow a multi-stage detection process involving separate steps for proposing regions, extracting features, and classifying them. This multi-stage process increases the computational load and inference time.

2. **Inference Time:** R-CNN and its variants, such as Fast R-CNN and Faster R-CNN, suffer from longer inference times due to their complex pipelines. YOLO, on the other hand, significantly reduces inference time by combining all detection steps into a single network pass, making it much faster and more efficient for practical applications.

Chapter 4

Experiments & Results

Common changes :

In utils folder , in the metrics.py file , we have changed the fitness for mAP-50 to be 0.9 and mAP-50-95 to be 0.1.

(i) Before Augmentation:

These are the results before augmentation.

Model	Precision	Recall	mAP@50	mAP@50-95	Multiple_Folds(P)	Multiple_Folds(R)	Multiple_Folds(mAP@50)	Multiple_Folds(mAP@50-95)	Time Taken (Hrs)	GFLOPs
YOLOv5s	0.894	0.833	0.906	0.536	0.790	0.766	0.818	0.333	0.261	15.8
YOLOv5m	0.881	0.881	0.899	0.546	0.763	0.763	0.804	0.323	0.396	47.9
YOLOv5l	0.882	0.872	0.895	0.553	0.766	0.744	0.795	0.357	0.572	107.7
YOLOv5x	0.900	0.885	0.905	0.543	0.806	0.806	0.815	0.337	0.915	203.8

(ii)After Augmentation:

These are the results after adding 600 augmented images in the training dataset.

Model	Precision	Recall	mAP@50	mAP@50-95	Multiple_Folds(P)	Multiple_Folds(R)	Multiple_Folds(mAP@50)	Multiple_Folds(mAP@50-95)	Time Taken (Hrs)	GFLOPs
YOLOv5s	0.888	0.876	0.901	0.502	0.781	0.751	0.807	0.327	0.299	15.8
YOLOv5m	0.889	0.880	0.897	0.540	0.780	0.760	0.800	0.341	0.466	47.9
YOLOv5l	0.898	0.887	0.906	0.545	0.798	0.773	0.817	0.346	0.659	107.7
YOLOv5x	0.877	0.840	0.898	0.524	0.822	0.681	0.801	0.349	1.087	203.8

(iii) Shuffled Dataset(After Augmentation):

These are the results after we combined the images and labels from the train and val folder including the 600 augmented images from the train folder and shuffled the dataset into a 60:40, 70:30, 80:20, 90:10 split using random seed.

Shuffle Ratio	Precision	Recall	mAP @50	mAP@ 50-95	Multiple_ Folds(P)	Multiple _Folds(R)	Multiple_F olds(mAP @50)	Multiple_Fol ds(mAP@50-95)	Time Taken (Hrs)	GFLOPs
60-40	0.891	0.756	0.776	0.532	0.837	0.654	0.677	0.352	0.309	15.8
70-30	0.905	0.751	0.774	0.546	0.854	0.648	0.674	0.359	0.303	15.8
80-20	0.889	0.752	0.764	0.55	0.831	0.659	0.664	0.367	0.295	15.8
90-10	0.882	0.748	0.755	0.565	0.823	0.661	0.654	0.386	0.300	15.8

(iv) Evolve:

These are the results where we have used the ‘evolve’ method which is inbuilt in YOLOv5.

Model	Precision	Recall	mAP@50	mAP@50-95	val/box_loss	val/obj_loss	val/cls_loss	lr0	lrf
Evolve 1	0.81789	0.52589	0.65329	0.32501	0.060725	0.050504	0.0013615	0.01	0.01
Evolve 2	0.89829	0.82269	0.85388	0.51825	0.038746	0.031005	0.00055112	0.01	0.01
Evolve 3	0.88992	0.85808	0.89061	0.51803	0.036879	0.026227	0.00011221	0.01	0.01

Chapter 5

Conclusion

In this project, we effectively utilised the YOLOv5 model to detect and localise defects within pipes, focusing specifically on identifying "multiple folds." Using MATLAB, we created bounding boxes around these defects and generated frames from videos, along with text files containing bounding box coordinates in YOLO format. These were used to form two folders of images and corresponding annotations. We expanded our dataset with various data augmentation techniques, including HorizontalFlip, VerticalFlip, RandomRotate90, HueSaturationValue, and RandomBrightnessContrast, to enhance the model's robustness.

We trained the dataset using YOLOv5 and achieved a final accuracy of 89%. The model's performance was validated on new, unseen video footage, where it successfully highlighted defects with bounding boxes similar to those used during training. This confirms that the model can reliably extend its learned knowledge to new data.

Using OpenCV (cv2), we processed the test video to generate an output that visually marks the detected defects, demonstrating the model's real-world applicability for automated pipeline inspections. This system offers a clear and actionable way to spot defects, crucial for efficient pipeline maintenance. Future work could involve expanding the dataset further and incorporating real-time monitoring capabilities. These improvements would bolster the model's accuracy and reliability, advancing the project towards more sophisticated and automated inspection solutions.

Chapter 6

Future Scope

The future scope of this project is vast and encompasses several enhancements that can significantly improve its functionality and applicability:

1. **Segmentation with YOLOv8:**

   Enhanced Defect Detection: Transitioning from localization to segmentation using YOLOv8 can provide more precise detection of defects within the pipe. Unlike bounding boxes, segmentation can outline the exact shape and size of the defects, offering more detailed insights.

   Improved Accuracy: Segmentation can improve the accuracy of defect detection by reducing false positives and providing a clearer understanding of the defect's context within the pipe.

   Advanced Analysis: Segmentation allows for advanced analysis such as measuring the area of defects, which can be crucial for assessing the severity and potential impact of the defects on the pipeline's integrity.

2. **Real-Time Defect Detection**:

   Instant Feedback: Implementing real-time defect detection can provide instant feedback during inspections, allowing for immediate corrective actions and reducing downtime.

   On-the-Fly Adjustments: Real-time detection enables on-the-fly adjustments to the inspection process, enhancing the efficiency and effectiveness of pipeline maintenance operations.

   Scalability: Real-time capabilities can be scaled to monitor multiple pipelines simultaneously, offering a broader application scope in large-scale infrastructure projects.

3. **User Interface with Logging and Analysis:**

   Interactive User Interface: Developing a user-friendly interface where users can input video footage and receive an annotated output with highlighted defects can streamline the inspection process.

Logging and Timestamping: Integrating logging features that record the timestamps of detected defects can facilitate detailed analysis and reporting. This feature will allow users to easily identify when and where defects occur, enhancing the traceability and documentation of inspections.

Comprehensive Output: Providing an output video with highlighted defects, along with a detailed log of the defect occurrences, can offer a comprehensive inspection report. This feature will improve decision-making processes by providing both visual and textual data.

References

[1] B. Chen, L. Ma, and S. Liang, "Automatic Detection of Internal Corrosion Defects in Natural Gas Gathering Pipelines Using Improved YOLOv5 Model," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11681-11686, 2023.

[2] L. Wang, C. Song, G. Wan, and S. Cui, "A surface defect detection method for steel pipe based on improved YOLO," *Mathematical Biosciences and Engineering*, vol. 21, no. 2, pp. 3016-3036, Jan. 2024, doi: 10.3934/mbe.2024134.

[3] T. Mahendrakar, A. Ekblad, N. Fischer, R. T. White, M. Wilde, B. Kish, and I. Silver, "Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets," in *2022 IEEE Aerospace Conference,* pp. 1-12, 2022.

[4] S. K. Ahuja and M. K. Shukla, "A survey of computer vision based corrosion detection approaches," in *Proc. International Conference on Information and Communication Technology for Intelligent Systems*, 2017, pp. 55–63.