Data-Driven Pipe Object Detection and Classification for Enhanced Inventory Accuracy and Cost Reduction Using Artificial Intelligence Techniques

Abhinav R. Diwakar[1]
[1]Research Associate, Innodatatics, Hyderabad, India

Kartik B. Kuri[2]
[2]Research Associate, Innodatatics, Hyderabad, India

Monika[3]
[3]Mentor, Research and Development, Innodatatics, Hyderabad, India

Nagamani Grandhe[4]
[4]Team Leader, Research and Development, Innodatatics, Hyderabad, India

Bharani Kumar Depuru[5*]
ORC ID: 0009-0003-4338-8914
[5]Director, Innodatatics, Hyderabad, India

Corresponding Author:- Bharani Kumar Depuru[5*]

Abstract:- Object detection has revolutionized industries like manufacturing, healthcare, and transportation by enabling automated object identification and classification in images and videos. This paper explores cutting-edge architectures like YOLOv3, YOLOv4 and YOLOv8, highlighting their remarkable strides in accuracy, speed, and robustness. YOLOv8, with its balanced performance and ease of implementation, has emerged as a leading architecture. However, practical challenges like overfitting, data collection difficulties, model complexity, and high hardware demands can hinder its real-world adoption.

This paper investigates prominent toolboxes like MMDetection and Detectron2 to address these challenges. Detectron2 provides optimization techniques and hardware acceleration strategies to tackle model complexity and hardware demands.

The paper also explores deploying YOLOv8 on Streamlit/Flask platforms, offering a user-friendly interface for interacting with the model and visualizing its detections, facilitating integration into web applications. Strategies for overcoming deployment challenges and achieving optimal performance are also discussed.

Looking ahead, the paper investigates data-driven pipe detection and classification for enhanced inventory management. This promising approach utilizes computer vision algorithms to automate pipe identification and categorization, potentially revolutionizing inventory management practices and streamlining operations.

By addressing YOLOv8 implementation challenges and exploring promising future directions, this paper contributes to the advancement of object detection technologies and their transformative impact across various industries.

Keywords:- *Object Detection, YOLOv8, Pipe Inventory Management, Image Processing, Artificial Intelligence, Computer Vision, Deep Learning, Object Tracking, Real-time Object Counting, Automated Inventory Management, Pipe Classification.*

I. INTRODUCTION

This article proposes a new automated pipe counting architecture based on the CRISP-ML(Q) methodology[Fig.1]. The CNN architecture is designed to Detect & classify the steel pipes in different shapes and dimensions.

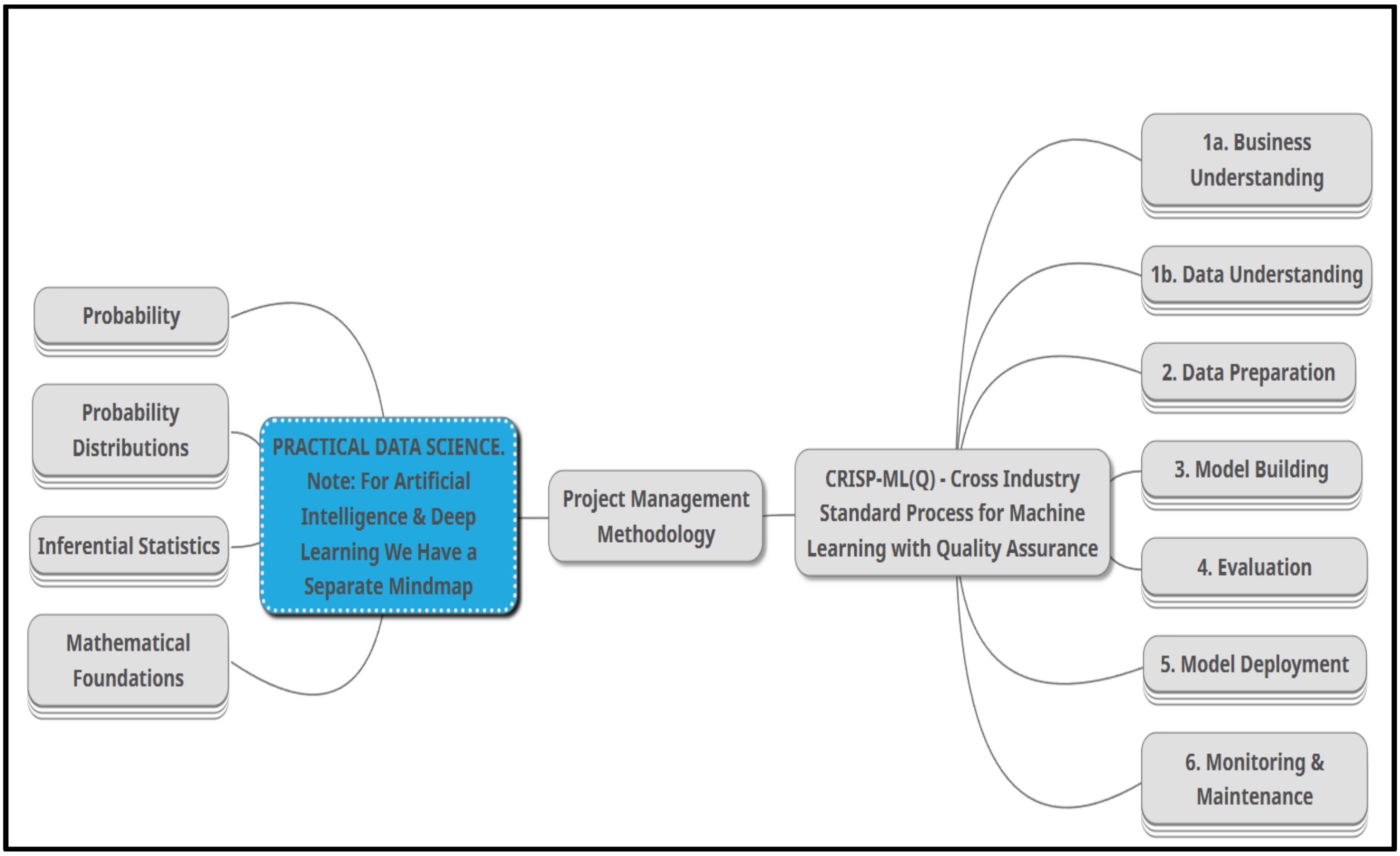

Fig 1 The CRISP-ML(Q) Architecture that we used for this Research can be Seen in the Figure Above
(Source: Mind Map - 360DigiTMG)

In the Business Understanding phase [Fig.1], we identified the business problem of manually counting and tracking [10] the number of steel pipes of different dimensions and shapes, which is a time-consuming and error-prone process that can lead to substantial waste of time and cost. Businesses can identify steel pipes of various sizes and shapes with high accuracy and efficiency by utilising machine learning for automated steel pipe detection and classification. This reduces the likelihood of human mistake and the need for manual intervention.

In the Data Understanding phase, we explored and understood the dataset of steel pipe images with different shapes and dimensions [11], and identified the need for data cleaning and preprocessing such as noise removal, resizing, and image augmentation via techniques such as random cropping, flipping, and rotating images.

In the data preparation phase [Fig.2], we pre-processed and transformed the data by converting it into images in a format of JPG that was suitable for model training. Additionally, we divided the data into test and training sets.

In the Modelling phase [Fig.3], we experimented with different machine learning models [5]and model architectures to identify the one that performs best on our task of automated steel pipe detection and classification. We also augmented the training set with additional data to improve the model's generalization performance.

Once the model is trained [6], we will evaluate its performance on a held-out test set to ensure that it meets our requirements. If the model does not meet our requirements, we will return to the Modelling phase and make further adjustments.

Once we are satisfied with the model's performance, we will deploy it to production [13] so that it can be used to detect and classify steel pipes in real-time. The model can be integrated into a software application or with cameras.

The proposed automated pipe detection system, which follows the CRISP-ML(Q) [Fig.1] methodology and focuses on modifying and using ML Workflow Architecture for machine learning models [Fig.2]., has the potential to improve the speed, accuracy, and efficiency of object detection, reduce the need for manual labour, and prevent the shipment of incorrectly counted materials. This could lead to significant cost savings and product loss reductions.

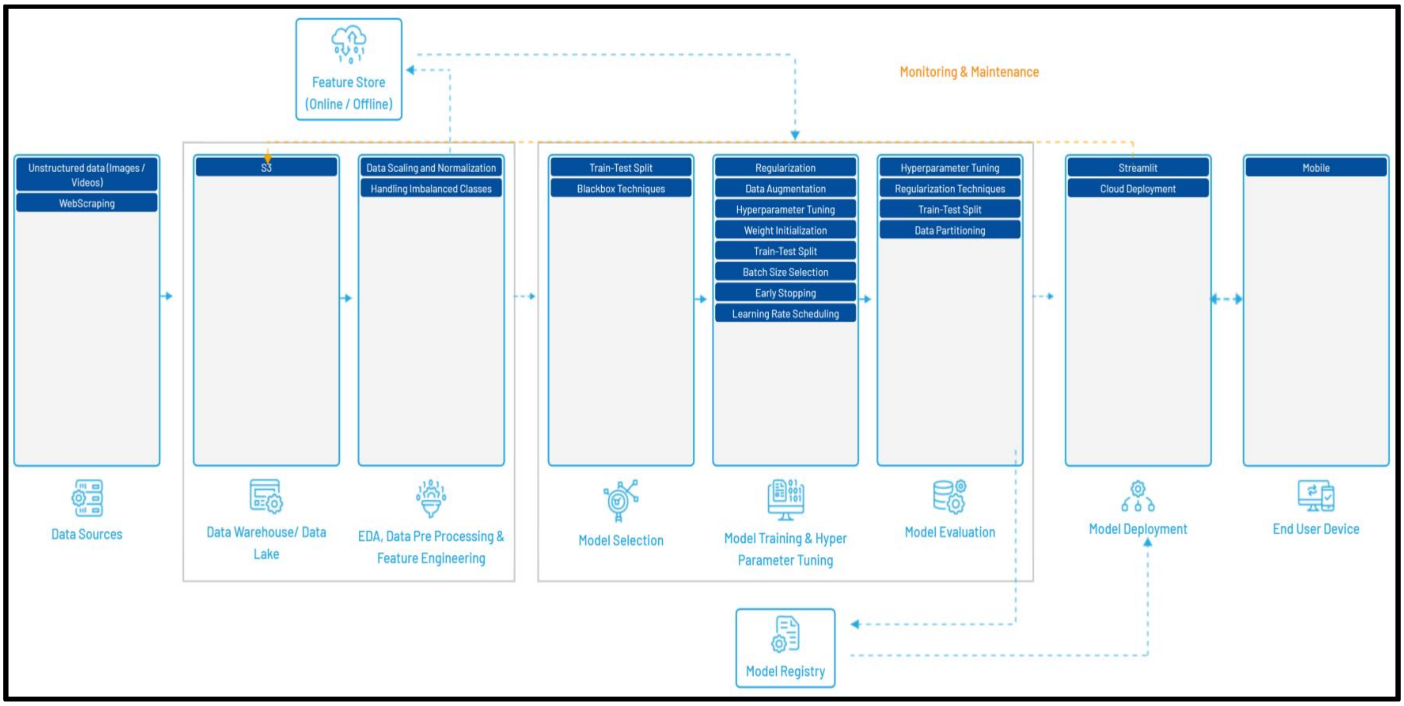

Fig 2 ML Workflow Architecture Used for the Research - A Detailed Overview of the Deep Learning Pipeline for Object Detection and Classification
(Source : ML Workflow - 360DigiTMG)

II. METHODS AND TECHNIQUES

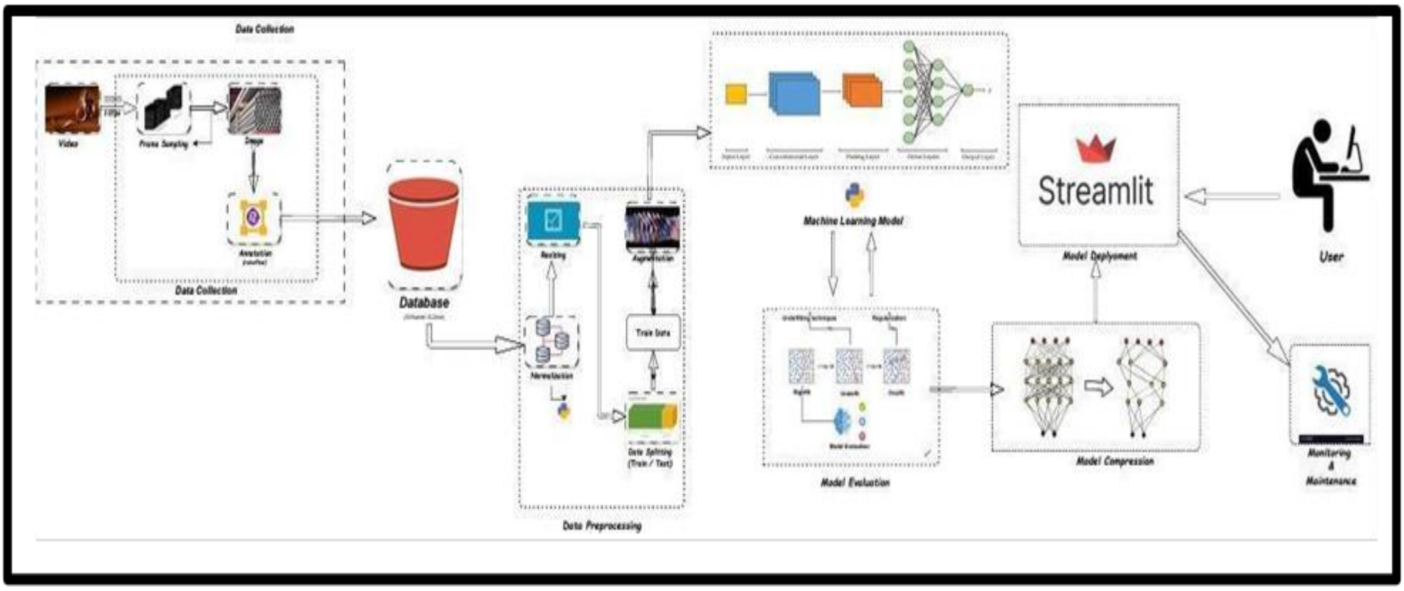

Fig 3 Architecture Diagram: Showcasing the Components and Flow of Data for Automated Pipe Detection and Classification using the CNN Machine Learning Technique
(Source: ML Workflow - 360DigiTMG)

As depicted in [Fig.3], Building and deploying a YOLOv8-based object detection model involves a well-defined workflow that encompasses data preparation, model training, evaluation, and deployment. Let's explore each phase of this procedure:

➢ *Data Collection:*

Securing and annotating data form the critical foundation for constructing effective machine-learning models for pipe detection and classification. These crucial steps involve acquiring and preparing the data that the models will be trained on, ensuring their ability to accurately identify and categorize pipes encountered in real-world settings.

The data [Table.1] employed in this study was sourced directly from a warehouse specializing in steel pipes. This comprehensive collection encompasses twenty-one distinct categories of pipes, featuring diverse shapes such as square, rectangular, and circular configurations.

ISSN No:-2456-2165

Table 1 Overview of the Data Used in This Research

Data Source	**Client Data**
Data Type	Unstructured
Video Format	Mov
Image Format	Jpg
Raw Images	450
Image Size	640x640
Augmented Images	1350
No. of Train Data	940
No. of Test Data	410
Total Data Size	2GB

➢ *Data Preprocessing*

This section details the data preprocessing steps undertaken to prepare the dataset for training the YOLOv8[5] model for steel pipe detection.

- *Image Extraction*

Video footage captured within the warehouse served as the primary data source. Images were subsequently extracted [4] from these videos utilizing the OpenCV [9] library. This process facilitated the creation of a dataset containing individual images for training and evaluation purposes.

- *Data Annotation*

Object annotation refers to the process of labelling and outlining specific objects or regions of interest within images. This process provides crucial supervised learning data for training machine learning models. Annotations typically involve:

✓ Drawing bounding boxes around pipes
✓ Specifying the pipe's class
✓ In some cases, marking key points or defining polygons for more detailed information

For this study, we utilized the polygon annotation tool within the Roboflow platform [Fig.4]. This tool allowed for precise and efficient labelling of steel pipes of various shapes and sizes, enabling the creation of a comprehensive and accurate training dataset.

- *Image Resizing*

To ensure consistency and uniformity throughout the training and analysis process, all images within the dataset were resized to a standardized dimension of 640x640 pixels [Table.1]. This standardized input size facilitated efficient model training and allowed for smoother comparison and analysis of results.

- *Data Augmentation*

The dataset was divided into training data comprising 80% and test data comprising 20%. To augment the training dataset [Table.1], various techniques were implemented to enhance variability and the model's robustness [12]. These included:

✓ Flip: Horizontal
✓ 90° Rotate: Clockwise, Counter-Clockwise
✓ Shear: ±15° Horizontal, ±15° Vertical
✓ Brightness: Between -25% and +25%

These data augmentation techniques significantly increased the size and diversity of the training set, leading to a more robust and generalizable YOLOv8[5] model for steel pipe detection [1].

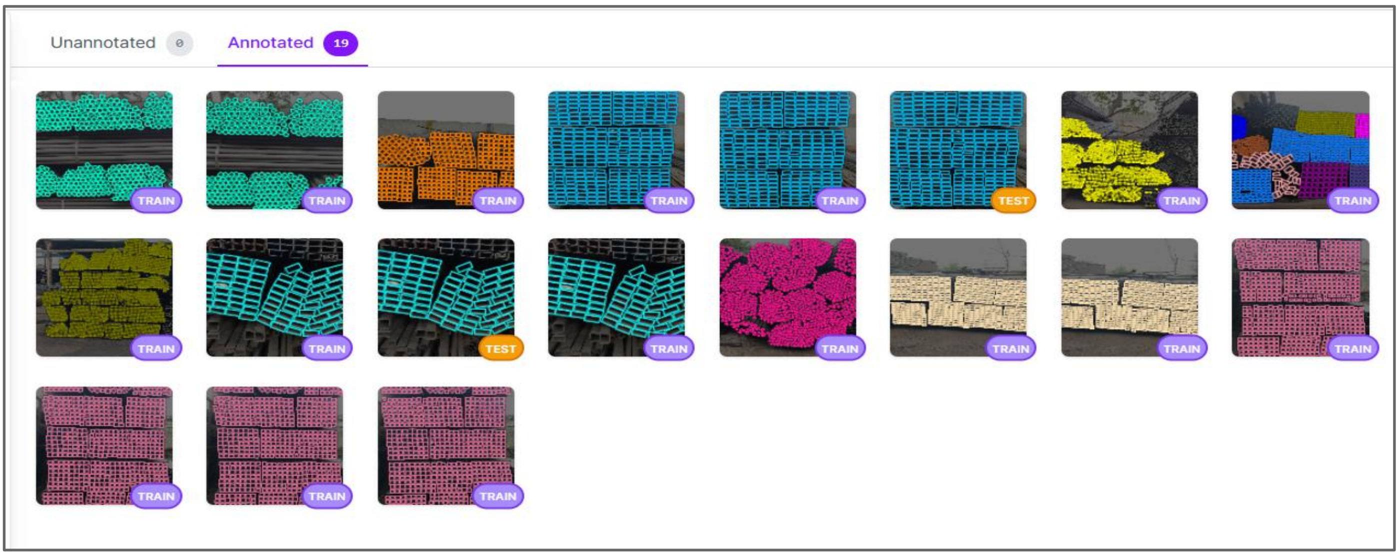

Fig 4 Displaying Preprocessing Task
(Source: https://app.roboflow.com)

ISSN No:-2456-2165

➢ *Model Training and Evaluation:*

To implement automated steel pipe detection and classification, the study has undergone training in various models like YOLOv8[5], YOLO NAS[2], DETR[6], Detectron2[3], and MM Detection. Each model has its advantages and disadvantages, so it is important to choose the right model for the specific needs of the application.

CNNs[8] are a type of deep learning model that is specifically designed for image classification tasks. CNNs work by extracting features from images using a series of convolutional and pooling layers. The extracted features are then used to train a classifier to predict the class of the image.

- *CNNs have Several Advantages for Automated Steel Pipe Detection and Classification:*

✓ CNNs can learn to extract features from images that are relevant for steel pipe detection and classification, such as the presence of different types of steel pipes, defects, and corrosion.
✓ CNNs are robust to noise and other variations in the data.
✓ CNNs can be trained to achieve very high accuracy on steel pipe detection and classification tasks.

- *YOLOv8:*

YOLOv8[5] is a state-of-the-art object detection model that is based on a CNN architecture. YOLOv8 is known for its speed and accuracy, making it a good choice for real-time object detection tasks such as automated steel pipe detection and classification.

- *YOLO NAS:*

YOLO NAS[2] is a neural architecture search (NAS) method for designing YOLO models. YOLO NAS models are typically more accurate and efficient than manually designed YOLO models.

- *DETR:*

DETR[6] is a transformer-based object detection model. DETR models are known for their ability to detect objects in images with complex backgrounds.

- *Detectron2:*

Detectron2[3] is a popular open-source object detection framework. Detectron2 provides a variety of pre-trained models for object detection tasks, including steel pipe detection and classification.

- *MM Detection:*

MM Detection[7] is another popular open-source object detection framework. MM Detection provides a variety of pre-trained models for object detection tasks, including steel pipe detection and classification.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	35	30404	0.904	0.395	0.621	0.424
C 48 2.5	35	915	0.797	0.637	0.743	0.44
C 48 3.2	35	2578	0.891	0.208	0.532	0.355
C 60 3.2	35	1	1	0	0	0
C 89 3.6	35	164	0.99	0.994	0.995	0.751
R 20 40 1.2	35	9683	0.968	0.248	0.602	0.448
R 20 40 2.5	35	786	0.186	0.0178	0.0596	0.0383
R 60 40 2	35	1244	0.997	0.263	0.63	0.441
R 80 40 2	35	5731	1	0.366	0.683	0.494
R 80 40 2.5	35	279	0.969	0.226	0.599	0.416
R 80 40 2.9	35	77	1	1	0.995	0.767
R 96 48 2.5	35	108	0.998	1	0.995	0.608
S 20 20 1.2	35	853	0.526	0.185	0.359	0.174
S 25 25 1.9	35	517	0.919	0.132	0.526	0.248
S 38 38 2	35	788	1	0.284	0.642	0.411
S 38 38 2.5	35	784	0.961	0.254	0.606	0.375
S 50 50 2	35	358	0.996	0.788	0.891	0.606
S 60 60 1.6	35	1151	0.999	0.521	0.759	0.591
S 72 72 1.6	35	4163	0.988	0.356	0.671	0.526
S 72 72 2	35	224	1	0.0223	0.511	0.368

Speed: 0.5ms preprocess, 2.9ms inference, 0.0ms loss, 36.6ms postprocess per image
Results saved to [1mruns/detect/train [0m

Fig 5 Best Model Training Summary (YoloV8)

Table 2 Shows the Train Accuracy and Test Accuracy of 5 Different Deep-Learning Models

S. No	Model	Train Accuracy (MAP-50)	Test Accuracy (MAP-50)	Hyperparameters
1	YOLOV8	62.00%	60.00%	Batch Size = 4, Epoch = 100, Image Size = 640x640
2	YOLO NAS	50.60%	46.76%	Batch Size = 4, Epoch = 30, Image Size = 640x640
3	DETR	48.00%	41.20%	Batch Size = 4, Lr = le-4, Lr Backbone=1e-5, weight decay = le-4, Max Epoch = 50, Image Size = 640x640
4	Detectron2	37.44%	27.80%	Batch size per image = 1, Max Iter = 2000, Eval period = 200, Base Ir = 0.001, num classes = 22, num workers = 2, mask format='bitmask', Image Size = 640x640
5	MM Detection	41.00%	33.80%	Batch Size = 1, Lr_start_factor = 1.0e-5, Epoch = 20, Dsl_topk = 13, Loss_cls_weight = 1.0, Loss_bbox_weight = 2.0, Qfl_beta = 2.0, Weight_decay = 0.05, train_num_workers = 2, Image Size = 640x640

The YOLOv8 [5] model's performance [Fig.5] was assessed using a rigorous set of evaluation metrics, including precision, recall, F1 score, Intersection over Union (IoU) [Fig.6], and mean Average Precision (mAP)[Table.2]. Precision measured the proportion of correct positive predictions among all positive predictions, while recall measured the proportion of correct positive predictions among all actual positives. The F1 score, a harmonic mean of precision and recall, provided a combined measure of model accuracy measured the overlap between predicted bounding boxes and ground truth boxes, while mAP provided a comprehensive summary of the model's overall performance

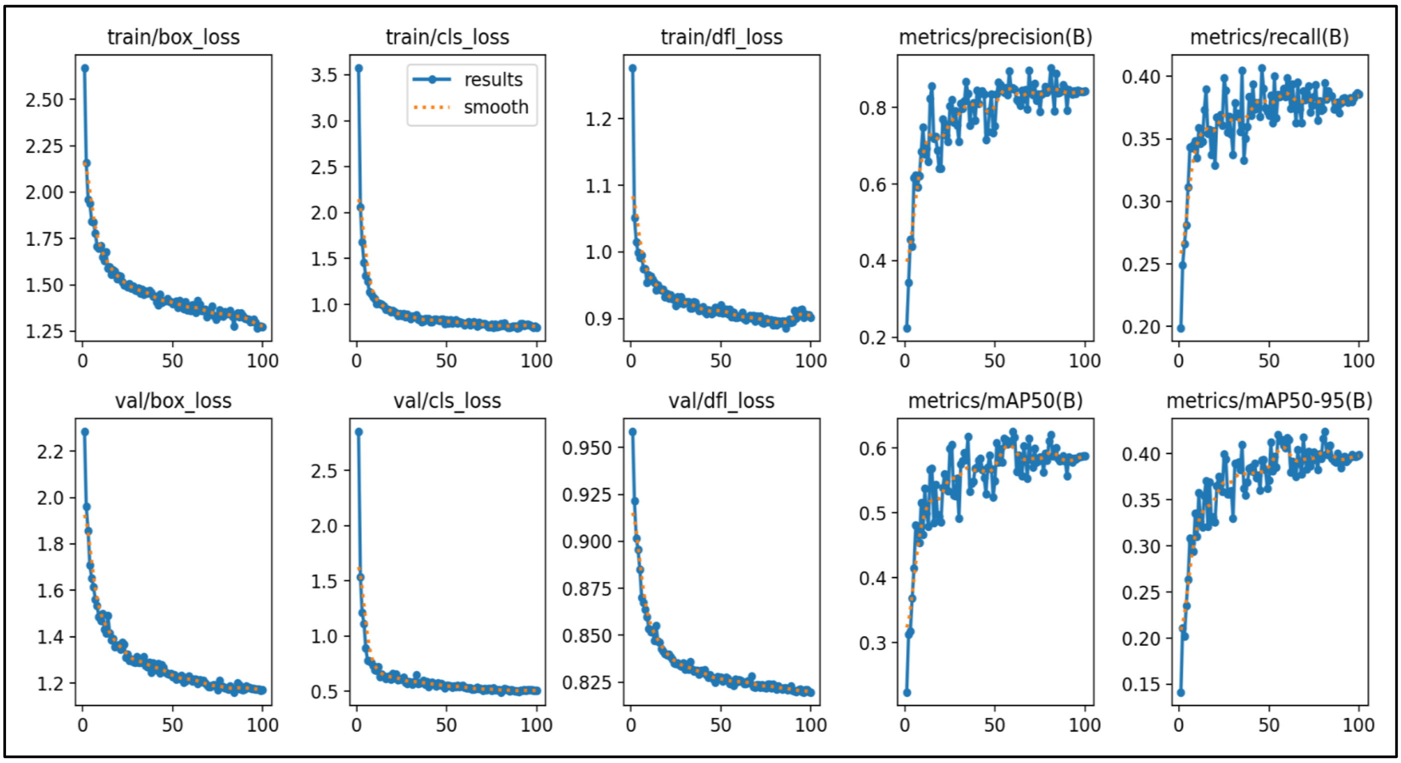

Fig 6 Visualization of Training Results of Yolov8

- *Deployment Strategy*

Model deployment is the process of making a trained machine-learning model available for use in a production setting. This involves integrating the trained model [Table.2] with an application or system so that it can be used to make predictions or generate output.

Streamlit[13] is an open-source Python framework for building data-driven web applications. It allows developers to quickly create web apps using Python by providing a simple and intuitive API for constructing web interfaces.

The Streamlit interface [Fig.7] enables users to upload new images. The uploaded image will undergo analysis by the model, which will predict the classes and display the predicted image with bounding boxes along the way.

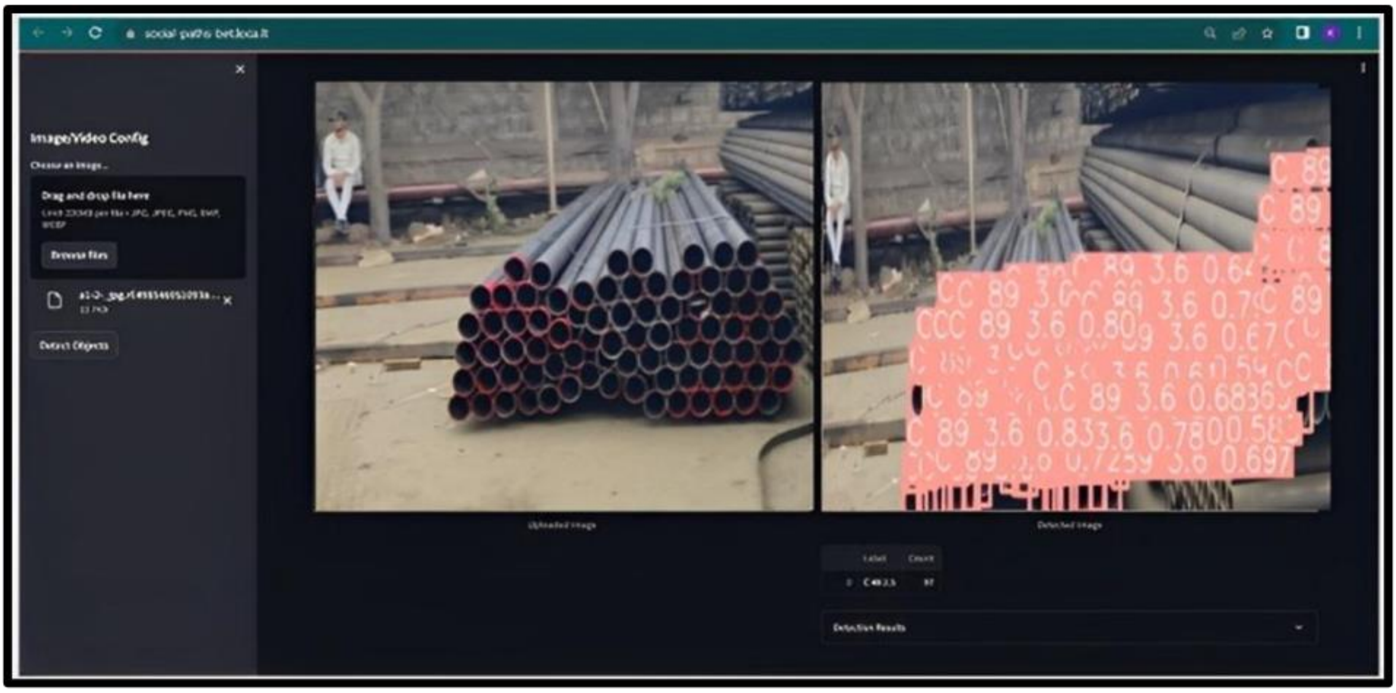

Fig 7 The Above Figure Depicts the Predicted Pipe Class and Counts using Streamlit Application

III. RESULTS AND DISCUSSION

Our study investigated the performance of five deep learning models for automated steel pipe detection and classification: YOLOv8, YOLO NAS, DETR, Detectron2, and MM Detection. The results presented in [Table.2] reveal that YOLOv8 outperforms the other models in terms of accuracy, achieving a train mAP50 of 62.00% and a test mAP50 of 60.00%. Beyond its high accuracy, YOLOv8 delivers significant improvements in efficiency and accuracy compared to manual counting. Its ability to reliably count pipes of various dimensions and shapes eliminates the human error and inconsistencies inherent in manual processes.

The findings of this study highlight YOLOv8 as a promising technology for automating steel pipe counting and tracking. Its high accuracy, efficiency, cost-effectiveness, and contribution to worker safety position it as a compelling solution for businesses in the manufacturing and warehousing industries. As the technology matures and further research refines its capabilities, YOLOv8 is expected to gain wider adoption across various industrial sectors, leading to enhanced productivity, safety, and efficiency in numerous applications.

IV. CONCLUSION

This research investigated the application of the YOLOv8 deep learning model for automating steel pipe detection and classification, historically a labour-intensive and error-prone process. The results demonstrate YOLOv8's potential to significantly improve efficiency, accuracy, and cost-effectiveness in this domain.

YOLOv8 achieved superior performance compared to other investigated models, exhibiting a train mAP@50 of 62.00% and a test mAP@50 of 60.00%. This high level of accuracy translates to reliable detection and classification of steel pipes, regardless of their dimensions and shapes.

These benefits, combined with YOLOv8's high accuracy and efficiency, position it as a compelling solution for automating steel pipe counting and tracking in various industries, particularly manufacturing and warehousing. As the technology continues to evolve and mature, YOLOv8 holds immense potential to revolutionize the way steel pipes are counted and tracked, ultimately contributing to greater efficiency, safety, and cost savings across various industries.

REFERENCES

[1]. Redmon, J., & Farhadi, A. (2018). "YOLOv3: An Incremental Improvement." *arXiv preprint arXiv:1804.02767*: https://arxiv.org/abs/1804.02767 (accessed: Nov. 20, 2023).

[2]. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934. 2020. https://arxiv.org/abs/2004.10934 (accessed: Nov. 20, 2023).

[3]. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., ... & Feng, Z. (2019). "MMDetection: Open MMLab Detection Toolbox and Benchmark." arXiv preprint arXiv:1906.07155:https://arxiv.org/abs/1906.07155 (accessed: Nov. 20, 2023).

[4]. Wu, Y., Kirillov, A., Massa, F., Lo, W. Y., & Girshick, R. (2019). "Context R-CNN: Long Term Temporal Context for Per-Camera .." arXiv preprint arXiv:1912.03538: https://arxiv.org/abs/1912.03538 (accessed: Nov. 20, 2023).

ISSN No:-2456-2165

[5]. Glenn Jocher and Ayush Chaurasia and Jing Qiu (2023). “yolov8_ultralytics” YOLOv8 - Ultralytics YOLOv8 Docs (accessed: Nov. 20, 2023).

[6]. Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko (2020). “End-to-End Object Detection with Transformers” [2005.12872] End-to-End Object Detection with Transformers (arxiv.org) (accessed: Nov. 20, 2023).

[7]. Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Boxuan Li,. and others (2019). “Welcome to MMDetection.” https://mmdetection.readthedocs.io/en/latest/ (accessed: Nov. 21, 2023).

[8]. Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu, and others(2019). "Object Detection with Deep Learning: A Review." Object Detection with Deep Learning: A Review (accessed: Dec 09, 2023).

[9]. P.Devaki, S.Shivavarsha, G.Bala Kowsalya, M.Manjupavithraa, E.A. Vima (2019). “Real-Time Object Detection using Deep Learning and Open CV.”www.ijitee.org/wp-content/uploads/papers/v8i12S/L110310812S19.pdf (accessed: Dec 09, 2023).

[10]. Weili Fang, Lieyun Ding, Hanbin Luo, Peter E.D. Love (2018) “Falls from heights: A computer vision-based approach for safety harness detection.” https://www.sciencedirect.com/science/article/abs/pii/S0926580517308403 (accessed: Dec 09, 2023).

[11]. Vehicle Detection and Identification Using Computer Vision Technology with the Utilization of the YOLOv8 Deep Learning Method, October 2023, SinkrOn 8(4):2150-2157, DOI:10.33395/sinkron.v8i4.12787

[12]. Development of an Image Data Set of Construction Machines for Deep Learning Object Detection | Journal of Computing in Civil Engineering | Vol 35, No 2 (ascelibrary.org)

[13]. Streamlit at Work,Mohammad Khorasani,Mohamed Abdou & Javier Hernández Fernández Streamlit at Work | SpringerLink