ISSN No:-2456-2165

Optimizing Construction Efficiency: AI-Powered Inventory Management for Steel Rods through Automated Counting and Image Processing

Bajjuri Pavani[1], Asit Kumar Das[2], Nagamani Grandhe[3], Deba Chandan Mohanty[4], Bharani Kumar Depuru[5]
[1,2] Research Associate, Innodatatics, Hyderabad, India.
[3]Mentor, Research and Development, Innodatatics, Hyderabad, India.
[4]Team Leader, Research and Development, Innodatatics, Hyderabad, India.
[5]Director, Innodatatics, Hyderabad, India.

*****Corresponding Author: Bharani Kumar Depuru
ORC ID: 0009-0003-4338-8914

Abstract:- In the construction, manufacturing, and various industrial sectors, the routine task of manually counting steel rods is known to be laborious, time-consuming, and error-prone. This paper focuses on addressing the challenge of object detection and accurate counting of steel rods—a task of critical importance across various computer vision applications.

The approach taken here excels in swiftly, accurately, and robustly counting steel rods under diverse conditions encountered at construction sites. The workflow commences with the conversion of video content into image format, a preliminary step that streamlines subsequent processes, including annotation and augmentation.

For the automated counting of rods, a diverse range of models were employed, designed to process both images and videos. These models not only provide precise counts of the rods but also furnish valuable insights into rod diameter, enhancing the depth of information available. In this article, a comprehensive comparison of model accuracies can be found, which is a crucial step in identifying the best-performing model for this task.

The model proposed here offers a transformative solution by eliminating the need for manual counting efforts, effectively mitigating the potential for human errors that plague traditional counting methods. Powered by advanced image processing techniques, this system not only accelerates the counting process but also substantially enhances accuracy. Consequently, it introduces substantial time and cost savings for construction projects.

Keywords:- *Counting Steel Rods, Artificial Intelligence, Image Preprocessing, Advanced Models.*

I. INTRODUCTION

The act of counting steel rods may appear simple, yet for a construction and manufacturing study, it holds high importance. It's not just about mere numerical enumeration; it's also about upholding project schedules, optimizing inventory management, cost containment, and ensuring the structural robustness of the final output. Hence, the utilization of cutting-edge methods such as computer vision models becomes pivotal. These sophisticated techniques not only refine the counting process but also stand as pillars for the triumph and safety of construction ventures. [1]

In the construction domain, steel reinforcement is the backbone that ensures the strength and durability of buildings and infrastructure projects. Counting and managing steel bars, also known as rebars. Traditionally, this process has been labor-intensive and susceptible to errors. The approach to this research study is based on the CRISP-ML(Q) methodology available as open-source on the 360DigiTMG website[Fig.1].

In an effort to automate the counting of steel rods in construction projects, the process was initiated by converting video footage into a more manageable format: images. This foundational step laid the groundwork for developing an automated counting system, acknowledging the need for efficiency in construction site operations.

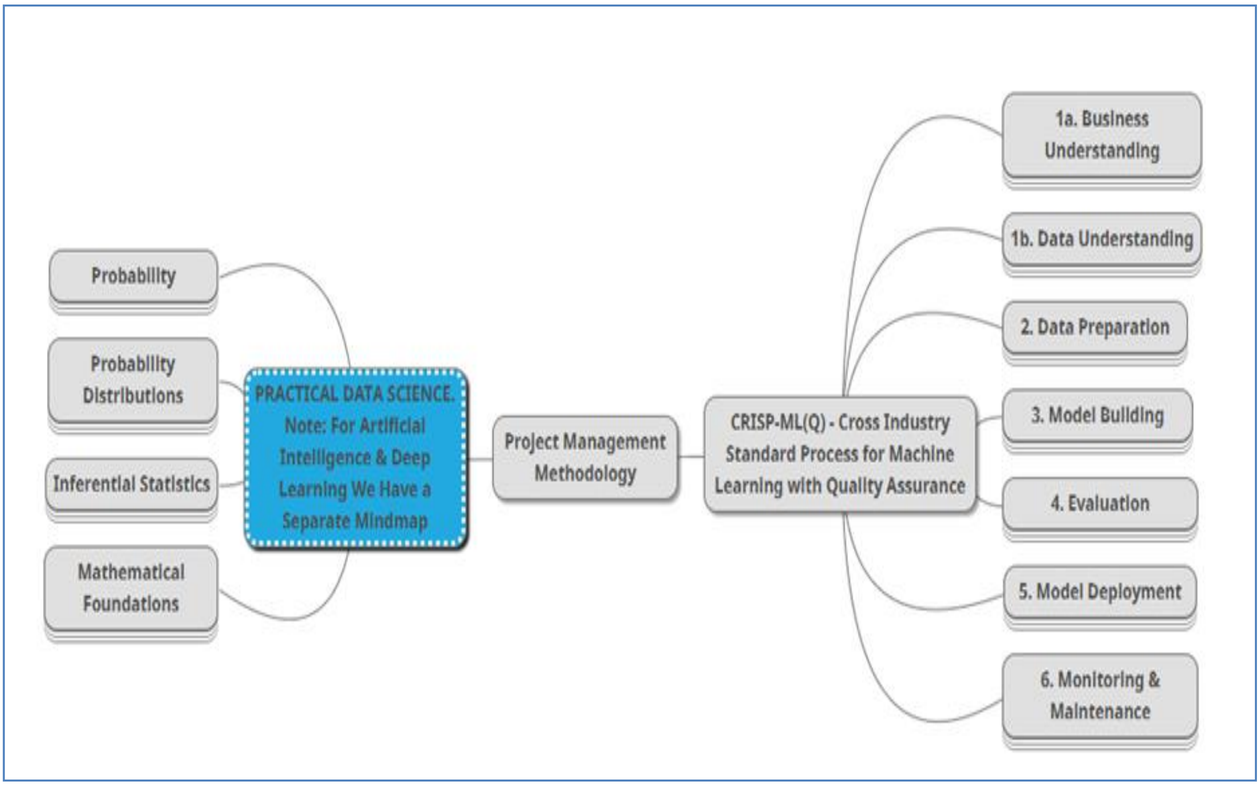

Fig. 1: The Above Figure Displays the CRISP-ML(Q) Architecture that was Used for this Research Project. (Source: Mind Map - 360DigiTMG)

After extracting the image data, the next crucial phase involved annotation. Each steel rod underwent precise labeling to create an annotated dataset, providing essential ground truth data. This step was essential for training the system effectively and ensuring its accuracy in identifying and counting steel rods.

Augmentation was applied on the annotated dataset to enhance the robustness of the automated system. These augmentations aimed to diversify the data, enabling the system to handle a broad spectrum of real-world scenarios. This prepared dataset set the stage for the subsequent modeling phase.

The foundation of the automated counting system involved assessing multiple contemporary computer vision models, such as ResNet, Efficient Net, Faster R-CNN, CNN-DC, YOLOv3, YOLOv4, YOLOv5, SSD, SA-CNN-DC, and YOLOv7, among others. Following extensive testing, YOLOv8 emerged as the top performer, renowned for its precise object detection and efficiency in counting. YOLOv8 demonstrated remarkable suitability for accurately tallying steel rods in images. Its exceptional performance significantly elevated the automation and precision of the steel rod counting process, establishing YOLOv8 as the preferred model among those assessed. [3,5]

Following successful model development, attention shifted to the deployment phase. The automated counting system seamlessly integrated into real-world applications, marking the transition from development to practical use. Notably, the conversion of video footage into images facilitated a manageable format for deployment.

The implementation of YOLOv8 brought tangible benefits, improving the efficiency and accuracy of steel rod counting on construction sites. Beyond mere automation, the system offered potential cost savings and increased productivity. This technological advancement holds significant promise for the construction industry, where precise counting is paramount for project success.[4,7]

The envisioned automated system for TMT rebar detection, aligning with the CRISP-ML(Q) methodology [Fig1.1] and leveraging the ML Workflow Architecture for model development [Fig.2], holds promise in enhancing the swiftness, precision, and efficiency of TMT rebar identification. By minimizing reliance on manual intervention, it aims to mitigate errors in material counting during shipments, potentially resulting in substantial cost savings and reductions in product losses.

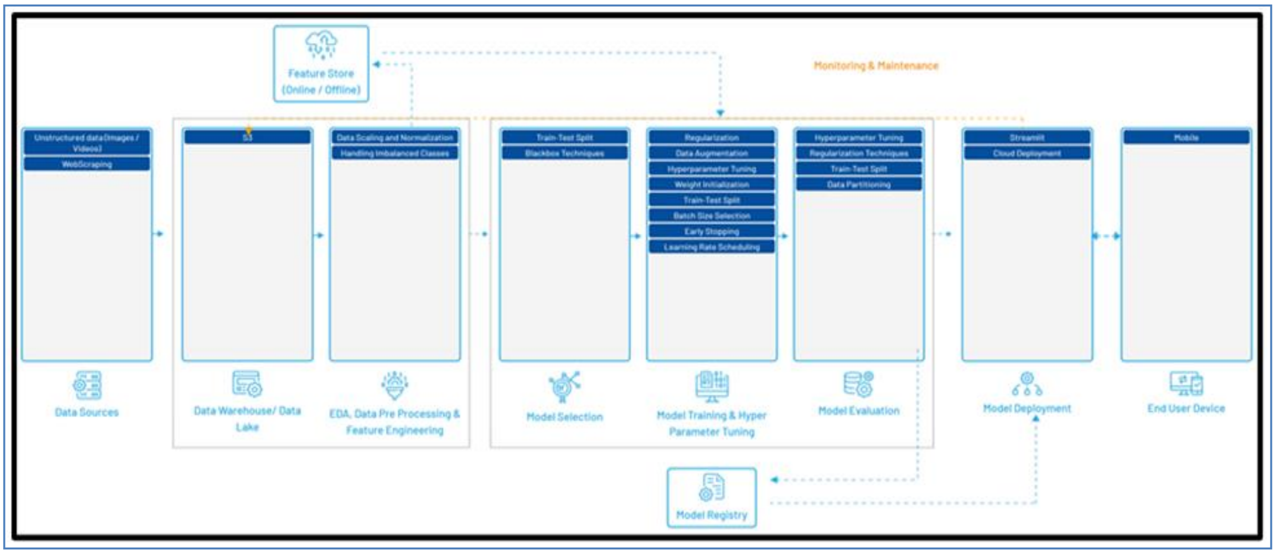

Fig. 2: ML Workflow Architecture Used for the Research - A detailed overview of the Deep Learning Pipeline for Object detection and classification (Source:ML Workflow - 360DigiTMG)

II. METHODS AND TECHNIQUES

- **Architecture Diagram:** This architecture integrates advanced computer vision models like YOLOv8 with augmented datasets, aiming for precise object detection and seamless real-world integration [Fig.3].

To get accurate results for our project, first the collected data was organized. Then, a model was created that understands this organized data well. After making sure the model works correctly, it was put into action in real-world situations. This way, the model can give reliable information to the people using it. The key is to go from getting the data ready to making a useful model, testing it, and finally using it to provide accurate results. [2]

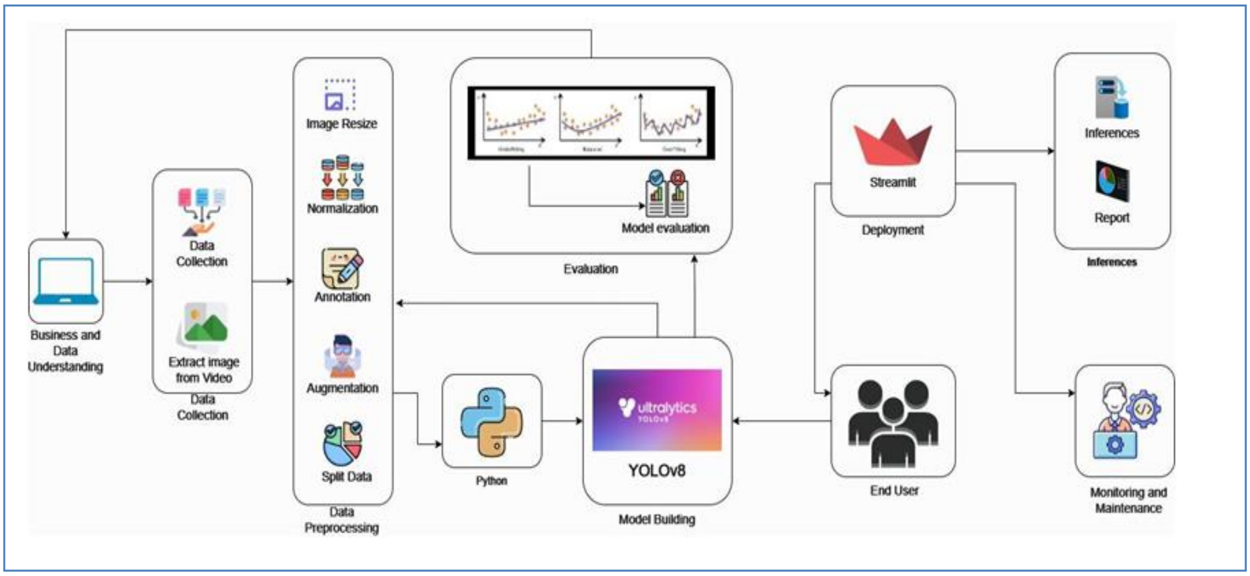

Fig. 3: Architecture diagram of data flow from one stage to another

A. *Data Collection:*

The foundation of the automatic steel rod counting system relies on the precise collection of data. In this endeavor, a rich dataset consisting of videos and images were collected, each featuring steel rods of various diameters. This diverse dataset was not randomly assembled; rather, it was thoughtfully collected to ensure it reflects the real-world complexities encountered in construction projects.

Within the construction industry, variations in steel rod diameters are common occurrence. The dataset was purposefully designed to capture these variations comprehensively. It not only included standard and commonly used rod diameters but also encompassed fewer common sizes. This approach ensured that the automated steel rod counting system was fully prepared to handle the entire spectrum of construction projects, regardless of the specific diameter requirements.

The dataset included three types of steel rods: 8mm, 16mm, and 32mm. For 8mm, there were 73 images and a video were present. In the 16mm category, there were 193 images and one video. The 32mm group consisted of 223 images and a video. These images and videos were essential for training and testing our automated counting system for steel rods. [Fig.4]

It's crucial to understand that data collection is an ongoing process. As new construction projects appear and building regulations change, the dataset must adapt and grow to stay in sync with these changes. So, it is imperative that there is continuous data.

In summary, the process of data collection forms the core of our automatic steel rod counting system. By thoughtfully assembling a diverse dataset of videos and images showcasing steel rods of varying diameters it is ensured that our system is not only accurate but also highly adaptable, making it a valuable asset in the ever-evolving world of construction projects. [6,9]

Fig. 4: Sample raw data

B. Preprocessing

➢ *Videos to Frames Conversion*

In the process of automating steel rod counting, the initial and crucial step involves converting videos into individual frames. Roboflow is being utilized for this task because it doesn't employ a one-size-fits-all approach; rather, it analyzes the video's duration and adapts the frame generation accordingly. For shorter videos, it generates fewer frames, while longer ones receive more frames, ensuring the dataset's accuracy and detail regardless of the specific characteristics of the video.

This dynamic approach to frame generation has a crucial impact. It means that the dataset is well-suited to the specific visual content of each video, which can capture steel rods at various speeds and levels of detail. This adaptability promotes uniformity in the data, making it easier to annotate, label, and train the automated system effectively.

Roboflow is a powerful platform that facilitates the conversion of videos into frames, allowing for efficient analysis and processing of individual images. One key parameter in this process is the Frames Per Second (fps), which determines the rate at which frames are extracted from the video. [Fig.5][10]

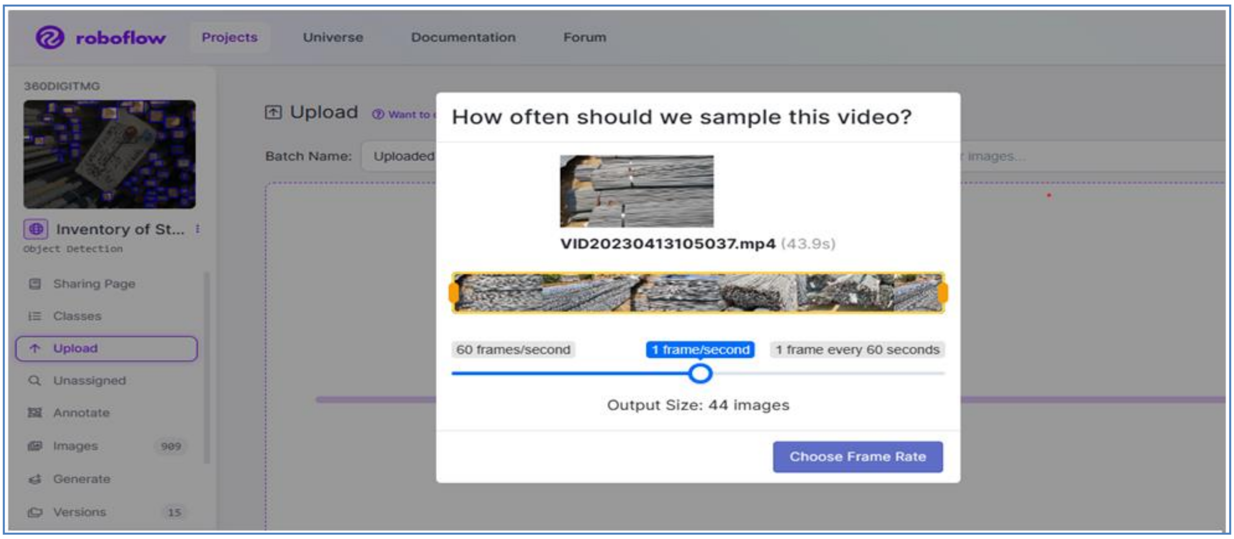

Fig. 5: Converting Video into Frames

ISSN No:-2456-2165

The number of frames (images) generated can be calculated using the following formula:

Number of Images = Chosen fps × Length of the Video

Where:

- Number of Images: Total frames extracted from the video.
- Chosen fps (Frames Per Second): User-specified parameter indicating the rate of frame extraction.
- Length of the Video: Total duration of the video in seconds.

For example,

Consider a video that is 60 seconds long, and frames are captured at a rate of 30 frames per second (fps).

Number of Images=30 fps × 60 seconds=1800 images

In this example, processing the entire 60-second video at 30 fps results in 1800 frames.

Steps to Generate Frames with Roboflow:

- Access Roboflow Platform: Log in to the Roboflow platform.
- Upload Video: Upload the video file to the Roboflow interface.
- Set Frames Per Second (fps): Specify the desired fps based on your analysis requirements.
- Generate Frames: Initiate the frame generation process to extract frames from the video using the chosen fps.
- Download Frames: Once the frame generation is complete, download the extracted frames for further analysis.

2.2.2 Data Annotation:

Data annotation is a crucial phase in the quest to automate steel rod counting. This process involves the systematic labeling and categorization of each image or frame within the dataset. Each labeled image serves as a reference point, enabling the automated system to learn and precisely count steel rods. Beyond basic counting, data annotation [Fig.6] includes detailed information about the diameter, position, and other relevant characteristics of each steel rod. This level of granularity empowers the system not only to count but also to classify steel rods, contributing to a more comprehensive analysis.

In many instances, data annotation requires collaboration between human annotators and artificial intelligence. Human experts, well-versed in understanding the details of steel rods, provide the initial labels, which are further refined through machine learning models. Quality assurance procedures play a vital role in the annotation process, ensuring the correctness of labels and reducing errors and inconsistencies within the dataset. This approach guarantees that our automated system is trained on accurate and reliable data.

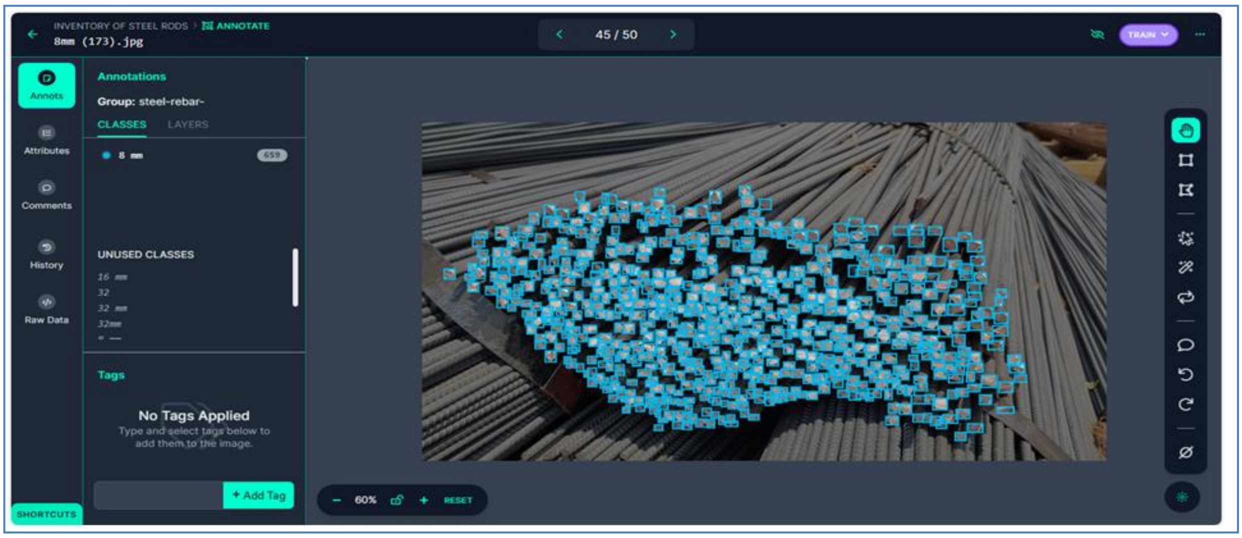

Fig. 6: Sample Annotation of data

- *Data Augmentation:*

Data augmentation is another step in automating steel rod counting. It involves techniques like rotation, scaling, and noise addition to enrich the dataset. This introduces diversity, mirroring real-world construction scenarios and helping the system better understand steel rod variations [Fig.7].

Data augmentation prevents overfitting, where the model becomes too specialized. By diversifying the data, it reduces this risk and enhances the model's ability to handle various real-world scenarios.

Data augmentation isn't static; it evolves with changing construction practices, ensuring the dataset remains robust. In essence, it's a dynamic and essential phase that enriches our dataset and boosts our automated steel rod counting system's adaptability.

ISSN No:-2456-2165

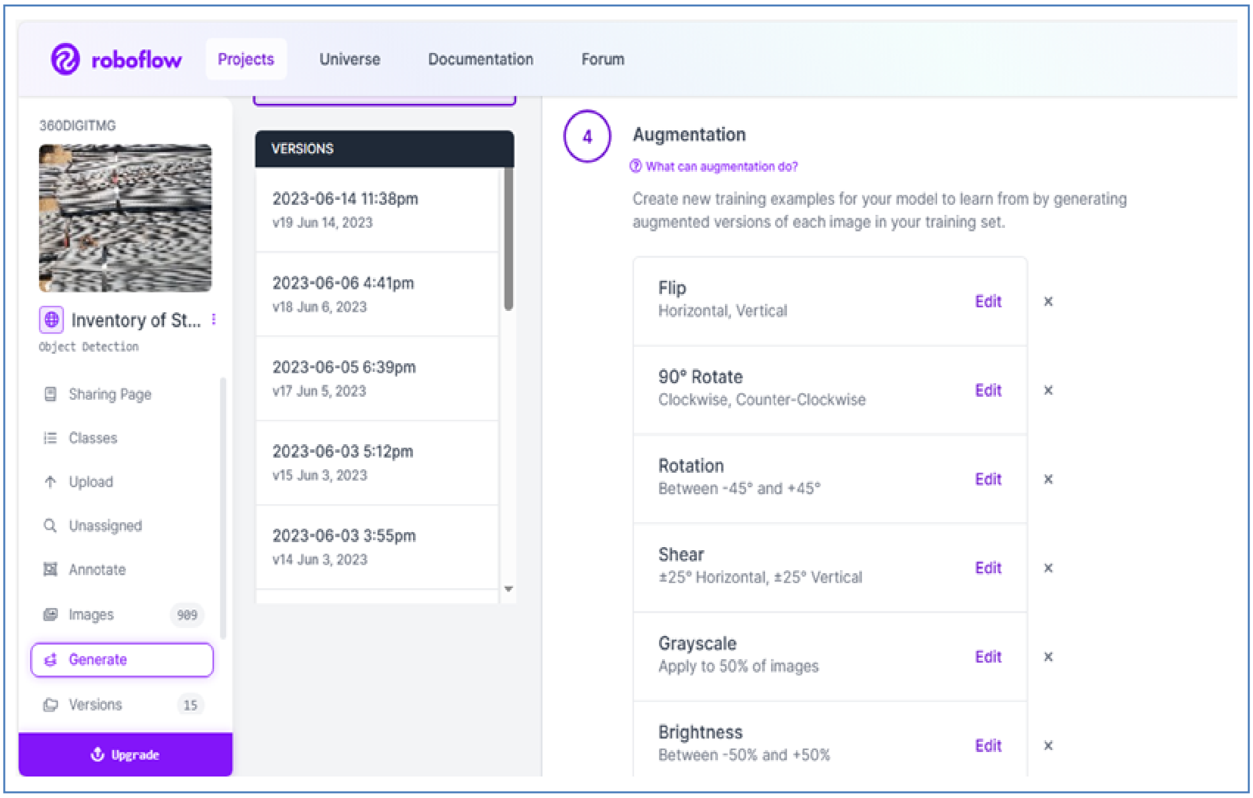

Fig. 7: Augmentation in Roboflow

➢ *Data Set Splitting:*

Data set splitting is a pivotal step in refining our automated steel rod counting system. It involves segregating our dataset into specific subsets, each with a unique purpose [Fig.8].

The training set forms the basis for our model to learn the details of steel rod counting. The validation set evaluates the model's performance and guides fine-tuning, ensuring it generalizes effectively. A test set provides an independent assessment of the model's real-world readiness.

Data set splitting strikes a balance between learning and adapting, enhancing our model's accuracy and adaptability. In summary, it's a crucial step in building a robust steel rod counting solution.

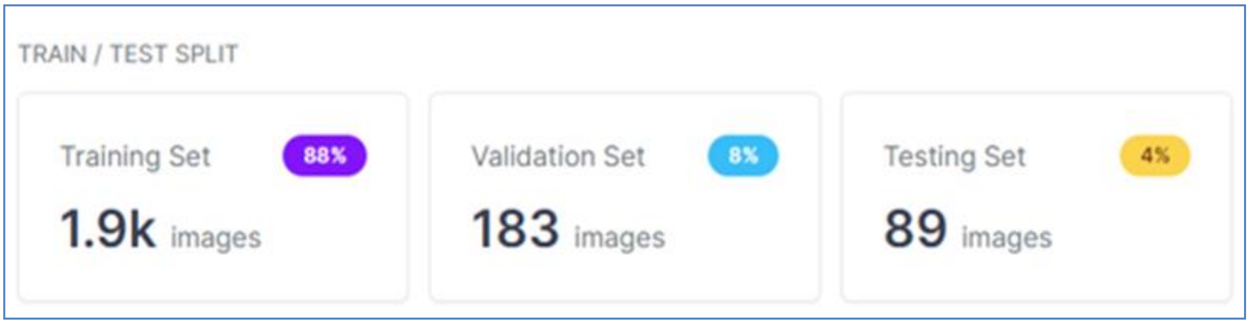

Fig. 8: Splitting the dataset

➢ *Data description table:*

In the initial phase of data preparation, a seamless transformation was carried out on a collection of MP4 format videos, resulting in the creation of a standardized image dataset. A total of 909 raw images were generated through this conversion process, subsequently being combined with existing JPEG images to form a cohesive dataset. This foundational set of images captured diverse scenes and scenarios from the original videos, setting the stage for subsequent data augmentation.

To enrich the dataset's complexity and enhance model performance, various data augmentation techniques were applied, leading to a total of 2183 images. [Fig.9] [8]

These augmented images were strategically divided into three subsets:

- 1911 for training,
- 183 for validation, and
- 89 for testing.

ISSN No:-2456-2165

Each subset serves a specific purpose in the model development lifecycle, with the training set facilitating the learning process, the validation set aiding in fine-tuning and hyperparameter optimization, and the testing set evaluating the model's generalization to new, unseen data. The inclusion of three distinct classes within the dataset ensures that different visual patterns within the data can be effectively discerned and classified by the trained model. This diverse dataset can now be used to empower the training of machine learning models for a variety of applications, from image classification to more advanced tasks such as object detection.

C. YOLOV8 model approach:

The project is focused on automating the detection and counting of steel rods, not only in static images but also within dynamic videos. At the heart of this mission lies the YOLOv8 model, or "You Only Look Once" version 8. YOLOv8 is a renowned object detection model celebrated for its efficiency and precision.[11]

The initial phase of the project involves converting videos into individual images. These images form the basis for our data annotation process in Roboflow. Each image is labeled, providing essential information about steel rod characteristics, including diameters and positions. Subsequently, the annotated images undergo resizing and augmentation, diversifying the dataset and enhancing the model's adaptability.

Data Source	Client data
video format	MP4
image format	jpg
Raw images	909
classes	3
Augmented data	2183
No.of Train images	1911
No.of Test images	89
No.of valid images	183
Total data size	257

Fig. 9: Dataset Details after Preprocessing

The augmented dataset becomes instrumental in constructing our YOLOv8 model. Then various YOLOv8 versions were explored, such as YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, assessing their performance. YOLOv8x emerges as the standout choice, consistently delivering exceptional results [Fig.10].

Model	# of Layers	Size (pixels)	mAP@[.5:.95]	Speed (ms)	Architecture
YOLOv8n	21	640	36.7	121	CSPDarknet53
YOLOv8s	26	640	44.6	147	CSPDarknet53
YOLOv8m	35	640	49.9	218	CSPDarknet53
YOLOv8l	41	640	52.3	279	CSPDarknet53
YOLOv8x	53	640	53.9	402	CSPDarknet53

Fig. 10: Different Version Details of YOLOv8 Model

The YOLOv8 model is supported by a Convolutional Neural Network (CNN) architecture, comprising two vital components: the backbone and the head. The backbone features a modified CSPDarknet53 architecture with 53 convolutional layers and cross-stage partial connections, enhancing information flow [Fig.11]. The head is composed of multiple convolutional layers followed by a series of fully connected layers. This configuration enables the prediction of bounding boxes, object scores, class probabilities and the model's accuracy.

A notable feature of YOLOv8 is the use of a self-attention mechanism in the head of the network. This mechanism enables the model to focus on different parts of an image, adjusting the importance of various features based on their relevance to the task. Additionally, a feature pyramid network is employed to detect objects of varying sizes and scales within an image, ensuring comprehensive coverage.

In summary, our project utilizes the power of YOLOv8 to automate steel rod detection and counting in both images and videos. Through careful data preparation and model selection, the capabilities of YOLOv8x can be leveraged to create a system ready to transform steel rod counting in construction, enhancing efficiency and precision.

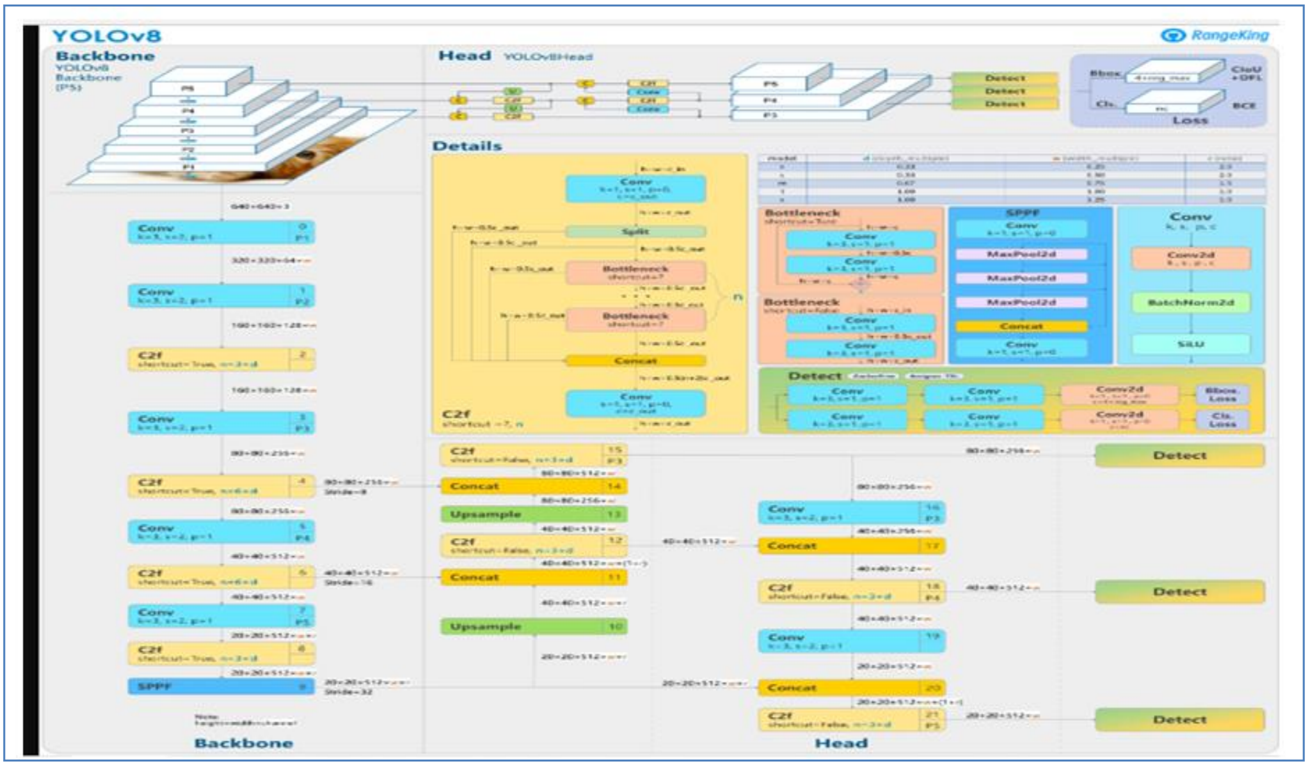

Fig. 11: YOLOv8 Architecture Diagram

➢ *Comparison of YOLOv8 versions and Hyperparameters:*

Different versions of YOLOv8 like YOLOv8n, YOLOv8s, YOLOv8l, and YOLOv8x can be compared for how they work by checking certain things. The tables were made to show important details for each version, like how well they learn, remember, and understand different things. These details include training box loss, class loss, recall, precision, DFL loss, mAP50B, and mAP50-95B. By doing this, it can be seen which version might be better for different tasks, helping us choose the right one based on what is needed to be done. [Fig.12,13,14,15,16]

YOLOv8n							
epoch	train/box_loss	train/cls_loss	train/dfl_loss	:s/precision(B)	etrics/recall(B)	rics/mAP50(B)	/mAP50-95(B)
0	3.4616	2.8228	1.4174	0.27814	0.22591	0.20583	0.07223
1	3.231	2.0122	1.3021	0.44971	0.35673	0.33449	0.11614
2	3.136	1.7811	1.2947	0.54333	0.38901	0.42421	0.15493
3	3.0543	1.7032	1.2871	0.53082	0.38503	0.4064	0.14765
4	3.0546	1.6607	1.2753	0.58734	0.40714	0.4447	0.16257

Fig. 12: Hyper Parameter Details of YOLOv8n Version

As the models undergo longer training periods, their capacity to understand and improve, measured by mAP accuracy, becomes more apparent. A larger batch size, representing the amount of data processed together during training, contributes to enhanced model performance. Notably, the table [Fig.17] illustrates varying accuracies across models, ranging from 0.447 to 0.5606. These numerical differences highlight YOLOv8x as the standout performer, showcasing superior accuracy in recognizing features within the training data.

YOLOv8n							
epoch	train/box_loss	train/cls_loss	train/dfl_loss	:s/precision(B)	etrics/recall(B)	rics/mAP50(B)	/mAP50-95(B)
0	3.4616	2.8228	1.4174	0.27814	0.22591	0.20583	0.07223
1	3.231	2.0122	1.3021	0.44971	0.35673	0.33449	0.11614
2	3.136	1.7811	1.2947	0.54333	0.38901	0.42421	0.15493
3	3.0543	1.7032	1.2871	0.53082	0.38503	0.4064	0.14765
4	3.0546	1.6607	1.2753	0.58734	0.40714	0.4447	0.16257

Fig. 13: Hyper Parameter Details of YOLOv8n Version

YOLOv8s							
epoch	train/box_loss	train/cls_loss	train/dfl_loss	:s/precision(B)	etrics/recall(B)	rics/mAP50(B)	/mAP50-95(B)
0	3.3189	2.0606	1.4299	0.55533	0.39896	0.42217	0.1515
1	3.0736	1.6996	1.3074	0.59203	0.42606	0.45587	0.16342
2	3.0026	1.619	1.2814	0.60602	0.43073	0.47209	0.17555
3	2.9316	1.5631	1.2685	0.61999	0.44074	0.4854	0.18134
4	2.8993	1.5442	1.2472	0.6168	0.46195	0.49515	0.18629

Fig. 14: Hyper Parameter Details of YOLOv8s Version

YOLOv8L							
epoch	train/box_loss	train/cls_loss	train/dfl_loss	:s/precision(B)	etrics/recall(B)	rics/mAP50(B)	/mAP50-95(B)
0	3.2304	1.8973	1.4233	0.56384	0.40647	0.41169	0.15033
1	3.0671	1.7229	1.3199	0.36657	0.33659	0.28004	0.1024
2	3.0206	1.6609	1.2938	0.54933	0.41476	0.44104	0.16281
3	2.9721	1.6455	1.2857	0.60616	0.43671	0.47065	0.17607
4	2.9305	1.581	1.2655	0.58786	0.45086	0.47309	0.17741
5	2.9192	1.5579	1.2682	0.61828	0.48203	0.51633	0.19943
6	2.8801	1.5207	1.2505	0.61274	0.4893	0.51313	0.19258
7	2.8507	1.5032	1.2466	0.63093	0.48768	0.52402	0.20178
8	2.8214	1.4786	1.24	0.58442	0.51919	0.52247	0.19849
9	2.8175	1.4544	1.2418	0.63041	0.49759	0.52855	0.20559
10	2.7868	1.4235	1.2223	0.62135	0.50478	0.525	0.19964
11	2.7589	1.4141	1.2193	0.63003	0.49883	0.53753	0.21145
12	2.775	1.418	1.218	0.60668	0.4896	0.51045	0.19305
13	2.7522	1.3888	1.2109	0.62024	0.49645	0.52752	0.20794
14	2.7415	1.374	1.2109	0.6412	0.51604	0.55611	0.21898
15	2.7118	1.373	1.204	0.6444	0.52575	0.55393	0.21818
16	2.6818	1.3561	1.1976	0.6482	0.5164	0.5532	0.21729
17	2.6953	1.3497	1.1933	0.63843	0.51282	0.54781	0.21555
18	2.6815	1.3475	1.1937	0.64692	0.51756	0.55847	0.22003

Metric	Epoch 0	Epoch 19
Box loss	3.2304	2.678
Class loss	1.8973	1.3323
DFL loss	1.4233	1.1848
Precision (B)	0.56384	0.53066
Recall (B)	0.40647	0.53066
mAP50 (B)	0.41169	0.56032
mAP50-95 (B)	0.15033	0.21981

Fig. 15: Hyper Parameter Details of YOLOv8l Version

YOLOv8X							
epoch	train/box_loss	train/cls_loss	train/dfl_loss	:s/precision(B)	etrics/recall(B)	rics/mAP50(B)	/mAP50-95(B)
0	3.2308	1.918	1.423	0.50685	0.38691	0.38109	0.13795
1	3.067	1.7325	1.3278	0.57705	0.42259	0.45485	0.16522
2	3.0292	1.6666	1.3045	0.5619	0.41106	0.44492	0.16544
3	2.9762	1.6465	1.2927	0.60973	0.42987	0.46727	0.17304
4	2.9418	1.5837	1.2737	0.60031	0.45154	0.47017	0.17537
5	2.9168	1.5695	1.274	0.61602	0.45382	0.49567	0.19008
6	2.873	1.5201	1.2525	0.60662	0.48287	0.49852	0.18617
7	2.8408	1.4921	1.2472	0.61959	0.48287	0.51705	0.19806
8	2.8199	1.4666	1.2425	0.59731	0.51498	0.52594	0.20477
9	2.8093	1.4452	1.2438	0.6337	0.49526	0.53161	0.20637
10	2.7805	1.4292	1.2236	0.61443	0.50438	0.52358	0.19809
11	2.7534	1.4198	1.2203	0.63289	0.50523	0.53859	0.20967
12	2.7701	1.421	1.2199	0.61329	0.48255	0.50994	0.19342
13	2.7495	1.3881	1.213	0.61575	0.49487	0.5224	0.20701
14	2.7257	1.3683	1.2076	0.64142	0.51452	0.55411	0.21943
15	2.7046	1.3671	1.2055	0.65293	0.52299	0.56065	0.2219
16	2.6764	1.352	1.1988	0.65115	0.51924	0.55687	0.21901
17	2.6888	1.3455	1.1933	0.63984	0.51418	0.55179	0.2172
18	2.6719	1.3438	1.1921	0.65164	0.51886	0.5601	0.22112

Metric	Value
Box loss	2.6704
Class loss	1.327
Detection loss	1.1844
Precision (B)	0.52703
Recall (B)	0.55978
mAP50 (B)	0.55978
mAP50-95 (B)	0.2219

Fig. 16: Hyper Parameter Details of YOLOv8x Version

ISSN No:-2456-2165

MODEL	mAP	EPOCHS	BATCH SIZE	IMAGE SIZE
YOLOv8s	0.4951	5	4	640
YOLOv8n	0.447	5	4	640
YOLOv8l	0.5603	20	4	640
YOLOv8x	0.5606	20	4	640
YOLOv8m	0.56	20	4	640

Fig. 17: Accuracy Comparisons of YOLOv8 Versions

D. Deployment Strategy

Inventory management for steel rods involves the deployment of a trained model, making it available for seamless integration into a production environment. The process encompasses associating the trained model with an application or system, allowing it to efficiently generate predictions and manage inventory data.

In this context, leveraging Streamlit, a Python-based open-source web application framework, proves to be advantageous. Streamlit simplifies the development of interactive and data-driven applications, providing a user-friendly API for creating web interfaces using Python.

➢ *Streamlit Application*

The Streamlit application designed for inventory management facilitates user interaction in various ways. Users can upload images or video of steel rods, initiating a process wherein the model analyzes the uploaded image or video. The model then predicts the class or category to which the steel rod belongs. [Fig. 18]

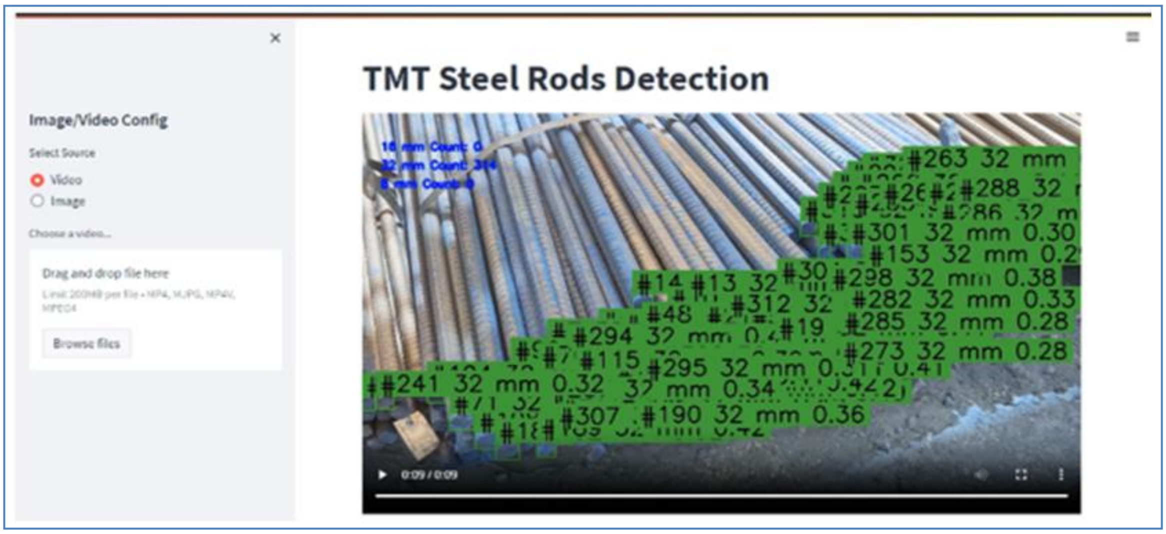

Fig.18: Streamlit Output

III. RESULTS AND DISCUSSION

The project's comprehensive exploration of YOLOv8 model versions highlighted the exceptional performance of YOLOv8x, making it the ideal choice for steel rod counting in both images and videos. The model's accuracy in recognizing steel rod characteristics, including diameters and positions, was a significant breakthrough.

The implications of the automated steel rod counting system for the construction industry are substantial. By improving accuracy and efficiency, reduced errors and safer buildings can be anticipated, while also expediting construction projects.

The investigation into key hyper-parameters demonstrated their pivotal role in shaping model behavior. The default values provided valuable starting points for further optimization, emphasizing the importance of hyper-parameter tuning in model development.

Looking ahead, our project opens the door to a range of potential applications, not only in construction but also in domains like manufacturing and inventory management that require object detection and counting.

Nonetheless, the system is not without limitations. Challenges include data quality and diversity, and the need for continuous model updates to adapt to evolving construction practices.

In conclusion, the project underscores the transformative potential of YOLOv8x for automated steel rod counting in construction. Safety, efficiency, and accuracy in construction projects stand to benefit, and the integration of advanced technologies and ongoing research will further enhance our system's capabilities and impact.

IV. CONCLUSION

This article using the YOLOv8 model for counting steel rods has the potential to make construction safer, faster, and more accurate. It can be useful in other areas too. While there are challenges, the work shows how technology can improve the way things are built, making them better and more efficient.

REFERENCES

[1]. Hernández-Ruiz, Ana & Martinez-Nieto, Alejandro &Buldain Pérez, Julio David. (2021). Steel Bar Counting from Images with Machine Learning. Electronics. 10. 10.3390/electronics10040402.
[2]. Shin, Yoonsoo, Sekojae Heo, Sehee Han, Junhee Kim, and Seunguk Na. 2021. An Image-Based Steel Rebar Size Estimation and Counting Method Using a Convolutional Neural Network Combined with Homography *Buildings* 11, no. 10: 463. https://doi.org/10.3390/buildings11100463
[3]. Yang Li, Yujie Lu, Jun Chen, A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector, Automation in Construction, Volume 124, 2021, 103602, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2021.103602.
[4]. Y. Wu, X. Zhou and Y. Zhang, "Steel bars counting and splitting method based on machine vision," 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 2015, pp. 420-425, doi: 10.1109/CYBER.2015.7287974.
[5]. Liu, H., Xu, K. (2021). Densely End Face Detection Network for Counting Bundled Steel Bars Based on YoloV5. In: Ma, H., *et al.* Pattern Recognition and Computer Vision. PRCV 2021. Lecture Notes in Computer Science(), vol 13019. Springer, Cham.
[6]. Gowroju, S., Santhosh Ramchander, N., Amrita, B., Harshith, S. (2023). Industrial Rod Size Diameter and Size Detection. In: Singh, P.K., Wierzchoń, S.T., Tanwar, S., Rodrigues, J.J.P.C., Ganzha, M. (eds) Proceedings of Third International Conference on Computing, Communications, and Cyber-Security. Lecture Notes in Networks and Systems, vol 421. Springer, Singapore.
[7]. Dang, DT., Wang, JW., Luong, V.N., Ha, V.L., Truong, V.T. (2023). Count the Number of Steel Bars Based on Deep Learning. In: Nguyen, T.D.L., Lu, J. (eds) Machine Learning and Mechanics Based Soft Computing Applications. Studies in Computational Intelligence, vol 1068. Springer, Singapore.
[8]. Ablidas, Mikale Rizton&Monsura, Armil &Ablidas, Lhora& Cruz, Jesse. (2019). An Application of Image Processing Technology in Counting Rebars as an Alternative to Manual Counting Process. International journal of simulation: systems, science & technology. 10.5013/IJSSST.a.20.05.02.
[9]. Fan, Z., Lu, J., Qiu, B., Jiang, T., An, K., Raj, A.N., & Wei, C. (2019). Automated Steel Bar Counting and Center Localization with Convolutional Neural Networks. ArXiv, abs/1906.00891.
[10]. Ciaglia, Floriana, Francesco Saverio Zuppichini, Paul Guerrie, Mark McQuade and Jacob Solawetz. “Roboflow 100: A Rich, Multi-Domain Object Detection Benchmark.” ArXiv abs/2211.13523 (2022): n. pag.
[11]. Reis, Dillon, Jordan Kupec, Jacqueline Hong and Ahmad Daoudi. “Real-Time Flying Object Detection with YOLOv8.” ArXiv abs/2305.09972 (2023): n. pag.