AI-Driven Concrete Quality: Automated Failure Analysis for Sustainable Construction

*M.S.Patil[1], R.B. Ghongade[2] and H.B. Dhonde[3]

[1] Assistant Professor, Indira College of Engineering and Management and Research Scholar at Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India
[2] Professor, E&Tc Bharati Vidyapeeth(Deemed to be University) College of Engineering, Pune, India
[3] Director,C-Probe Technologies Pvt. Ltd. Pune, India
meenakshi.madgunaki@indiraicem.ac.in

Abstract. This research aims to develop a system which is used to identify the concrete cube failure mode as acceptable or non acceptable. In the given system we have used image processing algorithm to detect the cracks on concrete cube and machine learning algorithms like YOLO version 10 for image cropping, annote and to analyse the cube images which are collected from lab experiments , on construction sites and some are augmented.

After implementing the system it is observed that the system above to detect the crack patterns and 90% accuracy in classifying the cube under acceptable and non acceptable category this reduces significantly the human error in classifying the cubes.Due to this f concrete evaluation become easier and faster and also this reduces the material waste and which in turn support sustainable construction practices. The future work includes increasing the dataset for more validity and developing a mobile application to make the process faster and user friendly.

Keywords: Concrete cube, Image processing, Machine learning, YOLOv10, sustainable practices, failure modes.

1 Introduction

We all know Concrete is widely used in construction sites as it provide strength, durability and versatility to the construction sites. In fact, after water, it is the most consumed material worldwide, reflecting its importance in modern infrastructure (Sadouskaya et al., 2021; Girma & Demiss, 2022). As concrete is widely used then it is important to ensure the quality and reliability and of that concrete the concrete cubes are made and that are also used in all the construction sites . Typically compressive strength test of concrete cubes is used(Kim et al., 2022).

In a compressive strength test, three cubes are casted and tested. In this we have considered the concrete with a maximum aggregate size below 20 mm, cubes of 100 mm × 100 mm × 100 mm and for larger aggregates require 150 mm × 150 mm × 150 mm cubes. During testing, a controlled compressive load is applied at a rate of 0 to 14 N/mm² per minute. Standards recommend that the variation in strength between cubes

M. Kaushik et al. (eds.), *Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, Advances in Intelligent Systems Research 210,
https://doi.org/10.2991/978-94-6239-723-1_36

should not exceed ±15%. The strength of each cube is calculated by dividing the applied load by the surface area under compression. These measurements help determine whether the concrete meets the required performance standards for construction.

Even with standardized testing, concrete cubes may show unsatisfactory failure modes. These can result from improper specimen preparation, handling errors, or limitations in testing equipment (Kim et al., 2022). The concrete cube failure mode are classified as satisfactory or non-satisfactory based on the pattern of cracks and how the damage is there on it and overall analysis of the concrete cube. Cubes with Satisfactory failures show a uniform crack pattern on all the four faces except the top and bottom face, indicating the concrete has responded appropriately to the applied load. In non-satisfactory failures modes of the concrete cube due to poor material composition or giving less time for curing, the irregular crack pattern and damage appear on the concrete cube . This leads to the unsatisfactory failure mode of the cube. Examples of both types of failure are shown in Fig. 1 and Fig. 2.

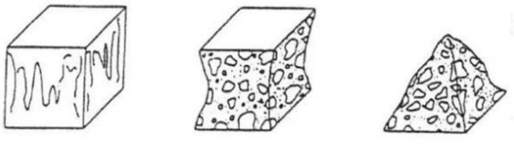

Fig. 1. Satisfactory failures of cube inspection.

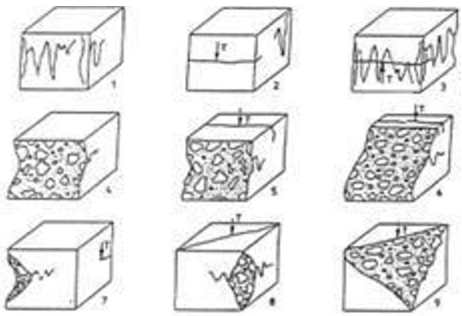

Fig. 2. Unsatisfactory failure mode of cube inspection.

The unsatisfactory failure patterns in concrete cubes may be due to problems in the specimens themselves or from issues with the testing equipment. The casting tray should be proper with as per the standard dimensions and the specimens should maintain uniform geometry. Premature demoulding, before the concrete has gained sufficient strength, can damage edges or corners and often leads to non-satisfactory failure modes. Likewise, large variations in cube size can affect how the load is distributed during testing, causing irregular fractures. Due to these challenges there is requirement of a reliable and human error free method for classifying the concrete cube failure modes to reduce the accidents in construction sites.

There is advancement in industry 4.o technologies and computer vision, image processing due to this there is significant improvement and automation in the testing process. Due to these tools we can identify failure pattern which helps in accurate assessment of concrete structures.

Currently much research is focused to evaluation of concrete strength and crack detection in the construction structural elements, and very less attention is given to identify the cube failure modes. In many construction practices, failure mode assessment is not done or performed visually, which leads to inappropriate decisions. As a result, many cubes are misclassified, which results in unnecessary rejection and this increased use of cement, which affects to environmental with higher carbon emissions.

In this study this gap is addressed by developing a semi-automatic system that uses image processing and machine learning algorithm to identify and classify concrete cube failures modes. The proposed system is based on Industry 4 in construction which analyzes crack patterns, damage area distribution to determine whether a cube is acceptable or not.. Due to less human intervention, this approach improves testing accuracy, and promotes sustainable construction practices by making optimum use of material and wastage is also minimized [12].

2 Image Processing and Machine Learning for Concrete Analysis

Compressive strength test is one of the crucial performance metrics in building and to evaluate already existing structures [9]. Now a days many image processing and machine learning algorithms are used to make more sensible and economical decisions by accurately estimating compressive strength.

In recent years, the construction industry using image processing techniques regularly to enhance structural analysis, enhance safety and reduce the risk of failures [10]. Image processing algorithms are used for detection of cracks and analyzing the crack pattern and to detect damage in concrete structures. These image processing techniques we can use to inspect areas which are difficult to reach and unsafe to access this also minimize the manual intervention and improve the accuracy of the data captured for further evaluation

In [10] assessment of crack is categorized into two approaches: image processing–based techniques and targeting-based techniques. Image processing methods are effective at identifying visible cracks, although there is surface irregularities or noise in the actual cracks. Targeting-based approaches can precisely locate cracks but difficult to locate exact location with geometry. Due to this there is requirement of more advanced methods which ensure reliable crack detection and evaluation.

2.1 Machine Learning and Its Application in Concrete Analysis.

Currently machine learning algorithms are widely used in concrete structure analysis these methodologies use mathematical and statistical techniques to detect patterns, analyze its relationship and to find out the meaningful information from the dataset available. In, machine learning algorithms the model is trained on the dataset available which help model to learn the pattern and perform better on new dataset.

Now a days in the field of infrastructure analysis, machine learning is a valuable tool to automate and evaluate the system for its accuracy. Researchers are exploring its use for the assessment of damage and identification of failure modes in concrete elements this machine learning models offer a reliable, data-driven alternative to traditional, manual inspection methods.

3 Methodology

In this study, we use a combined approach that integrates image processing and machine learning methodology to create a robust system which identify concrete cube failure modes. The methodology involves several key stages, as outlined below.

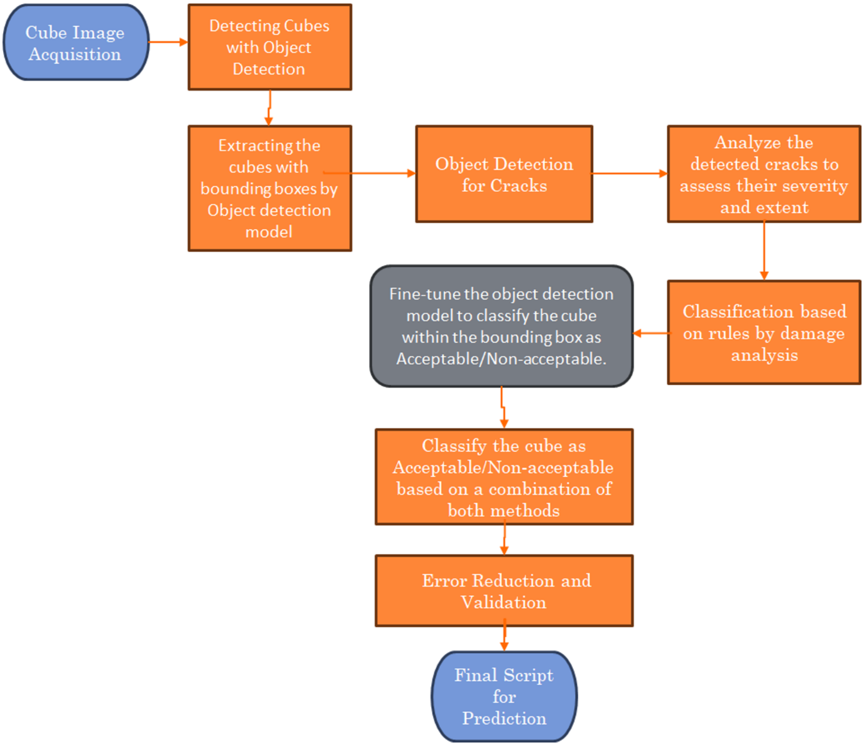

Fig. 3. Block Diagram

3.1 Dataset collection and preprocessing

We collected a dataset from several construction site for this we took help from C Probe Pvt. Ltd experts in collecting the dataset. Total about 500 concrete cube images collected and additional 1000 images were generated with data augmentation to improve the reliability. Each cube dataset consists of 6 images of each face Top, Bottom, and 4 sides. The dataset includes cubes showing both satisfactory and non-satisfactory failure modes.

After this all these images are annoted with proper lebelling. The image annotation is done under supervision of experts from industry and the dataset contains images of both satisfactory and unsatisfactory failure mode cubes. As per opinion of industry experts most of cubes are non satisfactory failure mode but due to human error they are classified as satisfactory and accepted to be used for construction So there is some imbalance in the dataset . The non satisfactory dataset is more than the satisfactory. The labeling done manually and it is followed the BS EN 12390-3 testing standards to ensure accuracy and consistency.

3.2 Detection of Objects

Object Detection for Cube:

. We have captured the images under real time scenario from construction sites thus the images contains unnecessary background images so the first step is to extract the required object from the background . For this purpose, we used the YOLOv10 (You Only Look Once, version 10) object detection framework to locate and identify the cubes from the background in the images. This object detection algorithm is fast and accurate in detecting the cube regions, which are then extracted for further processing [3]. The YOLOv10-based cube detection was carried out using the following steps.

a) Dataset Preparation:

Concrete cube images were prepared by locating bounding boxes that accurately enclose each cube specimen within the image.

b) Annotation:

To generate reliable ground truth data, the images were manually labeled using the LabelImg annotation tool. Bounding boxes were drawn around individual cubes, and each box was assigned the class label “cube”.

c) Dataset **Splitting:**

The labelled dataset was divided into training and testing subsets. This separation allows the model to be trained on known data and evaluated on unseen samples to assess its detection performance.

The YOLO10 model trained using the annotated dataset to accurately predict bounding boxes corresponding to concrete cubes. The bounding box detection mechanism in YOLO follows a structured prediction process, which includes the following stages:

Grid Formation:

Each concrete cube each face image is divided into an $N \times N$grid. A grid cell is responsible for predicting an object only when the center of the object lies within that cell.

Prediction of Bounding Box:
The bounding box is predicted along with confidence scores, indicating the likelihood that an object is present within the predicted box.

Class Probability Estimation:
Conditional class probabilities are calculated for each grid cell based on the presence of an object, enabling classification of the detected region.

In the YOLO framework, including YOLOv8, bounding boxes are defined using mathematical formulations that estimate the center coordinates (x, y), width (w), and height $(h)$of each detected object (Wang et al., 2024). The parameters used in these equations are defined as follows:

bx, by : Predicted center coordinates of the bounding box
b_w, b_h: Width and height of the predicted bounding box
t_x, t_y, t_w, t_h: Model output parameters
c_x, c_y: Coordinates of the corresponding grid cell
p_w, p_h: Width and height of the anchor (prior) box
σ: Sigmoid activation function
e: Exponential function

Center coordinates:

$$b_x = \sigma(t_x) + c_x \quad (1)$$
$$b_y = \sigma(t_y) + c_y \quad (2)$$

Width and height:

$$b_w = p_w \cdot e^{tw} \quad (3)$$
$$b_h = p_h \cdot e^{th} \quad (4)$$

The images taken from construction site contain unwanted background and other images and it is required to extract the desired cube image from the unwanted background. For this purpose, we have used YOLOv10 (You Only Look Once, version 10) for object detection to detect and accurately classify the concrete cubes within the images. This approach allows for rapid and accurate identification of the cube regions, which are then extracted for further processing (Savio et al., 2023). There are some basic steps involved in using YOLOv8 for cube detection.

a. Dataset Preparation: The input data, in this case, the concrete cube images, were annotated with bounding boxes around each cube.

b. Annotation: We need to label our images to create a ground truth for the model to learn from. Here we have used a tool Labellmg to draw bounding boxes around each cube in our images and assigned the label "cube".

c. Dataset Split: In this step, we have divided the annotated dataset into the training and testing part. This helps us to evaluate the model's performance.

The YOLOv8 model was trained on the annotated dataset to accurately predict bounding boxes around each cube. The process of bounding box recognition in YOLO involves the following steps:

- Grid Creation: YOLO model divides each image into an NxN grid and if an object's center falls within that grid cell, then the model predicts an object otherwise not.
- Bounding Box Prediction: Each grid cell gives the confidence score for each boxes indicating the probability of an object falls in that bounding box B
- Class Probability Prediction: Each grid cell gives conditional class probabilities for each object class and these probabilities are dependent on the presence of an object in the box.

In the YOLO (You Only Look Once) algorithm, including YOLOv8, bounding boxes are predicted using specific equations to determine the center coordinates (x, y), width (w), and height (h) of each bounding box. The equations typically involve the following variables: (Haoguang Wang et al.,2024)

bx, by: Center coordinates of the bounding box of predicted image

bw, bh: predicted bounding box

tx, ty, tw, th: Predicted parameters output by the model.

cx, cy: Center coordinates of the grid cell.

pw, ph : Prior (anchor) box width and height.

σ: Sigmoid function

e: Exponential function

Center Coordinates:

$$Bx = \sigma\,(tx)+cx \tag{1}$$

$$by= \sigma\,(ty)+cy \tag{2}$$

Width and Height:

$$bw=pw\cdot e^{\,tw} \tag{3}$$

$$bh=ph\cdot e^{\,th} \tag{4}$$

In these equation, the sigmoid function indicates that the center coordinates of predicted bounding box is within the bounds of the required grid cell and the exponential function is used to perform the flexible scaling of bounding box as per dimensions.With this analysis we can precisely predict the bounding box around the detected concrete cube.

Detection of cracks using object detection:

The model is fine-tuned to identify the crack regions and check their severity and spatial distribution. After the crack detection we can extract the important features such as crack count, length, width, and the damaged area.

With the help of these crack extracted information the further analysis is used to classify the cube as either acceptable or non-acceptable. Another criteria used further that ,a cube was considered non-acceptable if cracks or visible damage appeared on the top or bottom faces. On the other hand, if the top and bottom faces do not contain the cracks or damage and the four remaining faces showed a uniform crack pattern, and position the it will be classified as acceptable. And in all other situations, the cube was categorized as non-acceptable. This process required careful assessment of both the location and distribution of cracks across all faces.

To calculate the crack position and pattern on four faces of cube in crack detection process, we used the Intersection over Union method. IoU calculates the exact overlap between the bounding box of predicted and ground truth bounding box. It is calculated as the ratio of the intersecting area (where the predicted and actual boxes overlap) to the total area covered by both boxes. This metric provides a quantitative assessment of detection performance and is commonly used in object detection tasks.

Then the IoU is Calculated as

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \tag{5}$$

If IoU = 1, there is perfect overlap between the predicted and ground truth boxes.
If IoU = 0 then No overlap at all.
If IoU > 0.5 then it is generally considered a good prediction

An IoU value of 1 indicates a perfect match between the predicted and ground-truth bounding boxes, whereas an IoU value of 0 signifies no overlap between them. In practice, predictions with an IoU greater than 0.5 are generally considered acceptable and indicative of reliable object localization.

This metric provides a quantitative measure of how accurately a predicted bounding box aligns with the corresponding ground truth. In commonly used evaluation measures such as Mean Average Precision (mAP), a detection is regarded as correct when its IoU exceeds a predefined threshold, typically set at 0.5.

4 Result and Discussion

Figure 4 represents the performance metrics and loss function of object detection of concrete cube.

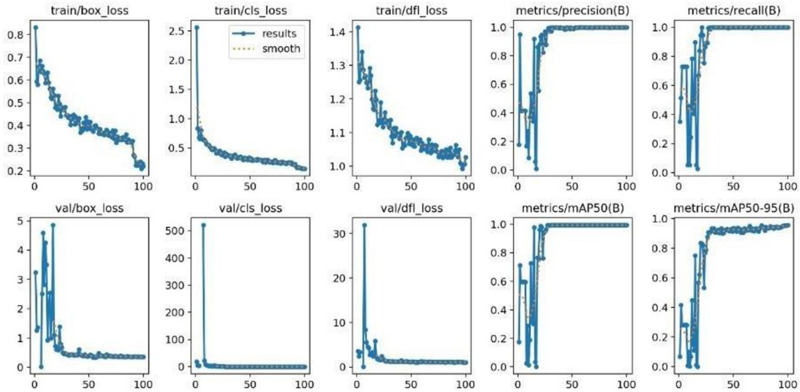

Fig. 4. Block Diagram

4.1 Losses in training period:

Top left box shows the train/box_loss metrix:
It shows that at start, the loss is relatively high at around 0.8, and then it slowly decreases to approximately 0.2 by the 100th epoch. This indicates that the model is improving in locating the bounding box.

Top- second from left shows train/cls_loss:
This starts at a high value of nearly 2.5 and steadily drops to about 0.1 by the 100th epoch. This indicates that the model improving its performance

Top Center shows train/dfl_loss:
This metrics begins at approximately 1.4 and decreases to nearly 0.8 by the end of training. This shows that the model is consistently learning.

Metrics:

metrics/precision(B) (Top-second from right):
This plot shows the model's precision during training. Precision starts at a relatively low level but rises quickly, eventually stabilizing near 1.0. It is observed that the model makes highly accurate predictions in positivev, with very few false positives.

metrics/recall(B) (Top-right):
The recall graph illustrates how well the model identifies relevant instances throughout training. Like precision, recall starts low but increases steadily, eventually reaching around 0.9. This demonstrates that the model is able to correctly detect most of the actual positive cases.

Validation Losses:

val/box_loss (Bottom-Left):
This plot shows the validation bounding box loss. Compared to the training loss, it exhibits more fluctuations but generally trends downward from about 5 to 1. This indicates that the model is learning to generalize well, despite some variability in the validation data.

val/cls_loss (Bottom-Second from Left):
The graph shows the rapid decrease which indicates that there is a significant improvement in the model's ability to correctly classify.

val/dfl_loss (Bottom-Center):
The given graph shows that there is improvement in the model performance for the given loss metric.

Overall Interpretation: The training and validation losses—box_loss, cls_loss, and dfl_loss— indicates that the model is learning effectively and it is also using that correctly to the new dataset. Precision,recall and MAP shous that mobel is capable in making accurate predictions.Precision and mAP50 reach values close to 1.0, while recall stabilizes around 0.9, demonstrating high accuracy and reliability.

4.2 Classification of cube and backgroud

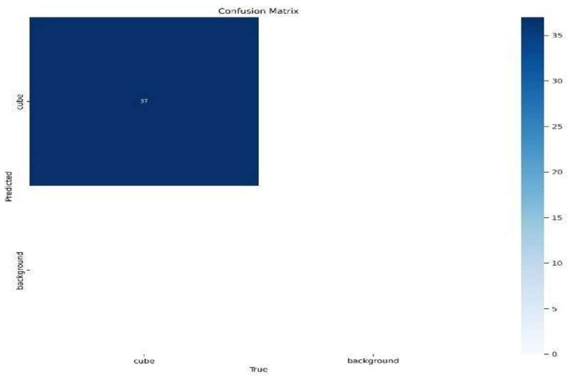

Fig. 5. The confusion matrix analysis

Confusion Matrix Analysis
The confusion matrix illustrates the performance of the classification model in distinguishing between two classes: "cube" and "background."

1. True Positives (TP): The model correctly classified 37 instances as "cube," corresponding to the true "cube" class (top-left cell of the matrix).

2. False Positives (FP): There were no instances where the model incorrectly predicted "cube" for actual "background" samples (bottom-left cell = 0).

3. False Negatives (FN): The model did not miss any "cube" instances; no "cube" samples were incorrectly classified as "background" (top-right cell = 0).

4. True Negatives (TN): There were no correctly classified "background" instances (bottom-right cell empty), indicating either the absence of "background" samples in the test data or the model did not identify any.

Overall, the model perfectly identified all "cube" instances. However, the performance on the "background" class cannot be evaluated due to the lack of corresponding samples.

4.3 Cube detection results

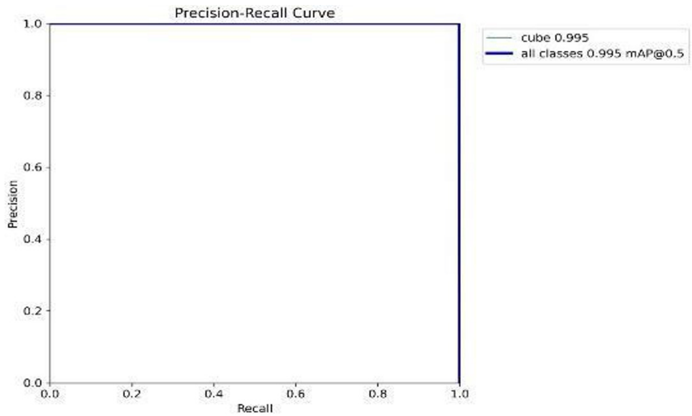

Fig. 6. Performance curve of model

The performance curve of the model gives the relationship between precision and recall for the "cube" class.

5. Precision: This is used to calculate the proportion of correct positive predictions among all instances predicted as positive, indicating how accurate the model's positive predictions are.

6. Recall: This used to measure the proportion of actual positive instances correctly calculated by the model, reflecting its ability to detect all relevant "cube" instances.

Curve Analysis:

1. The PR curve shows consistently high precision, close to 1.0, across all recall values, indicating that the model's positive predictions for the "cube" class are highly accurate.

2. The model achieves a recall of 1.0, meaning it successfully identifies all "cube" instances without missing any.

Mean Average Precision (mAP): The mAP for the "cube" class is 0.995, and the overall mAP at an IoU threshold of 0.5 is also 0.995. This demonstrates outstanding performance in detecting and classifying "cube" instances.

Overall, the Precision-Recall curve confirms that the model is highly effective in recognizing "cube" instances, achieving near-perfect precision and recall, with minimal false positives and no false negatives for this class

4.4 Metrics related to Training and validation for crack detection model

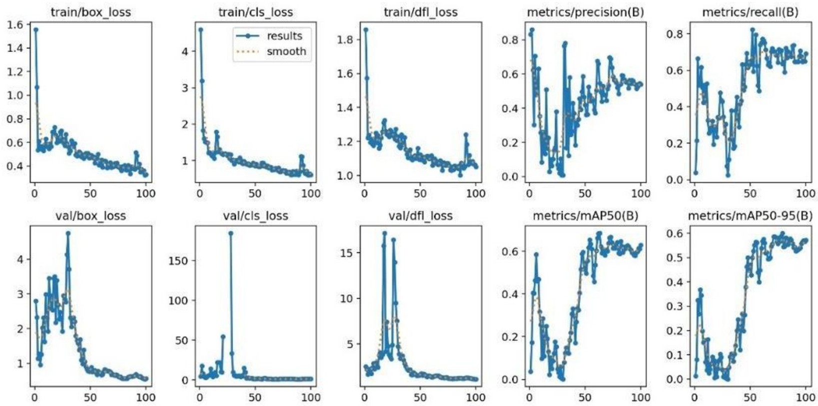

Fig. 7. Training and validation metrics

The graphs in Fig. 7 show that in detecting the crack ,the model improves consistently over time, with losses decreasing and precision, recall, and mean Average Precision (mAP) metrics increasing. The given model give better result to unseen data without overfitting because training and validation metrices shows similar pattern. The upward trends in mAP50 and mAP50-95 are especially noteworthy, demonstrating strong and reliable performance across different intersection-over-union (IoU) thresholds.

4.5 Evaluation of classifer for a crack detection model

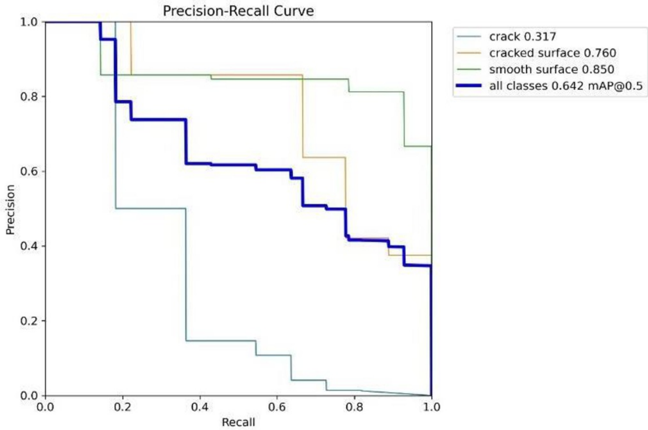

Fig. 8. Evaluation curve for a crack detection model

Overall Performance:

The model achieves an overall mAP@0.5 of 0.642 across all classes at an IoU threshold of 0.5. Performance varies by class, with the best results for smooth surfaces, followed by cracked surfaces, while detecting individual cracks is more challenging. The Precision-Recall curves show the usual trade-off between precision and recall, highlighting strong detection for smooth and cracked surfaces but indicating room for improvement in capturing fine cracks. Overall, the model performs reasonably well but can be further optimized for detailed crack detection.

5 Conclusion

The model successfully identified failure modes based on crack patterns, achieving high IoU and strong precision, recall, and F1-scores for classification. After implementing the system it is observed that the system above to detect the crack patterns and 90% accuracy in classifying the cube under acceptable and non acceptable category this reduces significantly the human error in classifying the cubes. Due to this concrete evaluation become easier and faster and also this reduces the material waste and which in turn support sustainable construction practices. The future work includes increasing the dataset in terms of the data diversification from different sites and increasing the dataset of both the category specially acceptable category for more validity and developing a mobile application is is user friendly. Overall, YOLOv10 shows strong potential for improving accuracy and efficiency in concrete cube failure analysis, supporting better quality control in construction.

6 Future Scope

Future work focus on Integrating 3D image stitching of all six faces of a cube could provide a full 360-degree view, enhancing the precision of failure mode classification. Additionally, developing a mobile or web-based application would allow for practical on-site deployment and wider use in the construction industry.In summary, the semi-automatic YOLOv10-based system has shown promising results in detecting and classifying concrete cubes as satisfactory or non-satisfactory. This approach not only increases the efficiency and reliability of assessments but also supports better quality control and promotes more sustainable, cost-effective construction practices.

Acknowledgments. This study was validated by C Probe technology pvt. Ltd. For validation of the results.

References

1. **S. Rajesh, K. S. Jinesh Babu, M. Chengathir Selvi, M. Chellapandian**, "Automated Surface Crack Identification of Reinforced Concrete Members Using an Improved YOLOv4-Tiny-Based Crack Detection Model," *Buildings*, vol. 14, p. 3402, 2024.
2. **S. Wang, X. Cheng, Y. Li, X. Song, R. Guo, H. Zhang, Z. Liang**, "Rapid visual simulation of the progressive collapse of regular reinforced concrete frame structures based on machine learning and physics engine," *Engineering Structures*, vol. 286, p. 116129, 2023.
3. **A.Savio,A.Torres,D.Salas,M.Oliversa,G.Ibarra**,"Detection and Evaluation of Construction Cracks through Image Analysis Using Computer Vision",Applied Science,vol.13,p.9662,2023.
4. **M. S. Patil, R. Ghongade, R. V. Salunke**, "Image processing and Machine learning in Concrete Cube Crack detection", International Conference on New Frontiers in Communication,Automation,Management and Security,pp1-5,2023.
5. **H.Shafei,A.radzi,M.Alahtanv,R.Rahman**,"Construction 4.0 Technologies and Decision making: A systemative Review and Gap Analysis",Building Vol12,p.2206,2022.
6. **J.Ding,N.Xue,G.Xia,X.Bai,W.Yang,M.Yang,S.Belongie,J.Luo,M.Datcu,M.Pelillo,L.Zhang**,"Object Detection in Aerial images: A Large Scale Benchmark and Challenges,"IEEEComputer Society,vol.44,pp 7778-7796,2022.
7. **J. Kim, Y. Cho, J. Lee, Y. H. Kim**, "Defect Detection and Characterization in Concrete Based on FEM and Ultrasonic Techniques," *Materials*, vol. 15, p. 8160, 2022.
8. **M. Girma, B. A. Demiss**, "Fresh, Mechanical, and Microstructural Properties Investigation on the Combined Effect of Biomedical Waste Incinerator Ash and Bagasse Ash for High-Strength Concrete," *Advances in Material Science and Engineering*, vol. 2022, pp. 1–15.
9. **A. Sadouskaya, S. Leanovich, N. Budrevich, E. Polonina**, "The compressive strength of steel fibre reinforced concrete obtained by testing cubes and cylinders," *EDP Sciences*, vol. 350, p. 00008,.
10. **E.Shahrokhinasab,N.Hosseinzadeh,A.Monirabbasi,S.Torkaman,**"Performance of Image based crack detection system in concrete structures",Journal of soft computing in civil Engineering,vol.4,pp 127-139,2020.

11. **T.Mukhopadhyay,S.Naskar,S.Dey,A.Chakrabarti**,"Condition Assessment and strengthening of aged atructures: Perspectives based on a critical case study", Practice periodical of structural design and construction,vol.24,2019.
12. **A.Downey,A.D'Alessandro,F.Ubertini,S.Laflamme**,"Automated crack detection in conductive smart concrete structures using a resistor mesh model",Measurement science and technology,vol.29,p035107,2018