

GARBAGE DETECTION USING CCTV

THANUSHA P S - CB.SC.U4AIE24154

VISHNU VARDAN - CB.SC.U4AIE24159

TEAM 6

CONTENT

- 1) OBJECTIVES*
- 2) METHODOLOGY*
- 3) RESULTS*
- 4) TIMELINE*

Objectives

*Comprehensive solution for detecting garbage
overflow using advanced computer vision
techniques (YOLOv8)*

Methodology

STEP 1:

- Collecting the Dataset from multiple sources such as roboflow, kaggle.
- The dataset contains 1764 images which can be annotated to various categories as garbage_overflow, open, open_overflow, close_overflow, close.

Methodology

STEP 2:

- The dataset which has been collected is preprocessed with the help of SVD (Singular Value Decomposition).
- The preprocessing technique is used for Noise reduction, compression and Feature Extraction.

RESULT (SVD)



Fig: Image



Fig: Preprocessed Image

Methodology

STEP 3:

- The dataset is preprocessed, after which we should format the dataset into YOLOv8
- Annotate the images
 - six classes are used for annotating
 - Open_overflow
 - open
 - close
 - close_overflow
 - damage
 - garbage_overflow

Methodology

- The next step is distributing images as follows:
 - Train (1233 images)
 - Test (170 images)
 - Valid (344 images)
- After formatting the dataset, the training process for detecting takes place.

Methodology

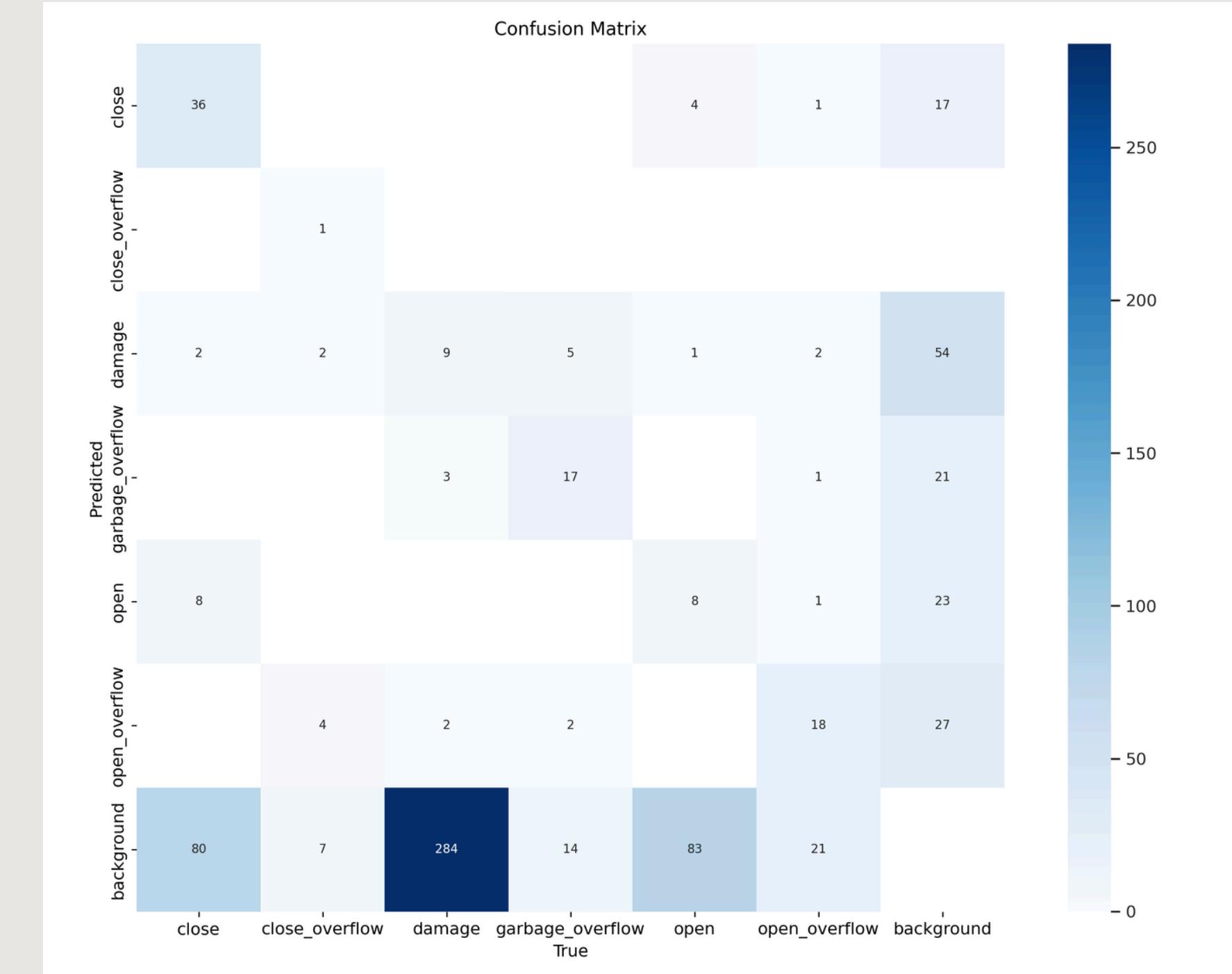


Fig: Confusion Matrix

Methodology

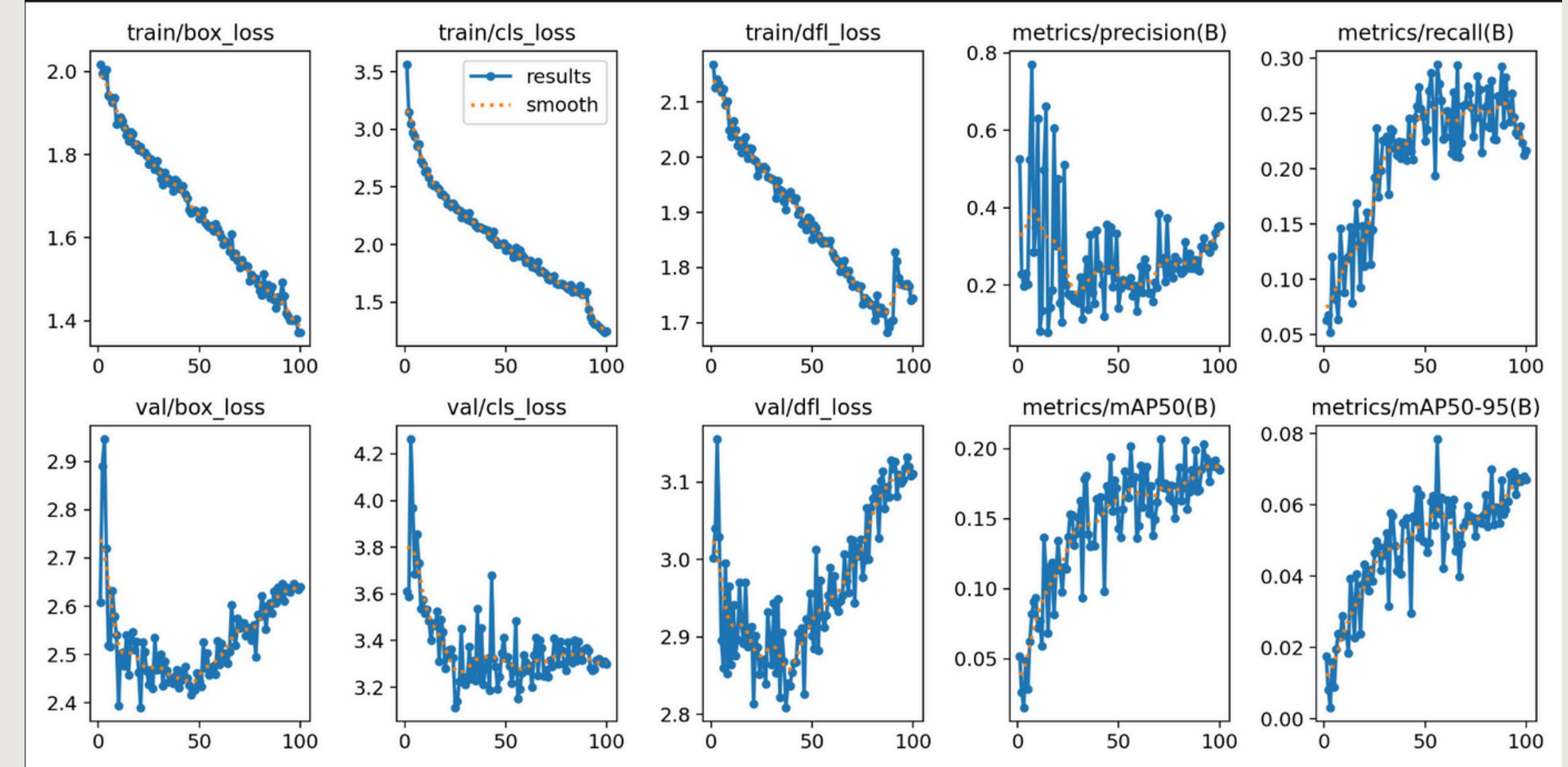


Fig: result



Fig: Detected images

Methodology

- Flask + OpenCV is used for uploading a video and to detect the video.
- Flask is a lightweight Python web framework used to create a web application.
- OpenCV is a powerful computer vision library used for image and video processing.
- Flask handles user input (file upload or webcam selection).
- OpenCV captures video frames and processes them using YOLOv8.
- Flask streams the processed frames to the browser as a live video.
- Flask allows stopping the detection, preventing resource wastage.

Result

- 1) We extract features to focus on what matters, reduce noise to improve clarity, and compress images to save space and make processing faster.
- 2) The inference we get from the training is

- mAP: 30.9%
- Precision: 36.2%
- Recall: 38.0%

Issue:

A value of 30.9% suggests that the model's performance is relatively low.

A low precision indicates that the model is making many false detections.

- 3) The trained model couldn't detect some images and wrongly detected due to mAP
- 4) There is an error in running the flask and OpenCV platform.

TIME LINE

WEEK 1:

Study existing garbage detection techniques

Understand YOLO and SVD preprocessing techniques

WEEK 2:

Dataset preparation

Preprocessing the dataset.

WEEK3:

Annotate the images of the dataset with the help of Roboflow.

Train Yolov8 on annotated images.

Evaluate model performance.

WEEK4:

Test real-time performance with CCTV footage.

Optimize false positive/negatives through threshold adjustments.

THANK YOU