NCVPRIPG 2025

RASOI

Recognition and Segmentation of Indian Thali using AI

Objective

To develop and evaluate models that can accurately demarcate each food item and label it in an image of a typical Indian Thali.

Team Name : Cyber_Hats

Team Members:

- Dr. P. Uma Maheswari
- Vimal Raj Nagarajan
- Sowmiya Maheswaran
- Dr.S.Gandhimathi @ Usha (Mentor)

College : Velmmal College of Engineering and Technology , Madurai

GitHub Link : https://github.com/Vimal-AFK/rasoi_2025

Preprocessing

Dataset Preparation :

Annotated raw images using Roboflow's platform

1	**Total Images**	**270**
2	Train Images	216
3	Val Images	54
4	Validation	40

Dataset File Structure

Model name : YOLOv8S

No. of Classes : 83

For this project, a total of 270 images of various Indian thali food items were annotated using Roboflow, a popular online tool for image labeling. Each object in the image was labeled using bounding box annotations with 4 coordinates (x, y, width, height). After annotation, the dataset was exported in YOLOv8-compatible format, which includes a data.yaml configuration file and label text files for each image. This format was directly used for training the YOLOv8 model.

Augmentation	**Function**
hsv_h: 0.015	Changes the color tone slightly.
hsv_s: 0.7	Adjusts color intensity randomly.
hsv_v: 0.4	Changes the image brightness.
degrees: 15.0	Rotates the image up to 15°.
translate: 0.1	Shifts the image left/right/up/down.
scale: 0.5	Randomly zooms in or out.
shear: 1.0	Tilts the image slightly.
perspective: 0.0003	Adds 3D-like distortion.
flipud: 0.0	Doesn't flip the image vertically.
fliplr: 0.7	Flips the image left-to-right.
mosaic: 1.0	Combines 4 images into 1.
mixup: 0.2	Blends 2 images together.
copy_paste: 0.2	Pastes one object onto another image.
erasing: 0.1	Randomly hides parts of the image.

We used data augmentation to make the model more robust and accurate, especially since the dataset had only 270 images. Techniques like color changes (hsv_h, hsv_s, hsv_v) helped the model handle lighting variations. Geometric transformations (degrees, translate, scale, shear, perspective) trained it to recognize food items from different angles and positions. Advanced methods like mosaic, mixup, and copy_paste created more complex training samples, and erasing taught the model to detect objects even when parts are missing. These augmentations improved the model's ability to generalize to new, unseen data.

Model

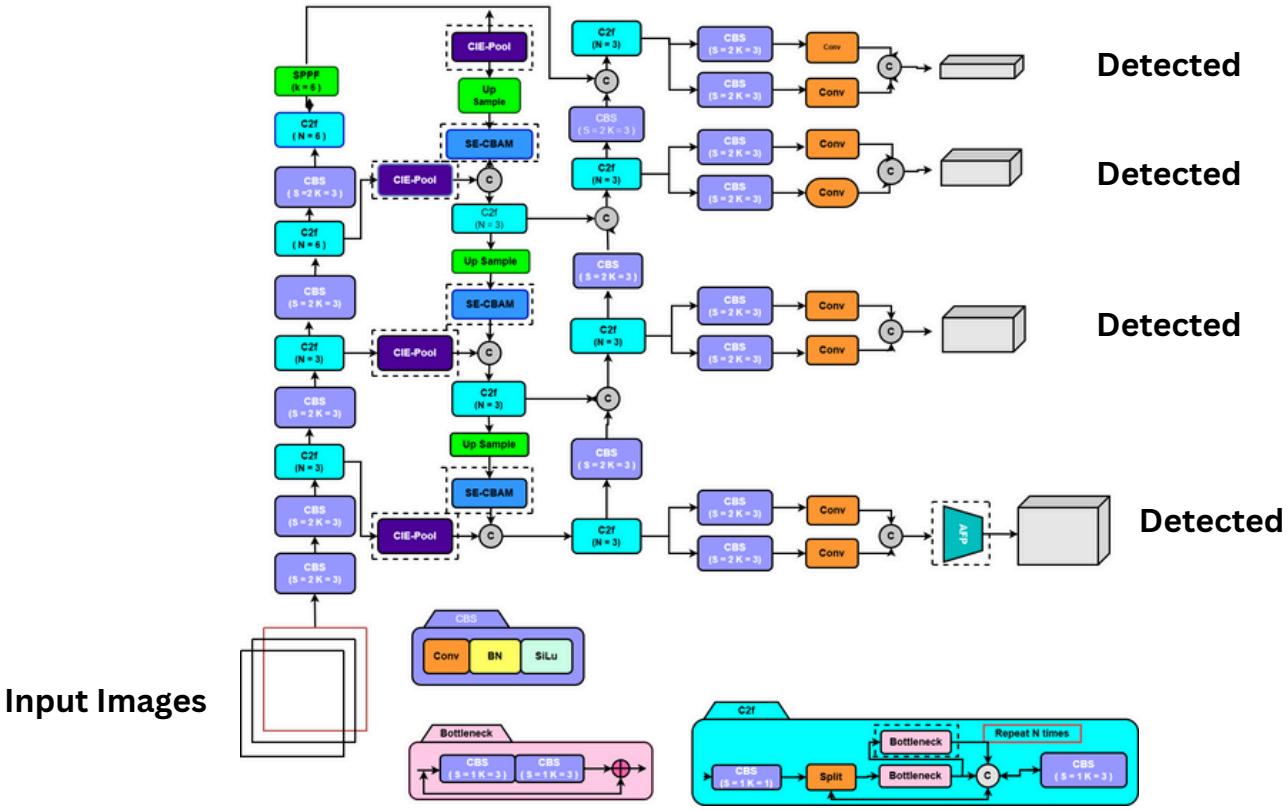
Base model: yolov8s.pt

Framework: Ultralytics YOLOv8

Model size: ~130MB

Used GPUfor training (device= 0) too toooo

Model Architectual



Training Parameters

Parameter	Value
epochs	100
batch	8
lr0	0.005
lrf	0.1
momentum	0.937
weight_decay	0.001
warmup_epochs	5
warmup_momentum	0.8
close_mosaic	15
single_cls	FALSE

These training parameters were chosen to ensure stable and effective learning. A moderate learning rate with warmup (lr0=0.005, warmup_epochs=5) helps the model start training gradually. momentum and weight_decay were tuned to balance learning speed and avoid overfitting. A small batch size of 8 suited the limited dataset size. close_mosaic=15 was used to stop complex augmentation early, and single_cls=False enabled multi-class detection, allowing the model to distinguish between different Indian food items.

Validation

Metric	Value	Meaning
Precision	0.6878	Around 69% of the predicted boxes were correct.
Recall	0.7157	Model found around 72% of all actual objects.
F1 Score	0.7015	Balance between precision and recall (overall performance).
mAP@0.5	0.7113	Model's accuracy in correctly locating and classifying food.

The YOLOv8s model was trained on 270 images and validated on 40. It achieved 68.7% precision, 71.5% recall, and an F1 score of 70.1%, showing balanced detection performance. The mAP@0.5 of 71.1% indicates good accuracy in locating and classifying Indian food items.

NOTE :

- Model will be avail in the google drive
- Put all your validation images inside the folder: *./validation_dataset/val_images*.
- Keep your ground truth labels in Excel format at: *./validation_dataset/val_labels.xlsx*.
- Install all Requirement packages from the *requirements.txt*
- All the paths correctly registered in the lines of 201 and 202 in inference.py
- Run the script to start evaluation. (**python inference.py**)
- The results will be saved as:
- predictions.csv – contains model predictions.
- metrics.json – contains overall and class-wise evaluation metrics.

Conclusion:

The YOLOv8s model performed well despite the small dataset, achieving over 70% accuracy across key metrics. With balanced precision, recall, and a strong mAP@0.5, the model effectively detects and classifies Indian food items, making it suitable for real-world applications with further data expansion.