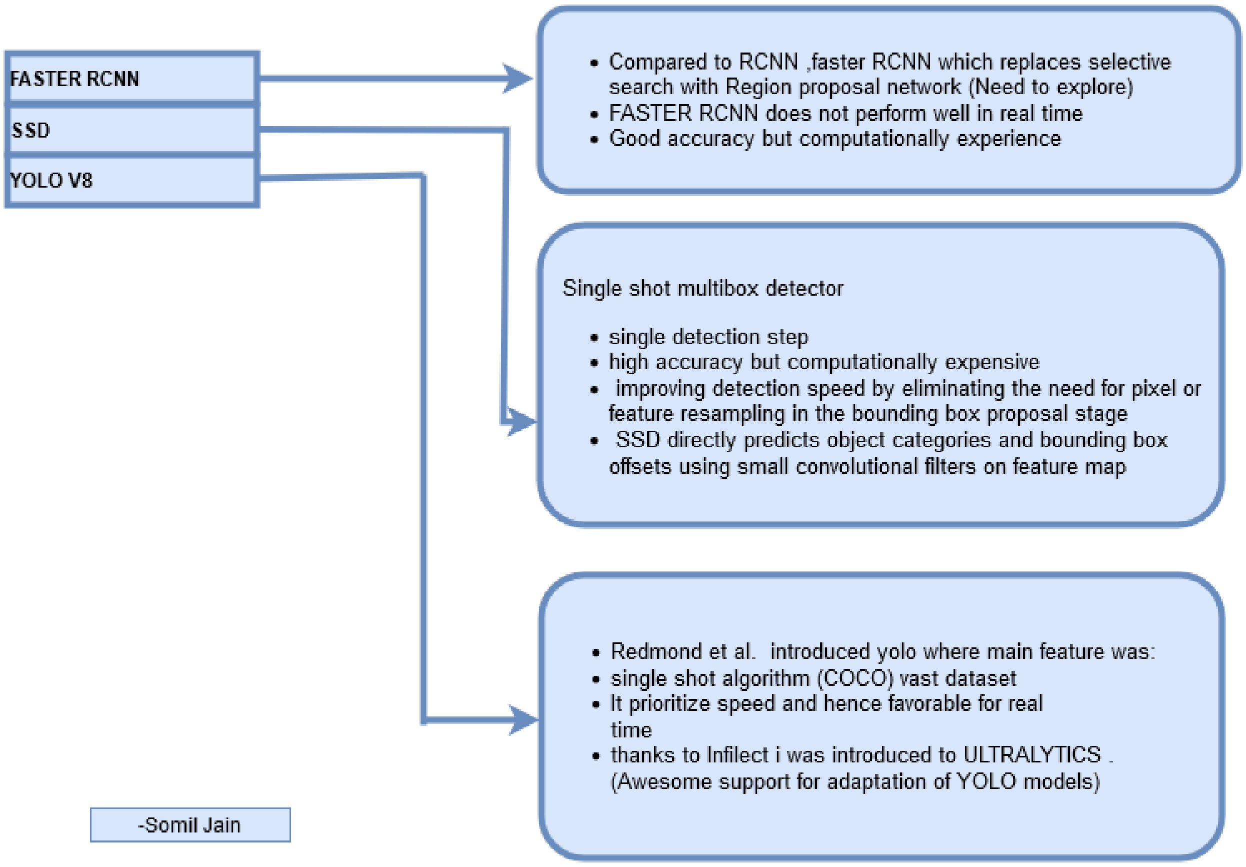
SOMIL JAIN
+919424551889

This submission is for the AI Engineer position at Infilect, where I developed an AI-based pipeline for grocery product detection, adhering closely to all the requirements outlined in the Assignment description.

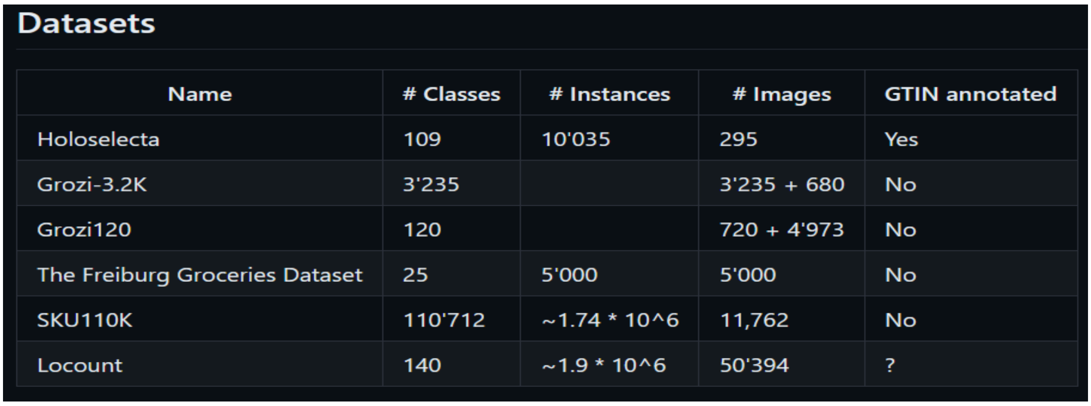


- Compared to RCNN ,faster RCNN which replaces selective search with Region proposal network (Need to explore)
- FASTER RCNN does not perform well in real time
- Good accuracy but computationally experience

Single shot multibox detector

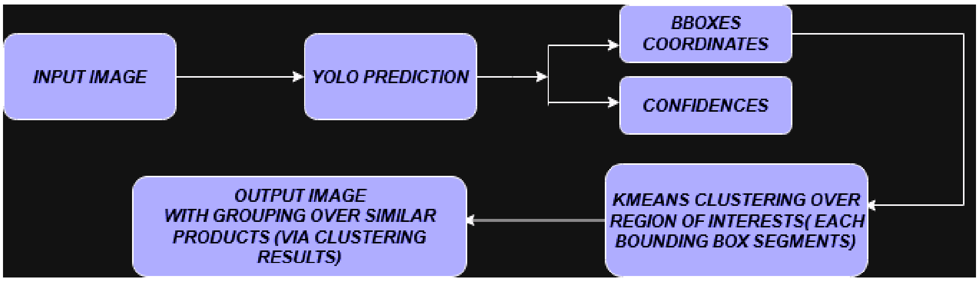
- single detection step
- high accuracy but computationally expensive
- improving detection speed by eliminating the need for pixel or feature resampling in the bounding box proposal stage
- SSD directly predicts object categories and bounding box offsets using small convolutional filters on feature map

- Redmond et al. introduced yolo where main feature was:
- single shot algorithm (COCO) vast dataset
- It prioritize speed and hence favorable for real time
- thanks to Infilect i was introduced to ULTRALYTICS . (Awesome support for adaptation of YOLO models)



Datasets

Name	# Classes	# Instances	# Images	GTIN annotated
Holoselecta	109	10'035	295	Yes
Grozi-3.2K	3'235		3'235 + 680	No
Grozi120	120		720 + 4'973	No
The Freiburg Groceries Dataset	25	5'000	5'000	No
SKU110K	110'712	~1.74 * 10^6	11,762	No
Locount	140	~1.9 * 10^6	50'394	?


INPUT IMAGE
YOLO PREDICTION
BBOXES COORDINATES
CONFIDENCES
KMEANS CLUSTERING OVER REGION OF INTERESTS(EACH BOUNDING BOX SEGMENTS)
OUTPUT IMAGE WITH GROUPING OVER SIMILAR PRODUCTS (VIA CLUSTERING RESULTS)

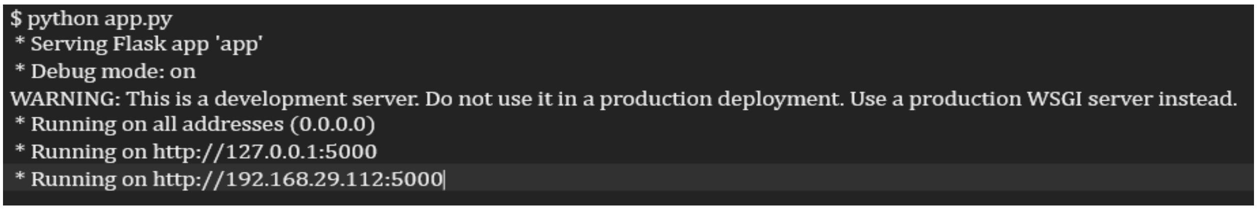
I applied k-means clustering on the regions of interest derived from bounding boxes generated by a custom YOLO model, which was pre trained to detect retail products in images. I used clustering because I couldn't successfully classify each product, so I opted to group similar products based on their regions of interest.

Project Structure

- **app.py**: The main Python file for running the Flask server.
- **exp/**: Contains the outputted results, including:
  - JSON files with detection data
  - Outputted images with bounding boxes
  - Metadata associated with the detection process
- **Image_object_detection.py**: Includes functions for performing model inference.
- **models/**: Directory containing the pretrained model files.
- **requirements.txt**: Lists all the required libraries for the project.

```
pip install -r requirements.txt
```

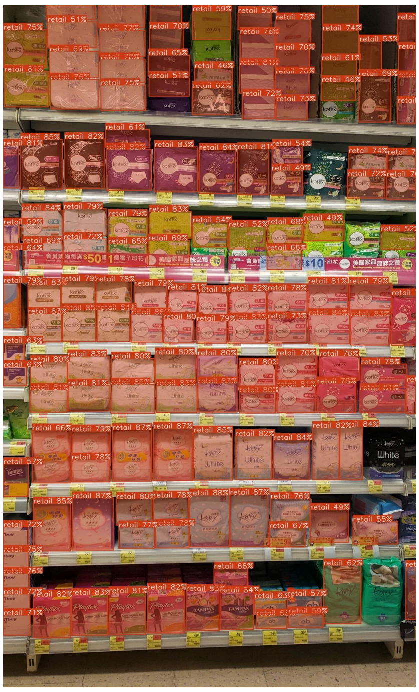
- **static/**: Contains image files used by the application.
- **templates/**: Contains HTML files for the front-end.
- **yolov8/**: Directory containing the model class and related files.



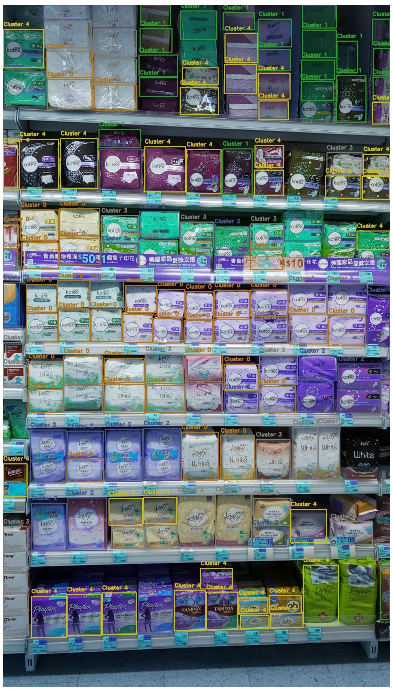
```
$ python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.29.112:5000
```

SOMIL JAIN
+919424551889

ORIGINAL IMAGE

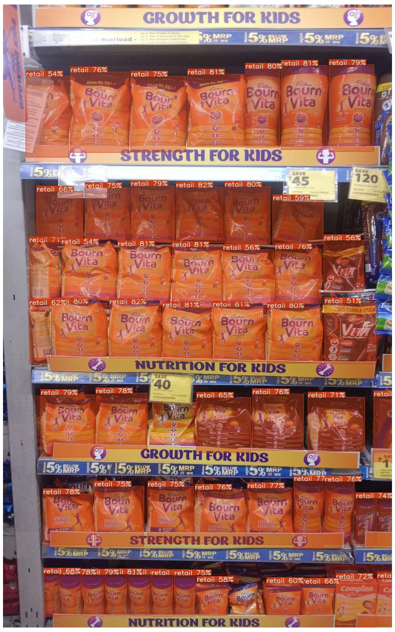


YOLO OUTPUT

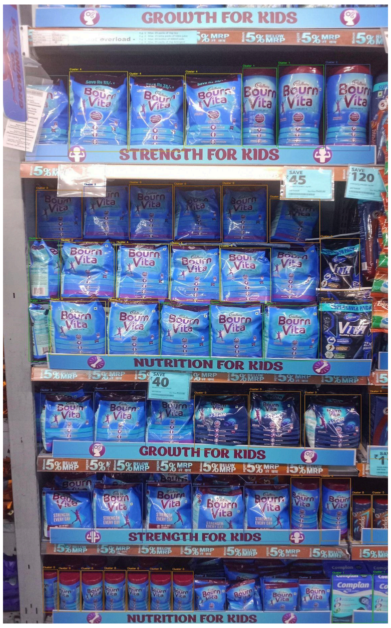


GROUPING OUTPUT

ORIGINAL IMAGE



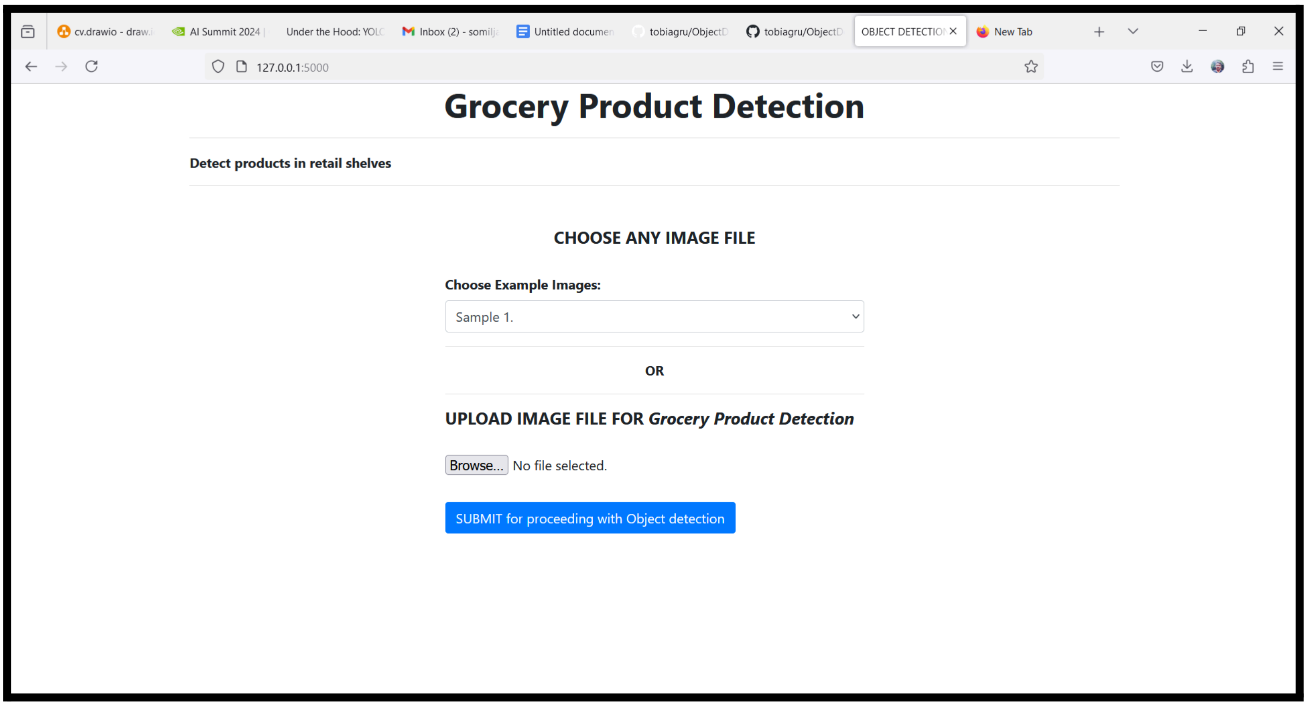
YOLO OUTPUT



GROUPING OUTPUT

SOMIL JAIN
+919424551889

SCREEN SHOTS:



Grocery Product Detection

Detect products in retail shelves

CHOOSE ANY IMAGE FILE

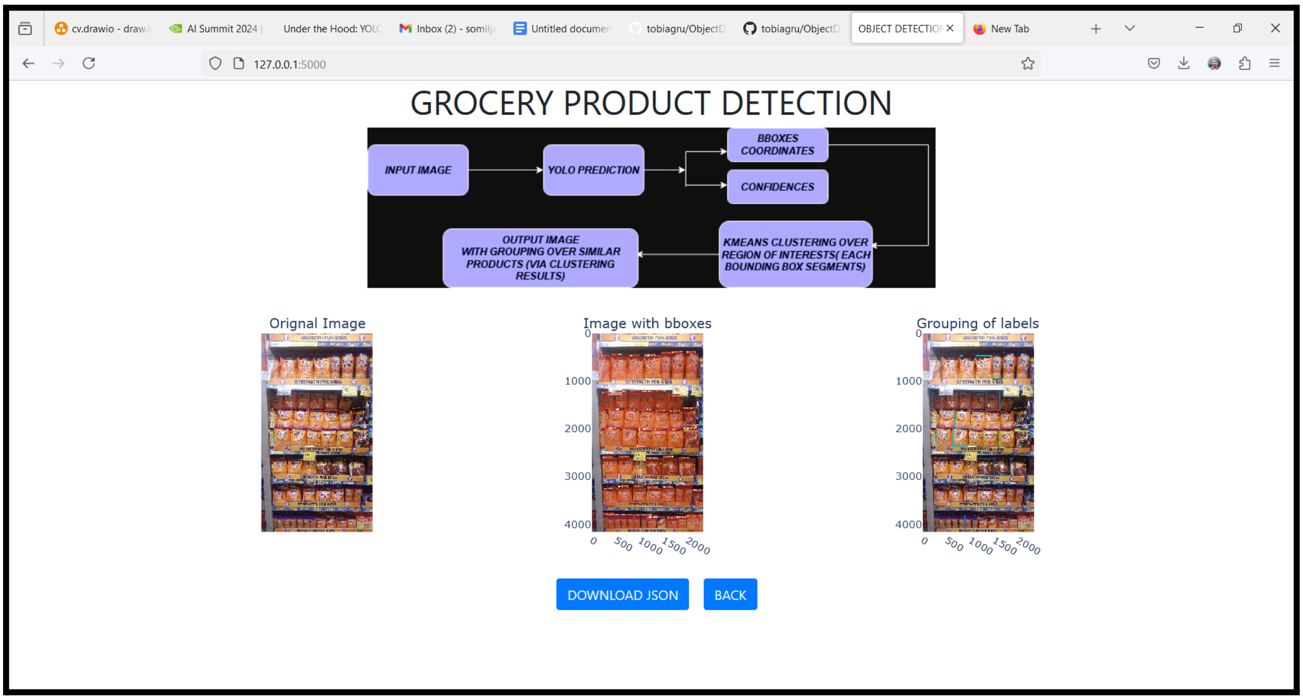
Choose Example Images:

Sample 1.

OR

UPLOAD IMAGE FILE FOR *Grocery Product Detection*

No file selected.



GROCERY PRODUCT DETECTION

Couple of things 👍

- Image input can be done via
    - i. Browsing through sample images
    - ii. Uploading a image (which gets stored to exp folder in server)
      (only takes Jpg format for processing (can be modified later))
- Json can be downloaded from results page

Future scope ➕

- Try some new object detection model (want to explore SAM type models)
- Try some feature extraction which can be used for modelling (like somehow transformer -> embedding -> clustering type solutions)

Note:
I have not worked on object detection systems but have worked on DNN for audio features, multi dimensional embeddings and followed by different types of clustering involving FAISS, AHC ,VB ,spectral clustering .