MAJOR PROJECT

GRASPING POINT CALCULATION FOR DESIRED OBJECT AMONG MULTIPLE OBJECTS

Indian Institute of Information technology, Kota

CONTENT

GOALS AND OBJECTIVES

Objective n° 1

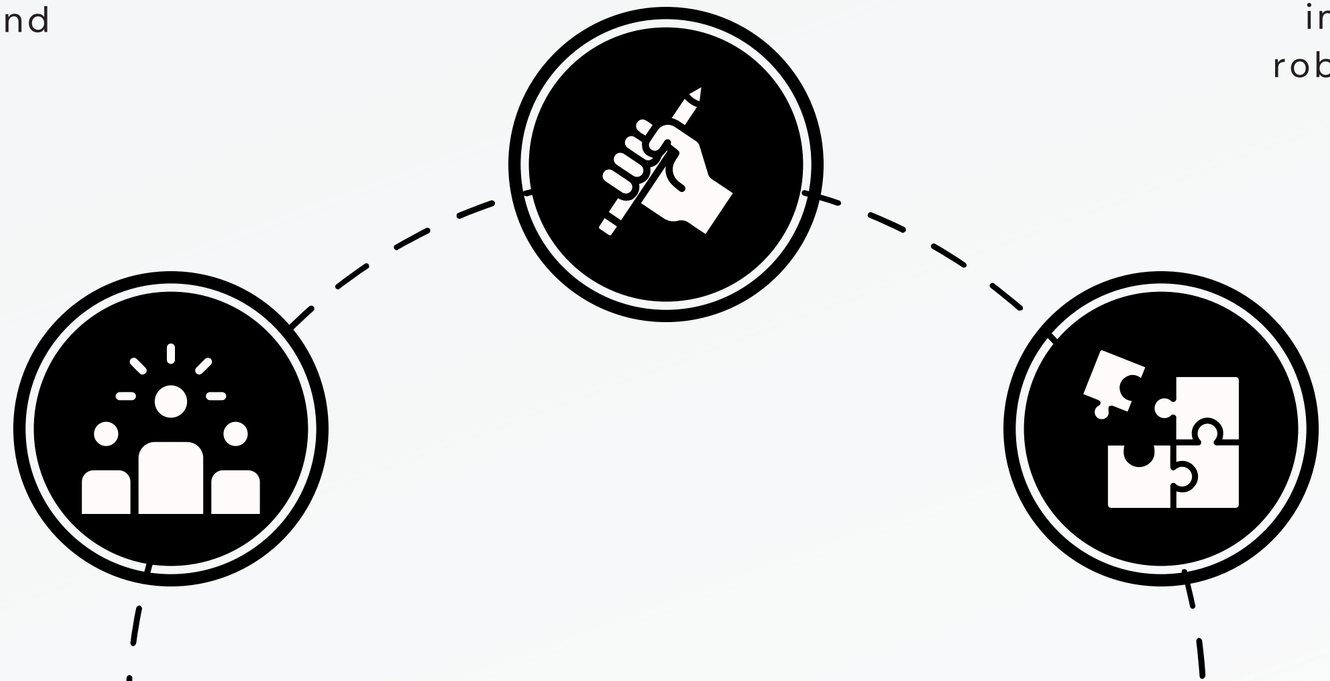
Review research papers to determine the current best algorithms for image segmentation and detection,

Objective n° 2

Compare various models for image segmentation and center object detection, evaluating accuracy and speed to identify the most effective model .

Objective n° 3

Develop a method to accurately detect grasping points in the center of segmented images, enhancing robotic manipulation



Literature Review

S. No	Title of the Paper	Authors	Inference
1.	ARMBench: An Object-centric Benchmark Dataset for Robotic Manipulation (29 march, 2023)	Chaitanya Mitash1 , Fan Wang1 , Shiyang Lu2 , Vikedo Terhuja1 , Tyler Garaas1 , Felipe Polido1 , Manikantan Nambi1	This paper presents ARMBench, containing 235K+ annotated pick-and-place activities on 190K+ unique objects, enabling research on object segmentation, identification, and defect detection.
2.	Mask R-CNN (24 January, 2018)	Kaiming He Georgia Gkioxari Piotr Dollar Ross Girshick ´ Facebook AI Research (FAIR)	The paper introduces Mask R-CNN, enhancing Faster R-CNN with a mask prediction branch for object segmentation. It employs pixel-to-pixel alignment, leading to efficient instance segmentation
3.	Object Detection Instance Segmentation on Amazon ARMBench Dataset using Mask RCNN	Pinak Jani	This article presents an implementation for training an instance segmentation model (Mask R-CNN) on Amazon's large-scale ARMBench dataset

S. No	Title of the Paper	Authors	Inference
4.	Object detection using YOLO: challenges, architectural successors, datasets and applications (8 August 2022)	Tausif Diwan1 & G. Anirudh2 & Jitendra V. Tembhurne	This paper presents a comprehensive review of single-stage object detection algorithms, focusing on YOLOs. It highlights the challenges, architectural aspects, optimization techniques, and applications of YOLO
5.	Enhancing Real-time Object Detection with YOLO Algorithm (5 December 2023)	Gudala Lavanya1 and Sagar Dhanraj Pande2,	This paper presents a comprehensive overview of the YOLO algorithm for real-time object detection using convolutional neural networks (CNN). It discusses the unified detection approach, network architecture, different versions (YOLOv2, YOLOv3, etc.)

MODEL 1(MASK R-CNN)

Object Detection and Segmentation using Mask R-CNN with ResNet50

Mask R-CNN with ResNet-50-FPN is selected for its advanced instance segmentation capabilities, combining object detection and pixel-level segmentation in a single model.

MODEL CHOICE

Leveraging pre-trained weights from COCO dataset enhances model performance on ARMBench, enabling faster convergence and better generalization.

TRANSFER LEARNING

The ARMBench dataset, tailored for robotic manipulation in warehouses, provides the foundation for training and evaluating the model.

DATASET

The model is adapted to handle three classes (background, tote, object), enabling precise segmentation and classification of warehouse items.

MULTI-CLASS SEGMENTATION

The code establishes a training pipeline using PyTorch, preparing data for training, optimizing the model with SGD, and monitoring loss during training.

TRAINING PIPELINE

MODEL 2(YOLOV8)

Object Detection and Segmentation using YOLOv8

YOLOv8 is a powerful and versatile model that excels at instance segmentation tasks. It provides a range of model scales (n, s, m, l, x) to balance speed and accuracy for different hardware requirements

MODEL CHOICE

YOLOv8 supports training on custom datasets in addition to pre-trained models trained on COCO dataset. COCO contains 200Kimages having annotations for object detection, segmentation, and captioning tasks. The dataset comprises 80 object categories

DATASET

YOLOv8 is known for its efficiency and real-time processing capabilities, making it suitable for applications that require fast inference speeds.

EFFICIENCY AND SPEED

The YOLOv8 segmentation model outputs two key components:
Bounding boxes: Identifying the location and class of objects
Segmentation masks: Pixel-wise masks outlining the shape of each detected object

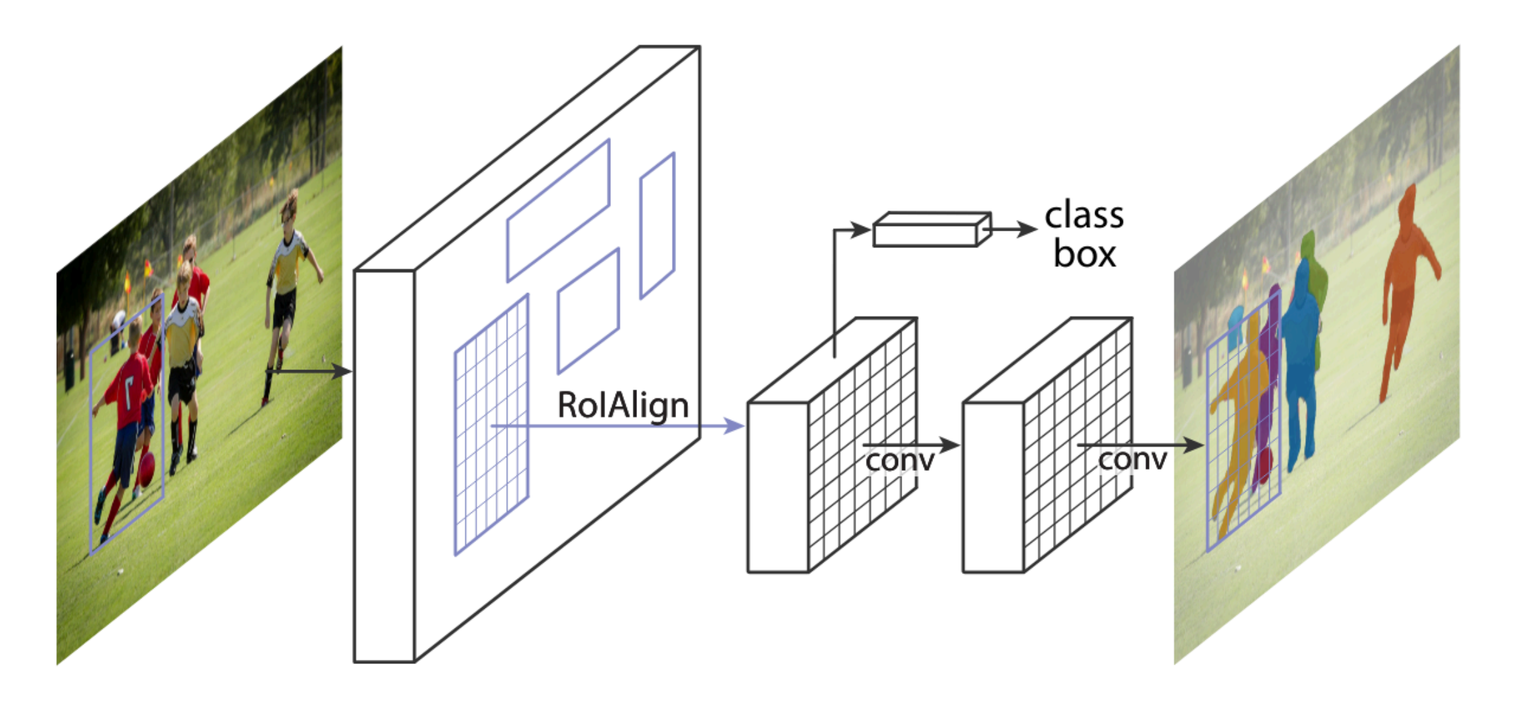
OUTPUT AND EVALUATION

The Mask R-CNN framework for instance segmentation

Backbone CNN: A convolutional neural network like ResNet-50 or ResNet-FPN is used as the backbone for feature extraction

RPN The RPN generates region proposals by classifying anchor boxes and regressing bounding box coordinates

Region-based Feature Extractor Features are extracted from each region proposal using ROIAlign, a variant of ROIPool



Classification and Bounding Box Regression Fully connected layers classify each region proposal and regress the bounding box coordinates

Mask Prediction: A small FCN (Fully Convolutional Network) predicts a segmentation mask for each class, independent of its position. These masks are then projected to the bounding box

Features	Mask RCNN	YOLO v8
Overview	two-stage instance segmentation	YOLOv8 is a one-stage instance segmentation model
Architecture	Uses RPN to generate object proposals, refined by the segmentation and bounding box branches.	Uses a single stage to predict bounding boxes directly from full images in one evaluation.
Speed	Slower than YOLOv8 due to its two-stage architecture.	Generally faster than Mask R-CNN due to its simpler architecture.
Accuracy	Comparable accuracy to YOLOv8 for instance segmentation, especially for small objects or objects with complex shapes.	Slightly higher accuracy than Mask R-CNN for object detection. and instance segmentation
Complexity	More complex architecture than YOLOv8.	Simpler architecture than Mask R-CNN, making it easier to train and deploy.

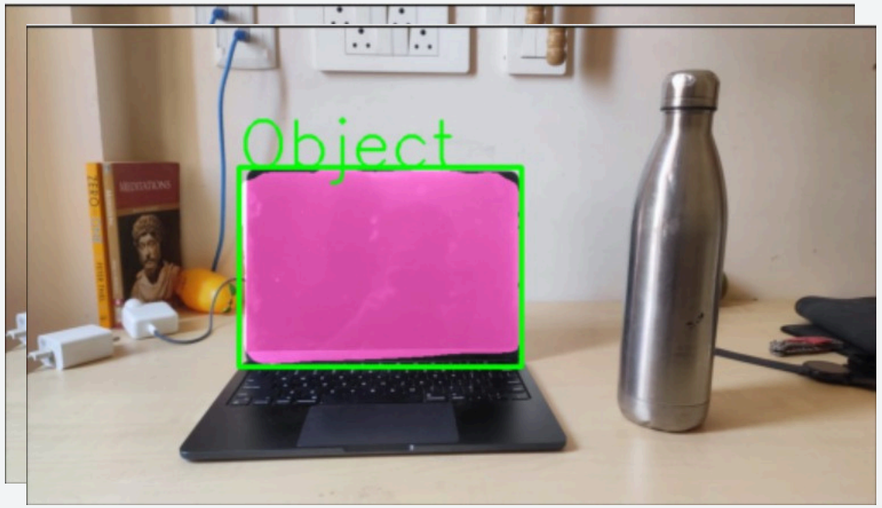
STATISTICS

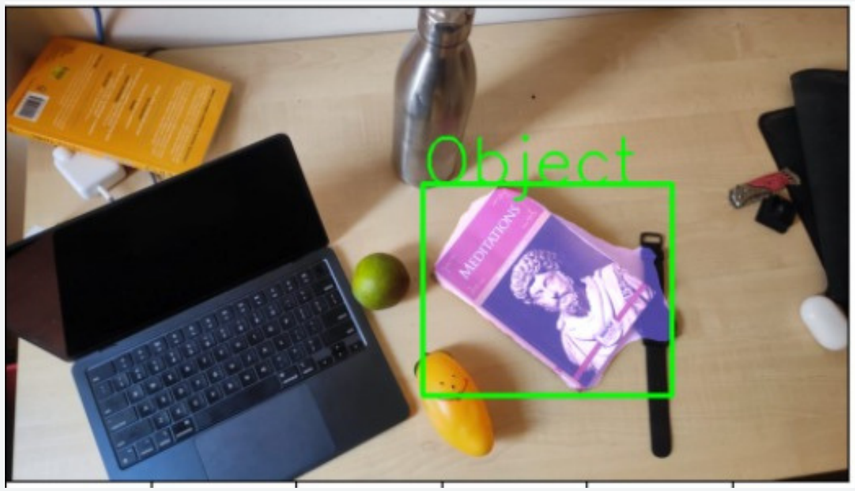
IN THE UPCOMING PAGES WE HAVE COMPARED THE RESULTS AND FINDINGS FOR BOTH MODELS

Evaluation Results

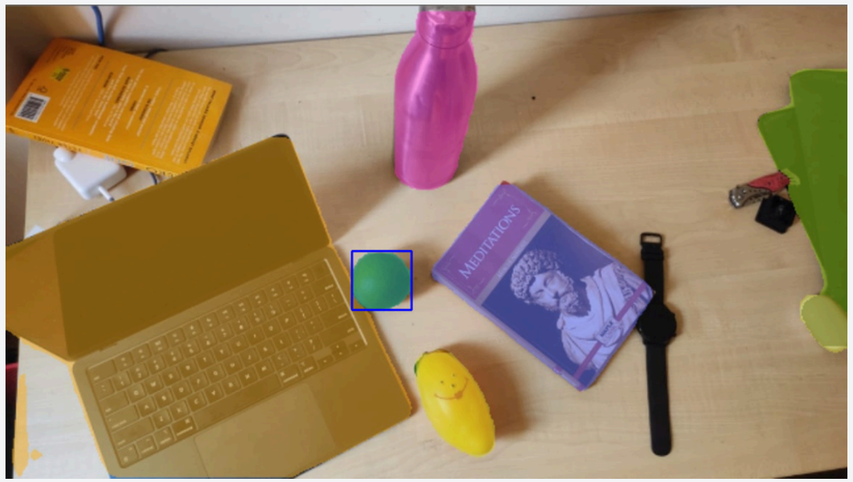
		MASK RCNN	YOLO V8
Average Precision (AP) @[IoU=0.50:0.95	area= all	**maxDets=100] = 0.419**	maxDets=100] = 0.367
Average Precision (AP) @[IoU=0.50	area= all	maxDets=100] = 0.498	**maxDets=100] = 0.522**
Average Precision (AP) @[IoU=0.75	area= all	**maxDets=100] = 0.495**	maxDets=100] = 0.398
Average Precision (AP) @[IoU=0.50:0.95	area= small	maxDets=100] = 0.000	**maxDets=100] = 0.179**
Average Precision (AP) @[IoU=0.50:0.95	area=medium	maxDets=100] = 0.004	**maxDets=100] = 0.404**
Average Precision (AP) @[IoU=0.50:0.95	area= large	maxDets=100] = 0.419	**maxDets=100] = 0.521**
Average Recall (AR) @[IoU=0.50:0.95	area= all	**maxDets= 1] = 0.445**	maxDets= 1] = 0.315
Average Recall (AR) @[IoU=0.50:0.95	area= all	maxDets= 10] = 0.448	**maxDets= 10] = 0.531**
Average Recall (AR) @[IoU=0.50:0.95	area= all	maxDets=100] = 0.455	**maxDets=100] = 0.586**
Average Recall (AR) @[IoU=0.50:0.95	area= small	maxDets=100] = 0.000	**maxDets=100] = 0.366**
Average Recall (AR) @[IoU=0.50:0.95	area=medium	maxDets=100] = 0.000	**maxDets=100] = 0.650**
Average Recall (AR) @[IoU=0.50:0.95	area= large	maxDets=100] = 0.460	**maxDets=100] = 0.767**
mAP		0.41916014	**0.46716667**

MASK RCNN

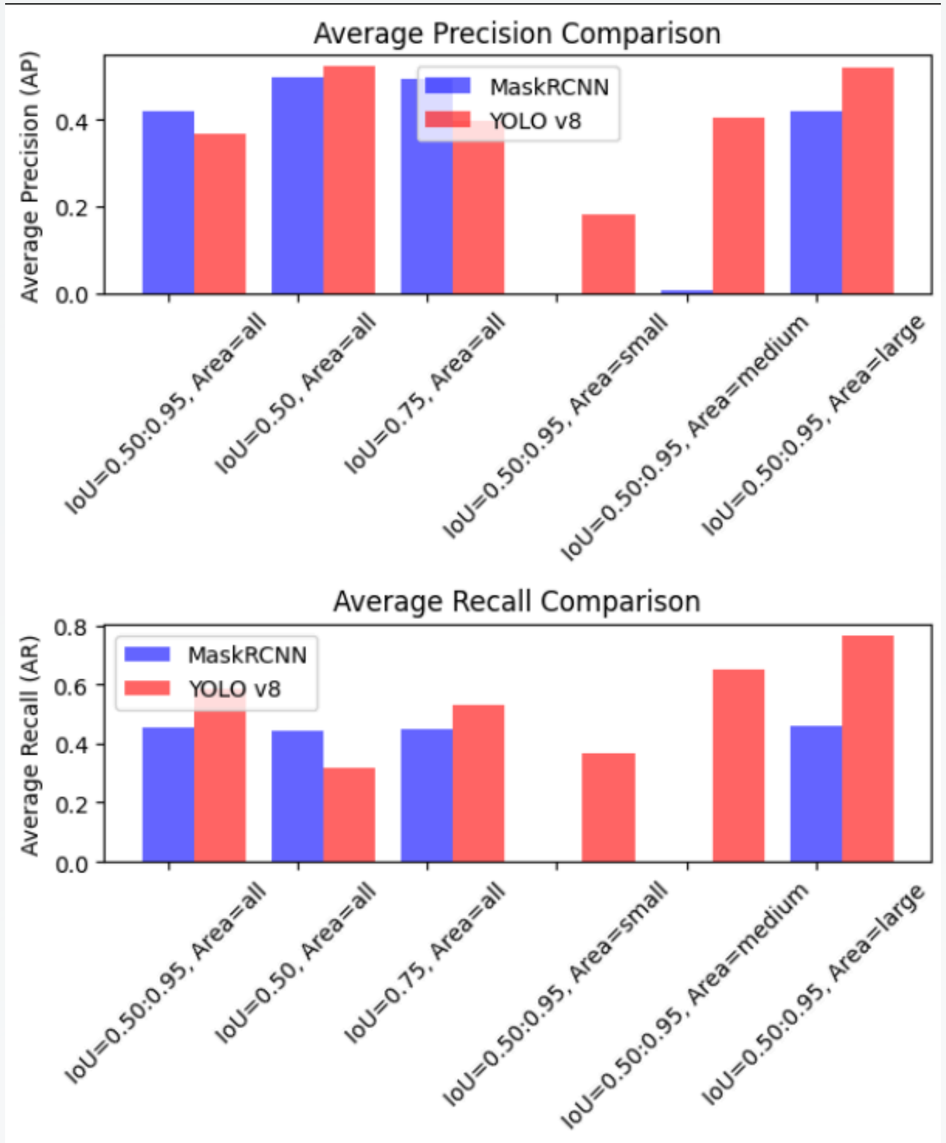


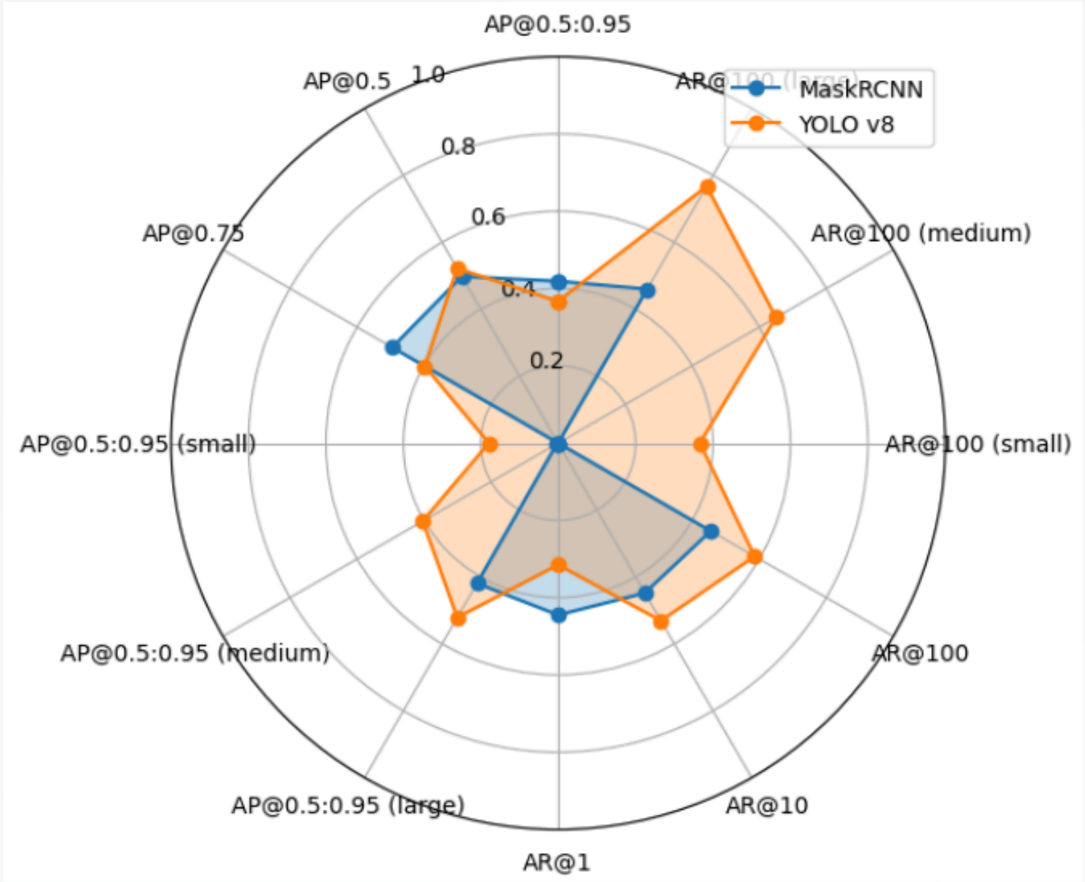


YOLO v8



Comparative Analysis





LONG TERM VISION

- Optimal balance between speed and accuracy.
- Suitable for real-time applications.
- Easier integration and maintenance.

USING YOLOV8 FOR NEXT PHASE

Leverage YOLOv8's bounding box outputs to identify object locations. Apply a grasping point calculation algorithm to these locations.

INTEGRATION WITH GRASPING ALGORITHM

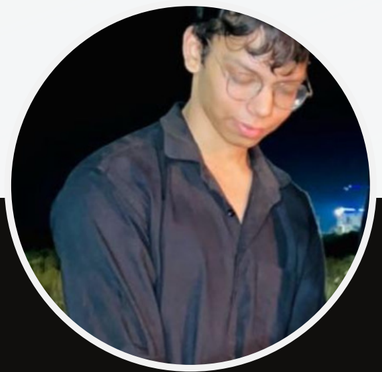
Efficient and accurate detection of grasping points.
Improved performance in real-time applications.
Enhanced reliability in diverse object environments.

EXPECTED OUTCOMES

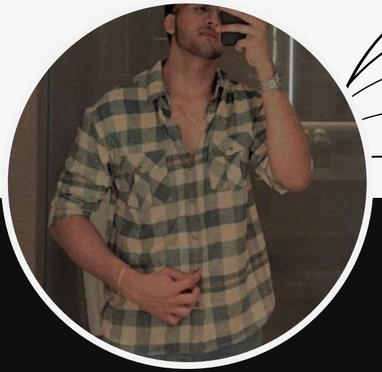
OUR TEAM

Lokesh
Kakkar
(2021kucp1102)

Anmol
(2021kucp1090)



Lokesh
Panjiar
(2021kucp1088)



Aryan
Pandey
(2021kucp1083)

THANK YOU