Grasping Point Calculation for Desired Object among Multiple Objects

B. Tech Project by

[Aryan Pandey , Lokesh Kakkar , Anmol , Lokesh Panjiar]

Under the guidance of

Dr. Amit Kumar

In Partial Fulfilment of the Requirements for the

Degree of Bachelor of Technology

Computer Science and Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

KOTA

21 MAY 2024

ACKNOWLEDGEMENTS

We like to share our sincere gratitude to all those who help us in completing the first phase of our project. During the work we faced many challenges due to our lack of knowledge and experience but many people helped us to get over all the difficulties and in final compilation of our idea to a shaped sculpture. We would like to thank our project coordinator Dr. Amit Kumar (Assistant Professor) for his governance, guidance, his continuous help and monitoring during the project work.

We would also like to thank the CSE Department of IIIT Kota for providing us such an opportunity to learn from these experiences.

Thank you

Aryan Pandey [2021KUCP1083]

Lokesh Kakkar [2021KUCP1102]

Anmol [2021KUCP1090]

Lokesh Panjiar [2021KUCP1088]

ABSTRACT

This report presents a cutting-edge methodology for Grasping Point Calculation for Desired Object among Multiple Objects, employing a hybrid model that combines the robust capabilities of Mask R-CNN with a ResNet50 and FPN backbone, in conjunction with YOLOv8 for object detection and segmentation. By synergizing the unique strengths of these models, the system adeptly identifies and isolates objects within a scene, facilitating the precise determination of the optimal grasping point for a specified object amidst a multitude of objects. This innovative approach not only enhances object recognition and localization but also significantly improves the accuracy of grasping point calculation, thereby propelling advancements in robotic manipulation systems operating within complex environments.

Key words: MaskRCNN, ResNet50, RFC, FPN, YOLOv8.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Robotic manipulation systems have seen significant advancements in recent years, with a growing emphasis on enhancing object recognition and manipulation capabilities. One critical aspect of such systems is the ability to accurately calculate the grasping point for a desired object among multiple objects in a given scene. This report introduces a novel approach that leverages a hybrid model combining Mask R-CNN with a ResNet50 and FPN backbone, along with YOLOv8 for object detection and segmentation, to address this challenge.

By integrating the strengths of these state-of-the-art models, our methodology aims to improve the accuracy and efficiency of grasping point calculation in complex environments. The utilization of Mask R-CNN with a ResNet50 and FPN backbone provides precise object segmentation, while YOLOv8 enables real-time object detection of multiple objects. This combination allows for the identification and isolation of the desired object, leading to the determination of the optimal grasping point amidst various objects present in the scene.

The research presented in this report contributes to the advancement of robotic manipulation systems by offering a robust framework for efficient object manipulation. The integration of cutting-edge technologies in object detection and segmentation not only enhances object recognition and localization but also paves the way for more accurate and reliable grasping point calculation. This introduction sets the stage for the detailed exploration of our hybrid approach and its implications for improving robotic manipulation tasks in diverse and challenging environments.

Chapter 2

RELATED WORK

The field of real-time object detection has seen significant advancements, primarily due to the development of deep learning techniques and convolutional neural networks (CNNs). Early methods for object detection, such as the Viola-Jones detector, relied heavily on feature selection and traditional machine learning approaches which often struggled with accuracy and speed in complex environments.

With the advent of deep learning, the Region-based Convolutional Neural Network (R-CNN) family marked a pivotal shift. The original R-CNN, followed by Fast R-CNN, Faster R-CNN, and the Region-based Fully Convolutional Networks (R-FCN), demonstrated significant improvements in accuracy by leveraging CNNs to extract features and classify objects within proposed regions. R-CNN approaches bounding-box object detection by attending to a manageable number of candidate object regions and evaluating convolutional networks independently on each region of interest (RoI). R-CNN was extended to allow attending to RoIs on feature maps using RoIPool, leading to fast speed and better accuracy. Faster R-CNN advanced this stream by learning the attention mechanism with a Region Proposal Network (RPN). Faster R-CNN is flexible and robust to many follow-up improvements and is the current leading framework in several benchmarks .

Driven by the effectiveness of R-CNN, many approaches to instance segmentation are based on segment proposals. Earlier methods resorted to bottom-up segments. DeepMask and similar works learn to propose segment candidates, which are then classified by Fast R-CNN. These methods, where segmentation precedes recognition, are slow and less accurate. Li et al. combined the segment proposal system and object detection system for "fully convolutional instance segmentation" (FCIS). In these methods, the

common idea is to predict a set of position-sensitive output channels fully convolutionally, addressing object classes, boxes, and masks simultaneously, making the system fast. However, FCIS exhibits systematic errors on overlapping instances and creates spurious edges, showing challenges in segmenting instances. Other solutions to instance segmentation are driven by the success of semantic segmentation, aiming to cut pixels of the same category into different instances. In contrast, Mask R-CNN is based on an instance-first strategy, offering a unique approach to instance segmentation.

The introduction of the You Only Look Once (YOLO) algorithm by Joseph Redmon et al. revolutionized the field by reframing object detection as a single regression problem, predicting bounding boxes and class probabilities directly from full images in one evaluation. This approach significantly enhanced the speed of detection, enabling real-time applications while maintaining competitive accuracy. Subsequent versions of YOLO have continued to refine this balance between speed and accuracy. YOLOv2, also known as YOLO9000, introduced batch normalization, high-resolution classifiers, and multi-scale training to improve detection performance. YOLOv3 further enhanced these capabilities by incorporating feature pyramid networks to handle objects at different scales better.

Recent improvements include YOLOv4 and YOLOv5, which have optimized training strategies and incorporated advancements such as mosaic data augmentation, self-adversarial training, and cross-stage partial connections (CSP) to improve both speed and accuracy. These versions have demonstrated state-of-the-art performance on benchmarks such as the Common Objects in Context (COCO) dataset, showcasing YOLO's robustness in various real-world scenarios.

Additionally, specialized versions like Tiny-YOLO and Fast YOLO have been developed for resource-constrained environments, highlighting the adaptability of the YOLO architecture. These models sacrifice some accuracy to achieve even greater speeds and lower computational requirements, making them suitable for embedded systems and applications requiring real-time processing with limited hardware.

Overall, the continuous evolution of YOLO and its derivatives underscores the significant progress in real-time object detection, driven by innovations in neural network design and training methodologies. These advancements have broad implications for fields ranging from autonomous driving and video surveillance to medical imaging and beyond.

Chapter 3

PROBLEM STATEMENT

The primary challenge addressed in this report is the accurate calculation of grasping points for a desired object among multiple objects in a given scene. In complex environments where various objects are present, precisely determining the optimal grasping point for a specific object is crucial for efficient robotic manipulation tasks. Existing approaches often struggle with accurately identifying and isolating the desired object, leading to suboptimal grasping point calculation.

To overcome this challenge, this research proposes a hybrid model that combines the strengths of Mask R-CNN with a ResNet50 and FPN backbone, along with YOLOv8, for object detection and segmentation. By integrating these state-of-the-art models, the system aims to enhance object recognition and localization, enabling more accurate grasping point calculation for the desired object amidst multiple objects.

The key objectives of this research are:

1. To develop a robust and accurate hybrid model for object detection and segmentation, leveraging Mask R-CNN with a ResNet50 and FPN backbone, and YOLOv8.

2. To integrate the outputs of Mask R-CNN and YOLOv8 to precisely identify and isolate the

desired object within a scene containing multiple objects.

3. To calculate the optimal grasping point for the identified desired object, ensuring efficient and reliable robotic manipulation in complex environments.

4. To evaluate the performance of the proposed hybrid model and compare it with existing approaches, demonstrating its superiority in terms of accuracy, speed, and robustness.

By addressing these objectives, this research aims to contribute to the advancement of robotic manipulation systems, enabling more efficient and reliable object grasping in real-world scenarios.

Chapter 4

Datasets Used

MODEL-1 MASK RCNN

Amazon ARMBench Dataset:

The Amazon Robotic Manipulation Benchmark (ARMBench) dataset serves as a pivotal resource for advancing robotic manipulation tasks within warehouse environments. Unlike existing datasets that often focus on a limited set of objects or rely on synthetic scenes generated from 3D models, ARMBench stands out for its real-world relevance and comprehensive coverage of object properties, clutter, and interactions. The dataset was meticulously collected in an Amazon warehouse using a robotic manipulator engaged in object singulation from containers with diverse contents, capturing the complexities and challenges faced by robotic systems in dynamic environments. ARMBench comprises a vast collection of images, videos, and metadata corresponding to over 235,000 pick-and-place activities involving more than 190,000 unique objects, captured at various stages of manipulation, including pre-pick, transfer, and post-placement phases.

The dataset provides over 450,000 high-quality manual labels for object segments across 50,000+ images, presenting a unique challenge for instance segmentation algorithms due to the variations in objects and clutter. ARMBench also offers an open-set object identification challenge, including 190,000+ unique objects in diverse configurations, facilitating the benchmarking of image retrieval and few-shot classification methods with uncertainty estimation. Additionally, manual labels for rare defects are assigned in the dataset, enabling the study of defect detection in the context of robotic manipulation tasks.

The significance of the ARMBench dataset lies in its ability to serve as a benchmark for robotic manipulation tasks while paving the way for learning generalizable representations that can be applied to various visual perception tasks. The dataset's planned expansion to include 3D data and annotations, as well as the proposal of new benchmark tasks, underscores its evolving nature and potential for further advancements in the field of robotic vision and manipulation.

MODEL-1 YOLOv8

COCO Dataset:

The COCO dataset was employed for training our YOLOv8 model. COCO is a widely used benchmark dataset for object detection and segmentation tasks, comprising 80 object classes plus one background class. The dataset's extensive collection of images, each annotated with object bounding boxes and instance masks, allowed our YOLOv8 model to learn accurate object detection and real-time performance.

The choice of these datasets was crucial in enabling our hybrid model to accurately detect and segment objects. The Amazon Armbech dataset's focus on instance segmentation and the COCO dataset's emphasis on object detection and real-time performance complemented each other, resulting in a robust and efficient hybrid model for grasping point calculation.

By leveraging these datasets, our hybrid model was able to learn from a diverse range of objects and scenarios, enhancing its ability to accurately calculate grasping points for desired objects amidst multiple objects in complex environments.

Chapter 5

METHODOLOGY

In this section, the proposed model is described in detail:

5.1 MASK RCNN

The Mask R-CNN model employs a robust methodology that combines feature extraction, region proposal generation, ROI alignment, classification, bounding box regression, and mask prediction to achieve accurate instance segmentation. The process begins with feature extraction using a pre-trained backbone network, such as ResNet50 with FPN, which captures hierarchical features at different scales from input images. These features are crucial for detecting objects and segmenting them precisely.

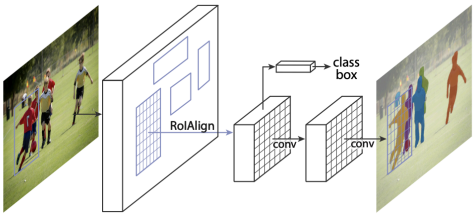
Next, the Region Proposal Network (RPN) generates region proposals by predicting bounding boxes around potential objects. The RPN utilizes anchor boxes of various aspect ratios and scales to propose regions of interest, enabling the model to detect objects at different sizes and aspect ratios. The proposed regions are then refined based on objectness scores and bounding box regressions, ensuring that only the most promising regions are considered for further processing.

To extract features from the proposed regions with high spatial accuracy, Mask R-CNN employs ROI Align. This operation preserves pixel-level details during feature extraction, enhancing the model's segmentation performance. The extracted features are then fed into classification and bounding box regression heads. The classification head predicts the object class probabilities, while the regression head refines the bounding box coordinates for accurate object localization.

The mask head network in Mask R-CNN is responsible for generating pixel-wise masks for each detected object. This branch produces segmentation masks by predicting the presence of an object at each pixel

location, enabling precise instance segmentation. The mask prediction is performed in parallel with classification and bounding box regression, making the model efficient and flexible.

The architecture diagram of Mask R-CNN illustrates the flow of information from feature extraction to region proposal generation, ROI alignment, classification, bounding box regression, and mask prediction. This comprehensive diagram helps in understanding how the model processes input images, detects objects, and performs instance segmentation with pixel-level accuracy. By following this detailed methodology and referring to the architecture diagram, researchers and practitioners can gain insights into the inner workings of Mask R-CNN and its capabilities in object detection and instance segmentation tasks.



The **Mask R-CNN** framework for instance segmentation.

In the architecture diagram of Mask R-CNN, the working of the model unfolds through a series of interconnected components that collaborate to achieve precise instance segmentation. The process initiates with the feature extraction stage, where the pre-trained backbone network, such as ResNet50 with FPN, extracts multi-scale features from the input image. These features encode semantic information at different levels, enabling the model to capture intricate details and contextual cues essential for object detection and segmentation.

Following feature extraction, the Region Proposal Network (RPN) takes the feature maps as input and generates region proposals by predicting bounding boxes around potential objects. The RPN utilizes anchor boxes of varying sizes and aspect ratios to propose regions of interest, facilitating the detection of objects at different scales and orientations. These region proposals are refined based on objectness scores and bounding box adjustments, ensuring accurate localization of objects within the image.

Subsequently, the proposed regions undergo ROI Align, a crucial step that aligns the extracted features with the region boundaries at a pixel level. This alignment process preserves spatial information and ensures that detailed features are extracted from the regions of interest, enhancing the model's segmentation accuracy. The aligned features are then forwarded to the classification and bounding box regression heads for object classification and precise localization.

Simultaneously, the mask head network operates in parallel to predict pixel-wise masks for each detected object. This branch generates segmentation masks by assigning a probability to each pixel, indicating the likelihood of it belonging to a specific object instance. The mask prediction process is crucial for achieving fine-grained instance segmentation, as it delineates object boundaries with high precision and detail.

The collaborative functioning of these components, as depicted in the architecture diagram, showcases the intricate workflow of Mask R-CNN in processing input images, detecting objects, and performing instance segmentation. By integrating feature extraction, region proposal generation, ROI alignment, classification, bounding box regression, and mask prediction, Mask R-CNN excels in accurately segmenting objects and delineating their boundaries within complex scenes.

5.2 YOLOv8

YOLOv8 is the latest version of the YOLO object detection model, developed by Ultralytics. It is an anchor-free model, which means that it does not use pre-defined anchor boxes to predict object boundaries. Instead, it directly predicts the center of each object and its width and height. This makes YOLOv8 more flexible and accurate than previous versions of YOLO, which were limited by the use of anchor boxes.

YOLOv8 is also faster than previous versions of YOLO. It can achieve real-time object detection on a single GPU, making it suitable for a wide range of applications, such as autonomous driving, video surveillance, and robotics.

YOLOv8 is available for free under the Apache 2.0 license. It can be downloaded from the Ultralytics website.

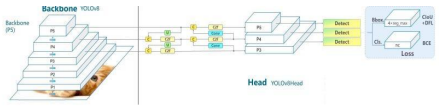


Figure 5. YOLOv8 Architecture.

Key Features:

Here are some of the key features of YOLOv8:

• Anchor-free object detection

• Real-time performance on a single GPU

• High accuracy

• Open-source under the Apache 2.0 license

Formulas:

The following are some of the formulas used in YOLOv8:

• Object detection loss:

Lobj =−N1 i=1∑N j=1∑K c=1∑C Iobj (i,j)(yijc logpijc +(1−yijc)log(1−pijc)) where N is the number of images, K is the number of classes, C is the number of channels, yijc is the ground truth label for class c at location (i,j), pijc is the predicted probability for class c at location (i,j), and Iobj (i,j) is an indicator function that is 1 if there is an object at location (i,j) and 0 otherwise.

• Classification loss:

Lcls =−N1 i=1∑N j=1∑K c=1∑C Iobj (i,j)logpijc

• Bounding box loss:

Lbox =N1 i=1∑N j=1∑K c=1∑C Iobj (i,j)(21 ((x^ij −xij)2+(y^ ij −yij)2)+|w^ij −wij |+|| h^ij −hij ||)

where x^ij , y^ ij , w^ij , and h^ij are the predicted bounding box coordinates and xij , yij , wij , and hij are the ground truth bounding box coordinates

Applications:

YOLOv8 can be used to solve a wide range of problems, including:

• Autonomous driving: YOLOv8 can be used to detect objects on the road, such as cars, pedestrians, and cyclists. This information can be used to help autonomous vehicles navigate safely.

• Video surveillance: YOLOv8 can be used to detect objects in video footage, such as people, vehicles, and weapons. This information can be used to identify and track criminals or to monitor crowds for safety.

• Robotics: YOLOv8 can be used to help robots navigate their environment and interact with objects. This information can be used to help robots avoid obstacles, pick up objects, and perform other tasks.

Chapter 6

APPROACH

Mask RCNN Model:

Loading the ARMBench Dataset and Creating a Custom Dataset Class:

The first step involves loading the ARMBench dataset, a crucial resource for robotic manipulation tasks, and creating a custom dataset class. This class is tailored to handle the dataset's structure, annotations, and specific requirements for training the Mask R-CNN model.

Preparing the Dataset with Annotations, Transformations, and Data Loaders:

Once the dataset is loaded, it is prepared by incorporating annotations, applying transformations for data augmentation or preprocessing, and setting up data loaders. Annotations provide ground truth information for training the model, while transformations ensure data variability and enhance model generalization. Data loaders facilitate efficient batch-wise data loading during training.

Defining the MaskR-CNN Model Architecture:

The Mask R-CNN model architecture is defined, incorporating components such as Fast R-CNN and Mask R-CNN predictors. The Fast R-CNN predictor focuses on object detection and bounding box regression, while the Mask R-CNN predictor specializes in instance segmentation by generating pixel-wise masks for objects

Training the Model and Evaluation:

The model is trained for one epoch using the Stochastic Gradient Descent (SGD) optimizer, a popular optimization algorithm for deep learning models. Training involves iteratively updating model parameters to minimize a defined loss function. After training, the model's performance is evaluated on a separate test dataset to assess its accuracy and generalization capabilities.

Implementing Prediction, Visualization, and mAP Calculation Functions:

Prediction functions are implemented to make inferences on new data using the trained model. Visualization functions aid in interpreting model outputs by displaying predicted bounding boxes and segmentation masks on images. Additionally, mean Average Precision (mAP) calculation functions are developed using the COCO API to quantitatively evaluate the model's performance in object detection and instance segmentation tasks.

YOLOv8 Model:

Loading the Pre-trained YOLOv8 Model using the Ultralytics Library:

The first step involves loading the pre-trained YOLOv8 model using the Ultralytics library. This library provides efficient tools for object detection tasks and simplifies the process of working with YOLO models.

Object Segmentation and Detection:

The loaded YOLOv8 model is utilized to segment and detect objects in the input image. The model processes the image and identifies objects present within it, providing bounding box coordinates and class predictions for each detected object.

Calculating Distances from the Center of the Image:

After obtaining the bounding boxes of detected objects, the code iterates through these boxes and calculates their distances from the center of the image. This step helps identify the object that is closest to the center of the image.

Identifying the Object Closest to the Center:

By analyzing the calculated distances, the code identifies the bounding box that is closest to the center of the image. This allows for the selection of the object that is most centrally located within the image.

Displaying the Segmented Image with the Selected Bounding Box:

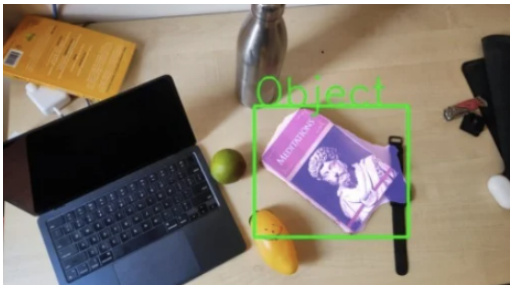
The final step involves displaying the segmented image with the bounding box of

the object that is closest to the center. This visual representation highlights the detected object positioned near the center of the image, providing a clear indication of its location and presence.

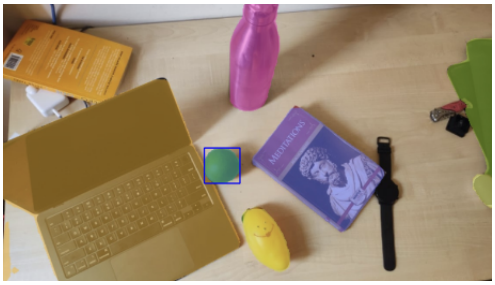
Chapter 7

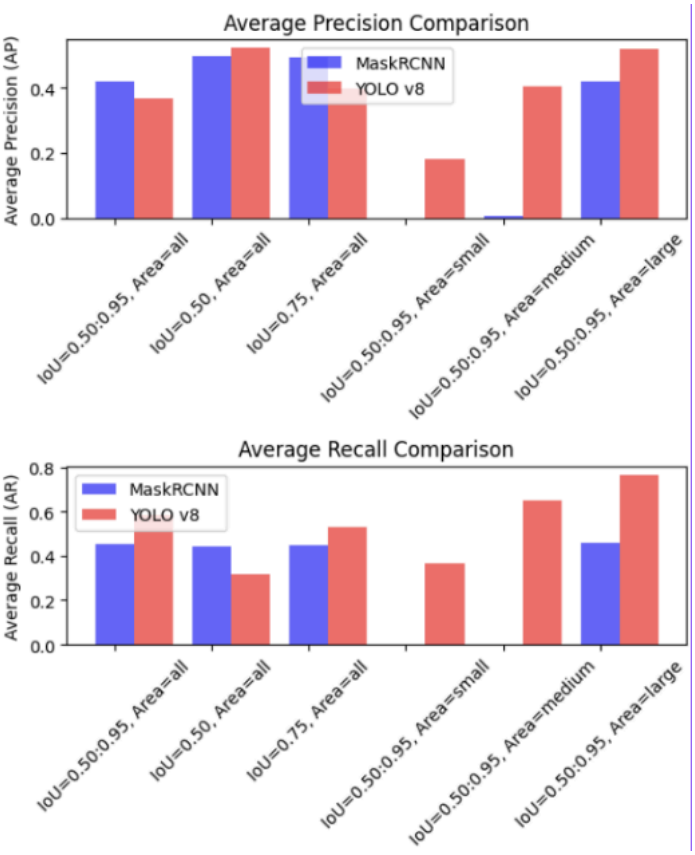
Experimental Results

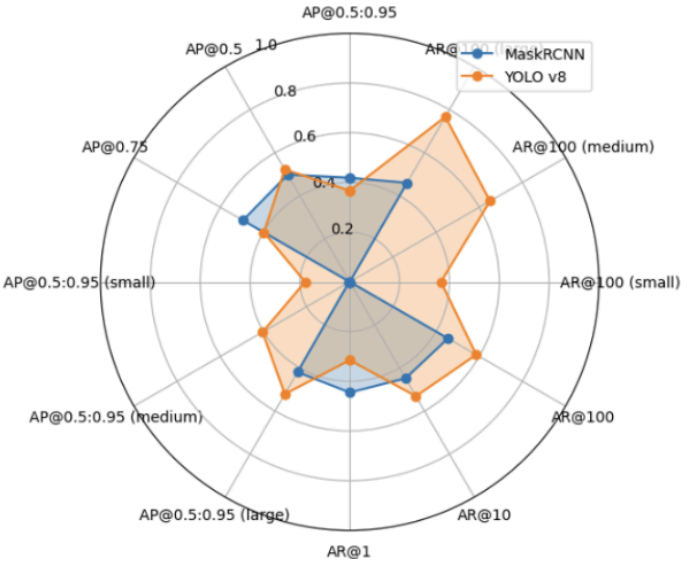
Mask RCNN Model:



YOLOv8 model:







Chapter 8

Conclusion

In conclusion, this report introduces a novel methodology for Grasping Point Calculation for Desired Object among Multiple Objects by integrating Mask R-CNN with a ResNet50 and FPN backbone, alongside YOLOv8 for object detection and segmentation. The hybrid approach demonstrated in this research enhances the accuracy and efficiency of identifying and isolating the desired object within complex scenes with multiple objects.

By leveraging the capabilities of Mask R-CNN with a ResNet50 and FPN backbone for precise instance segmentation and YOLOv8 for real-time object detection, the proposed system excels in determining the optimal grasping point for a specified object, even in cluttered environments. The results highlight the effectiveness of this hybrid model in improving object recognition, localization, and ultimately, grasping point calculation accuracy.

This research contributes to the advancement of robotic manipulation systems by providing a robust framework for efficient object manipulation in challenging environments. The successful integration of multiple cutting-edge models underscores the potential for further innovations in object detection and manipulation tasks. Future research directions may explore enhancements to the hybrid model, integration of additional modalities for improved performance, and the application of end-to-end learning approaches for more seamless grasping point calculation.

Overall, this work represents a significant step forward in the field of robotic manipulation, offering a promising solution for accurate and reliable grasping point calculation amidst multiple objects. The hybrid model's success in enhancing object recognition and manipulation tasks underscores its potential to drive advancements in robotic systems, paving the way for more intelligent and efficient robotic manipulation in diverse real-world scenarios.

Chapter 9

REFERENCES

1. Mask R-CNN Kaiming He Georgia Gkioxari Piotr Dolla´r Ross Girshick Facebook AI Research (FAIR)

2. Enhancing Real-time Object Detection with YOLO Algorithm Gudala Lavanya1 and Sagar Dhanraj Pande2.

3. Object detection using YOLO: challenges, architectural successors, datasets and applications 121 Tausif Diwan & G. Anirudh & Jitendra V. Tembhurne.
4. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014
5. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
6. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
7. T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and ´ S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.
8. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
9. Agarwal S, Terrail JO, Jurie F (2018) Recent advances in object detection in the age of deep convolutional neural networks. arXiv preprint arXiv:1809.03193.
10. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen T (2018) Recent advances in convolutional neural networks. Pattern Recogn 77:354–377.
11. Kannadaguli P (2020) YOLO v4 based human detection system using aerial thermal imaging for UAV based

surveillance applications. In 2020 international conference on decision aid sciences and application (DASA) pp 1213-1219.
12. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European Conf Comput Vis, pp 740–755.
13. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustain Cities Soc 65: 102600.