

RoboLLM: Robotic Vision Tasks Grounded on Multimodal Large Language Models

Zijun Long¹, George Killick¹, Richard McCreadie¹ and Gerardo Aragon-Camarasa¹

Abstract—Robotic vision applications often necessitate a wide range of visual perception tasks, such as object detection, segmentation, and identification. While there have been substantial advances in these individual tasks, integrating specialized models into a unified vision pipeline presents significant engineering challenges and costs. Recently, Multimodal Large Language Models (MLLMs) have emerged as novel backbones for various downstream tasks. We argue that leveraging the pre-training capabilities of MLLMs enables the creation of a simplified framework, thus mitigating the need for task-specific encoders. Specifically, the large-scale pretrained knowledge in MLLMs allows for easier fine-tuning to downstream robotic vision tasks and yields superior performance. We introduce the RoboLLM framework, equipped with a BEiT-3 backbone, to address all visual perception tasks in the ARMBench challenge—a large-scale robotic manipulation dataset about real-world warehouse scenarios. RoboLLM not only outperforms existing baselines but also substantially reduces the engineering burden associated with model selection and tuning. All the code used in this paper can be found in <https://github.com/longkukuhi/RoboLLM>.

I. INTRODUCTION

Recent advances in deep learning have led to the emergence of Multimodal Large Language Models (MLLMs) such as BLIP2 [1] and BEiT-3 [2]. These MLLMs, trained on large-scale datasets, act as general-purpose encoders for multiple modalities and offer substantial potential for transfer to various applications [3], [4], [5], [6]. While they have shown state-of-the-art performance in fields such as conversational agents, their utility in robotic vision remains largely unexplored. Early studies, like PaLM-E [7], have begun to delve into this area but primarily focus on vision and language understanding, largely overlooking the unique real-world challenges of robotic vision, such as task-specific camera poses, variable lighting, and clutter. Therefore, prompted by the success of MLLMs in other domains, this paper addresses the question: *To what extent can MLLMs improve vision tasks specific to the robotics domain?*

The recently introduced ARMBench dataset [8] from Amazon exemplifies the complexities inherent to robotic vision. It comprises three critical robotic vision perception tasks: object instance segmentation, object identification, and defect detection, and presents a large-scale representation of real-world scenarios—specifically, object grasping and manipulation tasks in Amazon warehouses. These tasks demand a vision system capable of handling an extremely large range of objects (e.g., 190K+ unique objects), robust to variable

lighting conditions, and capable of operating effectively in cluttered environments. Moreover, the ever-changing inventory of warehouses requires the vision system to perform well in transfer learning scenarios. The individual tasks, though extensively researched, have yielded task-specific models, complicating their integration into a unified vision pipeline. Such integration involves selecting the right model for each task, coordinating them effectively, and fine-tuning each one – a process that is both time-consuming and challenging in terms of engineering.

We argue that MLLMs have better transfer capability than previous task-specific models, owing to MLLMs’ large-scale pre-training, which allows them to serve as robust backbones for these tasks. This substantially reduces the engineering complexity of developing different backbones for multiple tasks and the time required for model selection and tuning. Additionally, MLLMs have shown particular resilience to out-of-distribution examples, such as objects presented in novel poses or amidst visual distractions, which are critical issues in robotic vision perception tasks. In this work, we introduce RoboLLM, a framework designed to adapt MLLMs to the multifaceted challenges of robotic vision, using the ARMBench dataset for evaluation. Our main contributions are as follows:

- We present RoboLLM, a generalized framework that employs pre-trained MLLMs as backbones for tackling complex robotic vision tasks, including object instance segmentation, object identification, and defect detection.
- We are the first work to address all three key vision tasks in the ARMBench dataset, representative of challenging, large-scale robotic manipulation scenarios.
- We introduce a lightweight variant of the BEiT-3 architecture aimed at increased performance and efficiency, thus broadening its applicability to resource-constrained robotic applications.
- Our experiments show that our RoboLLM achieves state-of-the-art results across all three ARMBench tasks.
- We further demonstrate RoboLLM’s robustness to object number variance and its superior performance on out-of-distribution examples in the object segmentation task, where previous works fail. More importantly, RoboLLM solves the object identification task with a 97.8% recall@1.

II. BACKGROUND

A. Emergence of Multimodal Large Language Models

Recently, Multimodal Large Language Models (MLLMs) have achieved new state-of-the-art result, delivering superior

This research has been supported by EPSRC Grant No. EP/S019472/1
¹ School of Computing Science, University of Glasgow, G12 8QQ, Scotland, United Kingdom z.long.2@research.gla.ac.uk

performance across a wide range of vision and vision-language benchmarks. Their exceptional capabilities are most notable in few-shot and zero-shot scenarios, as evidenced by a series of studies [9], [1], [2], [7], [10]. Their efficacy comes from extensive pretraining on large-scale corpora of image-text data, enabling them with superior transfer capabilities. Palm-E, a pioneer in the use of MLLMs in the robotics domain, has exhibited state-of-the-art performance in embodied robotic planning scenarios [7]. Open X-Embodiment [11] amalgamates extensive datasets, showcasing the efficacy of transfer learning with large-scale models. This approach significantly enhances the capabilities of various robots by leveraging shared knowledge across different platforms. Furthermore, research indicates that the multi-modal large-scale training of image-text pairs results in models with enhanced robustness to out-of-distribution examples [12] compared to their unimodal counterparts. This let MLLMs show superior performance in unimodal tasks, such as vision tasks, when compared to vision-specific pretrained models [2], [1]. As such, MLLMs present a compelling case for serving as backbone encoders in complex robotic vision applications, which often consist of multiple sub-tasks. This enables us to improve the effectiveness of robotic vision pipeline yet significantly reduce the engineering efforts and tuning time.

B. Image Segmentation

Object instance segmentation involves simultaneously predicting pixel-level instance masks and their corresponding class labels. Though the most prevalent backbones for these detector models have been Convolutional Neural Networks (ConvNets) [13], such as R-CNN [14] and Faster R-CNN [15], the Vision Transformer (ViT) [16] has emerged as a potent alternative for image classification tasks. The original ViT architecture is non-hierarchical; this characteristic hinders its applicability in object instance segmentation due to a lack of innate translational equivariance [17] and difficulties in handling high-resolution inputs because of the quadratic complexity of self-attention. Therefore, some models aim to mitigate these challenges by incorporating ConvNet designs into ViT, integrating hierarchical structures and translation-equivariant priors such as convolutions, pooling, and sliding windows (e.g., Swin [18], MViT [19], PVT [20]). This approach compromises the model’s general applicability by coupling pre-training and fine-tuning requirements. Subsequent efforts have explored plain ViT backbones specifically for object instance segmentation, such as UViT [21], a single-scale Transformer for object detection. Unlike UViT, ViTDet [22] offers an approach that retains the task-agnostic nature of ViT backbones, thereby facilitating their broader applicability. In this study, we extend the line of inquiry initiated by ViTDet in the field of robotic vision. Our approach employs a plain backbone architecture decoupled from the detection task, enabling the reuse of the same backbone across various robotic vision perception tasks. This strategy significantly alleviates the engineering challenges related to model selection and fine-tuning.

C. Image Retrieval

The object identification task in ARMBench can be tackled as an image retrieval task [23], [24], a well-studied area in computer vision with applications in robotics for scene localization [25] and place recognition [26], [27]. Traditional methods have focused on aggregating ConvNet feature maps using various pooling techniques such as R-MAC [28] and GeM [29], the latter of which has achieved state-of-the-art performance. Recently, the advent of vision transformers [16] has initiated a shift away from ConvNets, often surpassing them in performance [30], [31] and reducing the need for specialized aggregation methods [30]. As for the objective functions for training, they have commonly employed contrastive or triplet loss functions [32], [33], [34], [35], [36], [37]. Thus, contrastive learning has been shown to be effective in training image retrieval tasks.

In this work, we fine-tune a Multimodal Large Language Model (BEiT-3) with contrastive loss, leveraging its large-scale pretraining for effective object identification in the context of robotic vision. We argue that if the produced embeddings are sufficiently good, there is no need for complex feature processing methods or ranking techniques. This aligns well with the objective of RoboLLM, which aims to reduce engineering efforts in model tuning and enhance efficiency.

D. Defect Detection

Research in defect detection has been primarily oriented toward identifying surface defects in materials such as fabric, metals, and concrete [38], [39], [40]. Though the primary goal is to ascertain the existence of a defect, certain applications necessitate the specific type of defect to be classified, localized, and segmented [41], [42]. Feng et al. [43] employed an autoencoder pretraining method for defect detection with limited training data, while Hu et al. [44] introduced a lightweight spatial-temporal model incorporating local attention and PCA reduction to detect thermography defects. In contrast, ARMBench presents a large-scale challenge in which defects can manifest in various forms. Obtaining examples of defects for all objects is often infeasible, necessitating a model that can generalize to defects in unseen forms. We hypothesize that MLLMs, pretrained on large-scale datasets, can significantly improve the model’s ability to handle out-of-distribution samples.

III. MATERIALS AND METHODS

A. Overview

The proposed framework, RoboLLM, addresses three primary vision tasks in robotic manipulations as defined in the ARMBench dataset: *object segmentation*, *object identification*, and *defect detection*. As illustrated in Figure 1, the architecture of RoboLLM is inherently modular. This modular design allows for integrating a variety of Multimodal Large Language Models (MLLMs) as backbone encoders alongside task-specific heads tailored for distinct robotic vision tasks. This design principle of decoupling the backbone from specific downstream tasks offers several

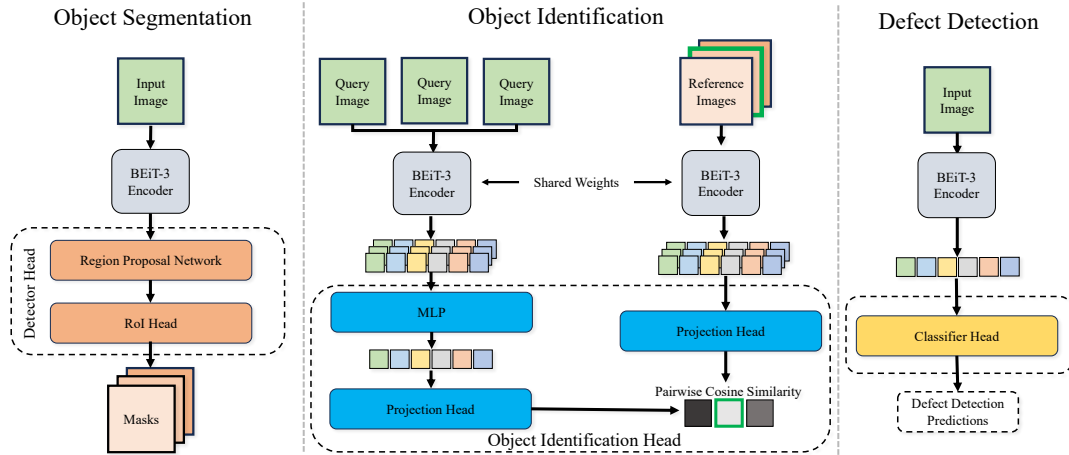


Fig. 1: The task-specific adaptations to BEiT-3, proposed in our RoboLLM framework, for tackling each challenge in ARMBench. BEiT-3’s large-scale vision-language pretraining allows it to be easily and effectively transferred to downstream tasks.

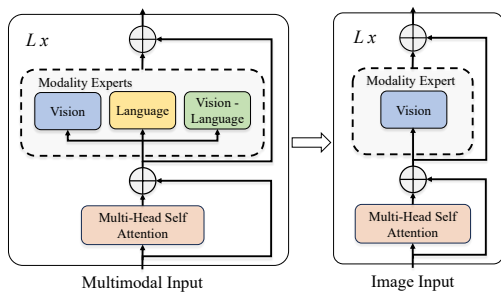


Fig. 2: Our lightweight modification of the BEiT-3 backbone only remains the vision experts.

advantages, including ease of maintenance and flexibility for quick and easy adaptation to different vision tasks while fully exploiting the benefits of large-scale pre-trained models. Task-specific heads are integrated to tackle each unique vision challenge, with the choice of heads justified by their proven effectiveness in the respective tasks. Subsequent sections elaborate on the role of the backbone encoder and provide specific configurations for each vision task in the ARMBench dataset.

B. Backbone Encoder

The backbone encoder is a critical component in the RoboLLM framework, which is responsible for producing feature maps for downstream tasks. It is based on a widely-used MLLM architecture featuring a MultiWay Transformer design [2]. Our design philosophy allows the backbone to be replaced with future MLLM developments, making the system future-proof in the rapidly evolving domain of robotic vision. Currently, we believe BEiT-3 is the most suitable option owing to its state-of-the-art performance in various vision benchmarks [2]. Moreover, BEiT-3 demonstrates that features learned in a multi-modal setting are more task-agnostic and generalized, thereby offering better transferability to different vision tasks without the need for extensive fine-tuning. These characteristics of

BEiT-3 make it highly suitable for robotic vision tasks.

The MultiWay Transformer blocks, illustrated on the left side of Figure 2, comprise a shared self-attention module and a pool of feed-forward networks known as modality experts. These experts are designed for handling different types of data. However, this paper is focused only on addressing robotic vision perception tasks, thereby eliminating the need for other modality experts besides vision. Thus, our streamlined architecture, as shown on the right side of Figure 2, eliminates non-vision modality experts, reducing computational overhead and the parameter count from 222 million to 87 million in the BEiT-3 Base model. This design choice not only improves computational efficiency but also retains the flexibility to incorporate other modalities in future iterations of the task. Importantly, the feature maps generated by the backbone are exploited across all three vision tasks addressed in this work, demonstrating the framework’s effectiveness and efficiency.

C. Object Segmentation

Object segmentation is the prerequisite task within the ARMBench dataset, aiming to identify and delineate individual objects in containers. Aligned with RoboLLM’s overarching philosophy of decoupling the backbone from downstream tasks, we employ a plain-backbone approach for object segmentation. Unlike most object detection approaches, this plain-backbone detection approach does not require any pretraining on detection tasks, thereby eliminating hierarchical constraints on the backbone. This is a necessary step, given the absence of a segmentation task during MLLM pretraining. Specifically, we transfer MLLM to object segmentation tasks, with modifications performed exclusively during the fine-tuning stage. The rationale behind this choice is to maximize the benefits accrued from pre-training on large-scale datasets, which in turn improves segmentation performance. Thus, to translate the capabilities of MLLM into practical applications for object segmentation,

a task-specific head is crucial for integrating and adapting the backbone’s generic feature maps to the task at hand.

1) *Task-Specific Head*: The task-specific head for object segmentation integrates a detector based on a Cascade Mask R-CNN [45]. This modular design facilitates seamless interaction with the feature maps generated by the backbone and is extensible for compatibility with other detector heads.

2) *Feature Pyramid Construction*: The cornerstone of our object detection strategy is the construction of a feature pyramid, following [46]. For this, we utilize only the last embeddings from the backbone, which is hypothesized to contain the most discriminative features. We then execute a series of parallel convolutions and deconvolutions to generate multi-scale feature maps. Specifically, starting with the default Vision Transformer (ViT) embeddings with a scale of $\frac{1}{16}$, we produce feature maps at scales of $\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}$ via convolutional strides of 2, 1, $\frac{1}{2}, \frac{1}{4}$, respectively.

3) *Object Detection Procedure*: These generated multi-scale feature maps are processed through the Cascade Mask R-CNN detector head. A Region Proposal Network (RPN) proposes candidate object bounding boxes, followed by mask generation for each object via a Region-of-Interest (ROI) head, which extracts pertinent features from the RPN.

4) *Advantages and Rationale*: Our design offers multiple benefits. Firstly, the plain-backbone approach necessitates only modest modifications during the fine-tuning stage, which preserves the generality and extensibility of the pre-trained MLLMs. Secondly, the task-agnostic nature of our backbone allows for seamless interchangeability with various detector heads, providing flexibility in addressing a broad range of object segmentation tasks. Lastly, the selective use of only the last feature map for object detection aims to utilize the most potent features, thereby enhancing the system’s overall efficacy and efficiency.

D. Object Identification

Object identification is the second task in the ARMBench dataset. This task is concerned with precisely categorizing detected objects among a database of predefined classes. Unlike traditional classification approaches, which become computationally challenging with a large number of categories (over 190,000 in the Amazon warehouse scenario), we tackle this task as an image retrieval problem.

1) *Task Variants*: In robotic manipulation within the context of the ARMBench dataset, the object identification task holds significance in pre- and post-object manipulation. In the pre-pick stage, object identification permits the retrieval of historically acquired objects or attributes for manipulation planning. While in the post-pick, the object’s unique identifier is crucial for quality control and subsequent tasks. Thus, both pre- and post-pick images of the object could serve as query images, while the reference images in the container manifest act as gallery images. The challenge lies in accurately matching the query images to the gallery images.

Therefore, this task offers two variants: one reliant on pre-pick images (one for each pick) and the other incorporating post-pick images (two more for each pick). While the

```

1 import numpy as np
2 # Q[n, h, w, c] - minibatch of query images
3 # R[n, h, w, c] - minibatch of reference images
4 # Wi[di, de] - learned proj of image to embedding
5 # labels - labels of whether Query image and
6 # reference image are the same class
7
8 # extract feature representations of each modality
9 if using both pre-pick and post-pick images:
10     # BEiT-3 Encoder
11     Qf = encoder(Q) #[n, 3 x di]
12     # Feature Aggregation
13     Qf = MLP(Qf) #[n, 3 x di] to [n, di]
14 else:
15     Qf = encoder(Q) #[n, di]
16
17 Rf = encoder(R) #[n, di]
18 # project embedding [n, de]
19 Qe = l2_normalize(np.dot(Qf, Wi), axis=1)
20 Re = l2_normalize(np.dot(Rf, Wi), axis=1)
21
22 # scaled pairwise cosine similarities [n, n]
23 # t - learned temperature parameter
24 logits = np.dot(Qe, Re.T) * np.exp(t)
25 # symmetric loss function
26 lossQ = cross_entropy(logits, labels, axis=0)
27 lossR = cross_entropy(logits, labels, axis=1)
28 loss = (lossQ + lossR)/2

```

Fig. 3: Numpy-like pseudo-code for the core of an implementation of our framework for Object Identification task.

latter images present a greater challenge due to differing perspectives, object poses, and presentations, they enable the incorporation of multi-view data, thereby enhancing the overall retrieval performance.

2) *Overall Architecture*: Consistent with our object instance segmentation task strategy, we employ a backbone plus a task-specific head architecture as shown in Fig. 1-center. For the two aforementioned task variants, the task-specific head comprises a projection head or adds a Multi-Layer Perceptron (MLP) for fusing pre- and post-pick images. A contrastive learning objective is applied to fine-tune the RoboLLM to optimize detection performance.

3) *Contrastive Learning Fine-tuning*: We argue that high-quality feature maps suffice for measuring the similarity between query and reference images using dot product calculations. We maintain computational efficiency by abandoning complex retrieval techniques such as re-ranking, which is crucial for real-time robotic applications. Thus, a contrastive objective is used in the fine-tuning stage to improve the quality of the generated feature maps. It aims to minimize the distance between the feature maps of positive pairs (the same class) and maximize the distance between negative pairs (different classes), thus enhancing the efficacy of our image retrieval approach.

The backbone encoder transforms an input image or three images input Q into an output representation Q_f , which is further processed by a linear projection head to produce the final feature vector. In cases involving post-pick images as query images, their feature maps are concatenated before input into the MLP, providing a fused representation before feeding into the linear projection head. An $L2$ normalization is applied to Q_f to mitigate the risk of numerical instability

Task	Mixed Object Tote		Zoomed Out Tote		Same Object Tote	
	mAP50	mAP75	mAP50	mAP75	mAP50	mAP75
ResNet50 + Mask RCNN *	0.72	0.61	0.25	0.19	0.11	0.10
RoboLLM	0.82	0.67	0.57	0.45	0.15	0.13

TABLE I: Mean Average Precision at IoU thresholds of 50 and 75 across the different segmentation task subsets. * indicates results obtained from [8].

during training. Specifically, given a batch of N (query image, reference image) pairs, our framework is trained to predict which of the $N \times N$ possible (query image, reference image) pairings across a batch actually occurred. To do this, our framework maximizes the pairwise cosine similarity of the query and reference image feature maps of the N real pairs in the batch, while minimizing the cosine similarity of the feature maps of the $N^2 - N$ incorrect pairings. We optimize a symmetric cross-entropy loss over these similarity scores. In Figure 3, we include the pseudocode of the framework’s core for the object identification task.

4) *Advantages and Rationale*: This approach presents multiple advantages. First, the scalability of the image retrieval method sidesteps the challenges tied to implementing a large-category linear classification layer. Second, the modular design allows task-specific heads to be easily interchanged, making the framework flexible and adaptable to different identification scenarios, whether involving single or multiple query images. Lastly, computational efficiency is assured through using a dot product similarity measure and the elimination of intricate retrieval techniques.

E. Defect Detection

The third task in the ARMBench dataset is defect detection, aimed at identifying defects resulting from specific robotic manipulation activities. This is a crucial task as it directly impacts the integrity of the workflow and the quality of the end product. The dataset includes two types of robot-induced defects: 1) multi-pick, where multiple objects are mistakenly picked and transferred from the source to the destination container; and 2) package-defect, indicating activities that result in the object’s packaging opening or the object deconstructing into multiple parts.

Consistent with the design strategy for earlier tasks, we employ a backbone plus a task-specific-head architecture. We reuse the same backbone encoder employed in the previous tasks. Utilizing the same backbone across multiple tasks ensures a cohesive and streamlined architecture. As in the object identification task, we leverage segmentations to locate objects. Unlike the object identification task, which has many categories, this task comprises only three classes: two types of defects and a nominal type. Therefore, a classification head is sufficient to conduct the classification and is appended to the backbone encoder to make predictions across these three categories. We opt for a standard cross-entropy loss function for training, which is particularly suitable for categorical classification tasks.

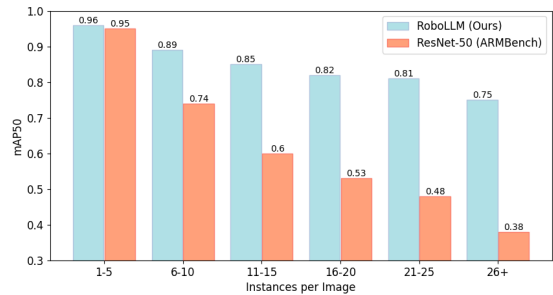


Fig. 4: Number of object instances per image against Mean Average Precision at 50 on the Mixed-Object tote test-set.

Model	Ref Set	Recall@1		Recall@2		Recall@3	
		N=1	N=3	N=1	N=3	N=1	N=3
ResNet50-RMAC	Container	71.7	72.2	81.9	82.9	87.2	88.2
DINO-ViT-S	Container	77.2	79.5	87.3	89.4	91.6	93.5
BEiT-3-Base*	Container	83.7	84.5	83.8	N/A	84.5	N/A
RoboLLM	Container	97.8	98.0	97.9	98.1	98.0	98.2
RoboLLM	All refs	74.6	78.2	82.6	85.7	85.3	89.10

TABLE II: Results on object identification at varying recall@k. * indicates no ARMBench fine-tuning. N=1 uses one pre-pick image, while N=3 uses three images per pick. “Ref set” specifies if reference images are container-specific or from the entire dataset.

IV. EXPERIMENTS

A. Experimental Setup

Our experiments target the ARMBench dataset, focusing on three key robotic vision perception tasks: object instance segmentation, object identification, and defect detection. The experiments are conducted using two state-of-the-art Multi-Modal Learning Models (MLLMs) as the backbones of the proposed RoboLLM framework, namely BEiT-3 Base and BEiT-3 Large. Notably, the BEiT-3 Large model is employed exclusively for the defect detection task because the performance of the BEiT-3 Base model fails to meet the ideal task-specific requirements. We utilize the test sets of all tasks to report metrics. The backbones are initialized with pre-trained weights. A batch size of 64 is employed for all three tasks, with fine-tuning conducted over 50 epochs. An initial learning rate of 2×10^{-4} is set for all tasks. Early stopping is implemented if there is no improvement in the metrics on the validation set over a span of 5 epochs. For the object identification task, the Multi-Layer Perceptron (MLP) consists of two linear layers, with middle dimensions set to 3072 for the Base model and 4096 for the Large model. The projection head is implemented as a single linear layer, with its embedding dimension matching that of the encoder.

B. Object Segmentation

Object segmentation is a critical stage within the robotic vision pipeline, influencing robotic planning, grasp operations, and object identification. Poor object instance segmentation can introduce defects during robotic manipulation, which is highly undesirable. The ARMBench challenge offers three data subsets to evaluate image segmentation performance: 1) *Mixed-Object-tote* serves

as a general benchmark. 2) *Zoomed-out-tote* assesses model generalization to new warehouse environments. 3) *Same-Object* examines segmentation performance on tightly packed instances of identical objects.

We evaluate performance using mean-average-precision at Intersection-over-Union (IoU) thresholds of 0.5 (mAP50) and 0.75 (mAP75) across all subsets, as presented in Table I. Our RoboLLM framework significantly outperforms a ResNet-50 baseline across all tasks: 1) On the *Mixed-Object-tote*, RoboLLM achieves 82% and 67% for mAP50 and mAP75, marking a 10% and 6% improvement over ResNet-50. 2) The performance for RoboLLM on the *Zoomed-out-tote* is much better than it of ResNet-50, demonstrating superior generalization capabilities. 3) *Same-Object* poses the most difficult segmentation challenge, yet BEiT-3 still improves mAP50 by 4% over ResNet-50.

Furthermore, high variation in object instances between containers is common. To investigate model robustness under varying numbers of objects within an image, we report mAP50 in *Mixed-Object-tote* for different numbers of object instances (Figure 4). While both RoboLLM and ResNet-50 perform similarly at fewer than five instances (approximately 95% mAP), performance for ResNet-50 degrades significantly (to 38%) when the number of instances exceeds 26. In contrast, RoboLLM demonstrates a modest decline, maintaining 75% mAP for instances exceeding 26. This evidences that RoboLLM is robust for segmenting a large number of objects within an image.

C. Object Identification

Object identification requires categorizing segmented objects to facilitate robotic planning for subsequent maneuvers. We use Recall@k as the evaluation metric, and the resultant performances are reported in Table II. Using the pre-trained BEiT-3-Base model as a strong baseline, we observe an uplift in recall@1 performance from 77.2% to 83.7%, outperforming the best previously reported result using DINO-ViT-S [8]. Employing our RoboLLM framework, recall@1 is further improved to 97.8% and 98.0% when using only pre-pick images and both pre/post-pick images, respectively. This is a 21% increase over the best prior result [8], effectively solving ARMBench’s object identification task.

1) Performance Under More Challenging Conditions:

We posit that this superlative performance could partially be credited to the specific challenge design, which restricts reference images to objects known to be present in the container. To evaluate RoboLLM in a more generalized scenario, we expand the set of reference images to include all unique objects within the dataset (190k+). As shown on the bottom line of Table II, even under this more demanding setting, RoboLLM maintains an impressive 89.1% at recall@3. This robust performance suggests that our framework can adeptly handle even more complex, large-scale retrieval problems than those posed by the ARMBench challenge. We also find that our framework benefits from including additional query images in this challenging setting. This corroborates our design choice to aggregate multiple query images into

Task	Multi-Pick			Package Defect			Combined		
	Precision	Recall	FPR	Precision	Recall	FPR	Precision	Recall	FPR
ResNet50 *	-	0.34	0.05	-	0.73	0.05	-	0.57	0.05
RoboM-LLM-B	0.84	0.98	0.04	0.94	0.91	0.04	0.90	0.94	0.03
RoboM-LLM-L	0.82	1.00	0.04	0.89	0.95	0.03	0.86	0.97	0.03

TABLE III: Metrics for defect detection tasks with RoboM-LLM B and L backbones. * denotes [8] results. Combined metrics in right column. ARMBench ideal performance: $recall > 0.95$, $FPR < 0.01$.

a single representation for retrieval, thereby enhancing the model’s robustness and versatility.

D. Defect Detection

The practical deployment of automated defect detection in commercial warehouse settings presents substantial challenges, given the high demand of the performance. To contextualize, the ARMBench challenge outlines ideal performance criteria, requiring a recall rate exceeding 0.95 and an FPR below 0.01. Our results for precision, recall, and FPR are reported in Table III. Our experiments demonstrate significant performance enhancements over a ResNet-50 baseline. Specifically, our RoboLLM with BEiT-Base achieves a combined recall rate of 94% across both types of defect detection, marking a 37% improvement over the baseline. Additionally, the combined FPR is reduced from 0.05 to 0.03. Despite these significant improvements, the system does not entirely meet the high recall and FPR criteria set forth by the ARMBench challenge.

To meet the desired performance set by ARMBench challenges, we conduct further experiments with a more powerful and larger model, BEiT-3 Large. Benefiting from our versatile yet robust framework, it is easy to adopt more powerful backbones as needed. Our results show that this model attains a combined recall rate of 97%, surpassing the desired recall target. However, the FPR remains at 0.03, indicating room for future enhancements in effectiveness. Overall, our RoboLLM framework shows significant improvements over existing methods, while a gap in the FPR warrants further investigation in future work.

V. CONCLUSIONS

In this paper, we introduce RoboLLM, an efficient and effective framework designed to establish a unified robotic vision pipeline aimed at the newly released Amazon ARM-Bench dataset [8], utilizing a streamlined BEiT-3 as a Multi-Modal Large Language Model backbone encoder. We evaluate our framework on three distinct visual perception tasks: object segmentation, object identification, and defect detection, which are representative of large-scale real-world robotic vision problems. We show that RoboLLM significantly outperforms previous benchmarks across all three challenges. Notably, RoboLLM achieves this with inherent knowledge from pretraining on large-scale multimodal data and only a minimal task-specific head for each task, significantly increasing performance and mitigating the engineering challenges associated with each. Furthermore, the modular design of the proposed RoboLLM makes incorporating more powerful backbones easy, as well as task-specific modules, to push performance further in the future if required.

REFERENCES

- [1] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*.
- [2] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu *et al.*, "Image as a foreign language: Beit pretraining for all vision and vision-language tasks," *CoRR*, vol. abs/2208.10442, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2208.10442>
- [3] Z. Yi, X. Wang, I. Ounis, and C. Macdonald, "Multi-modal graph contrastive learning for micro-video recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Conference Proceedings, pp. 1807–1811.
- [4] Z. Yi, Z. Long, I. Ounis, C. Macdonald, and R. McCreddie, "Large multi-modal encoders for recommendation," *arXiv preprint arXiv:2310.20343*, 2023.
- [5] Z. Long and R. McCreddie, "Is multi-modal data key for crisis content categorization on social media?" in *19th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2022)*, Conference Proceedings.
- [6] Z. Long, G. Killick, R. McCreddie, and G. A. Camarasa, "Multiway-adapater: Adapting large-scale multi-modal models for scalable image-text retrieval," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2024)*, Conference Proceedings.
- [7] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.
- [8] C. Mitash, F. Wang, S. Lu, V. Terhuja, T. Garaas, F. Polido, and M. Nambi, "Armbench: An object-centric benchmark dataset for robotic manipulation," *arXiv preprint arXiv:2303.16382*, 2023.
- [9] J. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr *et al.*, "Flamingo: a visual language model for few-shot learning," in *NeurIPS*, 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177cccbb411a7d800-Abstract-Conference.html
- [10] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "Coca: Contrastive captions are image-text foundation models," *Trans. Mach. Learn. Res.*, vol. 2022, 2022. [Online]. Available: <https://openreview.net/forum?id=Ee277P3AYC>
- [11] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, and A. Brohan, "Open x-embodiment: Robotic learning datasets and rt-x models," in *arXiv preprint arXiv:2310.08864*, 2023.
- [12] D. Mayo, J. Cummings, X. Lin, D. Gutfreund, B. Katz, and A. Barbu, "How hard are computer vision datasets? calibrating dataset difficulty to viewing time," 2023. [Online]. Available: <https://openreview.net/forum?id=zA7hVj3rR19>
- [13] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
- [14] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 580–587.
- [15] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: <http://arxiv.org/abs/1506.01497>
- [16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [17] P. Ding, D. Soselia, T. Armstrong, J. Su, and F. Huang, "Reviving shift equivariance in vision transformers," *CoRR*, vol. abs/2306.07470, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2306.07470>
- [18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 9992–10002.
- [19] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 6804–6815.
- [20] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 548–558.
- [21] W. Chen, X. Du, F. Yang, L. Beyer, X. Zhai, T. Lin, H. Chen, J. Li, X. Song, Z. Wang, and D. Zhou, "A simple single-scale vision transformer for object localization and instance segmentation," *CoRR*, vol. abs/2112.09747, 2021. [Online]. Available: <https://arxiv.org/abs/2112.09747>
- [22] Y. Li, H. Mao, R. B. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IX*, ser. Lecture Notes in Computer Science, vol. 13669. Springer, 2022, pp. 280–296.
- [23] X. Ge, F. Chen, J. M. Jose, Z. Ji, Z. Wu, and X. Liu, "Structured multi-modal feature embedding and alignment for image-sentence retrieval," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 5185–5193.
- [24] X. Ge, F. Chen, S. Xu, F. Tao, and J. M. Jose, "Cross-modal semantic enhanced interaction for image-sentence retrieval," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1022–1031.
- [25] A. Anosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, "Night-to-day image translation for retrieval-based localization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5958–5964.
- [26] Z. Chen, F. Maffra, I. Sa, and M. Chli, "Only look once, mining distinctive landmarks from convnet for visual place recognition," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*. IEEE, 2017, pp. 9–16. [Online]. Available: <https://doi.org/10.1109/IROS.2017.8202131>
- [27] Z. Long, R. McCreddie, and M. Imran, "Crisisvit: A robust vision transformer for crisis image classification," in *20th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2023)*, Conference Proceedings.
- [28] G. Toliás, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: <http://arxiv.org/abs/1511.05879>
- [29] F. Radenovic, G. Toliás, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, 2019. [Online]. Available: <https://doi.org/10.1109/TPAMI.2018.2846566>
- [30] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, "Training vision transformers for image retrieval," *CoRR*, vol. abs/2102.05644, 2021. [Online]. Available: <https://arxiv.org/abs/2102.05644>
- [31] C. H. Song, J. Yoon, S. Choi, and Y. Avrithis, "Boosting vision transformers for image retrieval," in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*. IEEE, 2023, pp. 107–117. [Online]. Available: <https://doi.org/10.1109/WACV56688.2023.00019>
- [32] "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.
- [33] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2016, pp. 378–383.
- [34] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016*.
- [35] Z. Long, R. McCreddie, G. Aragon Camarasa, and Z. Meng, "Lacvit: A label-aware contrastive fine-tuning framework for vision transformers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2024)*, Conference Proceedings.

- [36] Z. Long, G. Killick, L. Zhuang, R. McCreadie, G. A. Camarasa, and P. Henderson, "Elucidating and overcoming the challenges of label noise in supervised contrastive learning," in *arXiv preprint arXiv:2311.16481*, 2023.
- [37] Z. Long, G. Killick, L. Zhuang, G. Aragon-Camarasa, Z. Meng, and R. McCreadie, "Clce: An approach to refining cross-entropy and contrastive learning for optimized learning fusion," in *arXiv preprint arXiv:2402.14551*, 2024.
- [38] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14 531–14 544, 2020. [Online]. Available: <https://doi.org/10.1109/ACCESS.2020.2966881>
- [39] Q. Luo, X. Fang, L. Liu, C. Yang, and Y. Sun, "Automated visual defect detection for flat steel surface: A survey," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 626–644, 2020. [Online]. Available: <https://doi.org/10.1109/TIM.2019.2963555>
- [40] A. Rasheed, B. Zafar, A. Rasheed, N. Ali, M. Sajid, S. H. Dar, U. Habib, T. Shehryar, and M. T. Mahmood, "Fabric defect detection using computer vision techniques: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–24, 2020.
- [41] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020.
- [42] F. Li and Q. Xi, "Defectnet: Toward fast and effective defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.
- [43] S. Feng, H. Zhou, and H. Dong, "Using deep neural network with small dataset to predict material defects," *Materials & Design*, vol. 162, pp. 300–310, 2019.
- [44] B. Hu, B. Gao, W. L. Woo, L. Ruan, J. Jin, Y. Yang, and Y. Yu, "A lightweight spatial and temporal multi-feature fusion network for defect detection," *IEEE Trans. Image Process.*, vol. 30, pp. 472–486, 2021. [Online]. Available: <https://doi.org/10.1109/TIP.2020.3036770>
- [45] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.
- [46] Y. Li, H. Mao, R. B. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IX*, ser. Lecture Notes in Computer Science, S. Avidan, G. J. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., vol. 13669. Springer, 2022, pp. 280–296. [Online]. Available: https://doi.org/10.1007/978-3-031-20077-9_17