Comparison and Analysis of RT-DETR Module in Retico Pipelines Against YOLO Modules

Porter Rigby
Boise State University / 1910 W University Dr, Boise, ID 83725
porterrigby@u.boisestate.edu

Abstract

Object detection plays a critical role in creating engaging and enjoyable human-robot interactions, and is a core building block for making robots capable of incremental dialogue. Two new object detection modules for the ReTiCo framework are proposed to challenge the existing YOLOv8 module, based on the RT-DETR and YOLOv11 models respectively. Both modules provide improved perception of responsiveness, quickness, and consistency in human-robot interactions on limited hardware like the Anki Cozmo robot under a controlled environment, and uniquely contribute to the existing set of ReTiCo modules.

1 Introduction

Since their inception, YOLO models have become a strong contender for performing object detection tasks in real-time systems based on their notable speed (Redmon et al., 2015). Other models, such as the Detection Transformer model (Carion et al., 2020), have been proposed that offer better precision, but until recently have struggled to compete against YOLOs in both speed and the amount of computing power required to run. Recent work, however, has produced a Real-Time Detection Transformer (RT-DETR) promising to perform both faster and more precisely than traditionally used YOLO models in speed-critical tasks (Zhao et al., 2024). While Zhao et al. have studied the effectiveness of RT-DETR when given optimal input, there is no current research publicly available on the performance of RT-DETR models when used on limited hardware such as the camera feed from Anki's Cozmo robot, nor in the use of an incremental dialogue system like ReTiCo (Michael, 2020). Based on the findings of Zhao et al., this paper aims to explore the use of RT-DETR in an incremental dialogue environment and proposes a ReTiCo module for object detection tasks utilizing the Real-Time Detection Transformer model.

One of the core goals for achieving good human-robot interactions is creating systems that are fast and highly responsive, as such interactions occur in real-time and require systems capable of responding to new information quickly. This is especially true when considering vision, as it is one of the most important robot modalities (Robinson et al., 2023) and plays a critical role in how well a robot can interpret and act on its surroundings. Currently, the ReTiCo incremental framework utilizes a module based on YOLOv8 for object detection tasks requiring high speed and decent precision. This paper explores the use of a new RT-DETR module against the existing YOLOv8 module, as well as against an updated version of the module using YOLOv11, and shows the potential advantages of using RT-DETR and YOLOv11 over YOLOv8 in ReTiCo pipelines.

The rest of this paper is as follows. First, related and relevant work is introduced in section 2. Section 3 then discusses the methods used for developing and evaluating both the RT-DETR and YOLOv11 modules. The individual experiments and evaluations performed on each module are discussed in section 4, and section 5 interprets the results of the project and potential future directions that might be taken with this research.

2 Related Work

2.1 ReTiCo Framework

The ReTiCo framework is an incremental dialogue processing framework designed for modular and distributed operation (Michael, 2020). At the core of ReTiCo is the idea of passing incremental units (IUs), which hold some basic form of data, between modules responsible for handling and manipulating those IUs. The framework is highly extensible, and in addition to providing a wide range of prebuilt modules, allows for the creation and use of custom modules. Currently, the framework includes a

module for object detection based on YOLOv8 but lacks a module based on a detection transformer architecture. This paper introduces a new module based on the real-time detection transformer architecture, providing ReTiCo with the higher Mean Average Precision that RT-DETR is capable of producing (Zhao et al., 2024) while still being able to function in real-time.

2.2 YOLO

Proposed in 2015, the YOLO object detection model offers an incredibly fast option for real-time object detection tasks (Redmon et al., 2015). YOLO, or You Only Look Once, utilizes an architecture that allows processing an image once to produce predictions for the entire image. The algorithm works by splitting each image into a grid of regions and using each region to make localized predictions for the image. Because each grid region can make multiple predictions, it is possible for redundant outputs to exist. Non-maximum suppression is then used to ignore these redundant outputs by filtering out predictions that overlap too much with existing ones. YOLO has gone through many different iterations, with the current newest iteration being YOLOv11. In addition to evaluating RT-DETR, the research presented in this paper evaluates the efficacy of YOLOv11 against YOLOv8 within the ReTiCo framework for speed-critical object detection tasks.

2.3 RT-DETR

The proposed RT-DETR ReTiCo module is motivated by the potential optimization gains to be had from a model that does not rely on Non-Maximum Suppression (NMS) post-processing (Zhao et al., 2024). While YOLO models in general utilize NMS to remove duplicate or redundant predictions, RT-DETR inherently avoids such redundancies altogether by filtering out predictions beneath a given confidence threshold during inference. Zhao et al. suggest that RT-DETRs are fast enough to be used in real-time applications while maintaining their increased precision over YOLO models because they lack the need for NMS post-processing. An additional and interesting byproduct of avoiding NMS as noted by their research is an increased consistency in speed during inference. Models that require NMS post-processing can be less consistent with regard to speed, as the amount of redundant predictions that must be filtered varies. This paper tests the proposed gain in consistency to determine if it noticeably contributes to better human-robot interactions, as well as if RT-DETRs can operate fast enough to avoid degrading such interactions.

3 Methods

As part of the research done for this project, this paper proposes two new ReTiCo modules for real-time object detection tasks: one module based on YOLOv11, and a module based on RT-DETR. The main goal of the project was to evaluate the efficacy of using RT-DETR in ReTiCo pipelines and to compare its validity to the existing YOLOv8 module as a baseline. In addition, an updated version of the YOLOv8 module was created based on the newer YOLOv11 model as another point of reference and comparison. Both the YOLOv11 and RT-DETR modules function as drop-in replacements for the existing YOLOv8 module and each is capable of outputting bounding box and label predictions for detected objects. Currently, all three modules are only configured to output bounding box predictions. While both YOLO modules utilize the Ultralytics Python library for inference, the RT-DETR module instead uses the Hugging Face Transformers library as the ResNet18 and ResNet50 versions of RT-DETR are not currently available for use through Ultralytics [1].

The RT-DETR module can currently be configured to use the following two model sizes, listed from smallest to largest:

- 'r18'
- 'r50'

Similarly, both YOLO modules can be configured to use the five sizes listed below:

- 'n'
- 's'
- 'm'
- 'l'
- 'x'

Data was collected for both RT-DETR model sizes, and for YOLO sizes 'n'-'m'.

[1]The difference in code libraries used for inference potentially introduces an inconsistency between the YOLO and RT-DETR modules, possibly skewing results due to different levels of code efficiency. This is a point of interest for future research into the same topic.

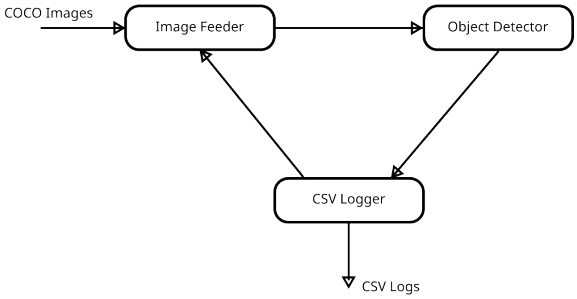

Figure 1: IU Data Logging Pipeline

Each module was tested both quantitatively and qualitatively, on a variety of metrics ranging from speed to consistency and responsiveness. Modules were tested for speed and detection output capabilities using the aforementioned model sizes, and tested in a ReTiCo pipeline for a symbol grounding task requiring human participants to rate each module on various qualitative metrics.

All modules in question were run on an RTX 3080ti laptop GPU.

4 Experiments

4.1 Experiment 1

4.1.1 Task & Procedure

The first experiment performed was a timing analysis of various model sizes for each module. A 100-sample subset of the COCO 2017 Test dataset was fed sequentially into each module, once for each model size, to collect IU data from the ReTiCo pipeline. Every ReTiCo IU by default contains a timestamp indicating when the IU was created, allowing for the difference between each IU's creation time to be calculated and logged into a CSV file for data analysis. Two custom testing modules were implemented to facilitate this experiment. The first was a module specifically for feeding COCO images into the object detection modules. The second was a module that would both log incoming IU data into a CSV file and detect when the object detection modules were done processing an image to indicate back to the image feeder that the next image could be sent. Figure 1 shows a diagram of the pipeline that was used for the first experiment.

4.1.2 Metrics & Baseline

Three different metrics were analyzed during this experiment for each model at 10% confidence threshold increments. The data collected for the existing YOLOv8 module was used as a baseline for comparison, as that is the module that this paper is proposing to replace and represents the minimum threshold that new modules should exceed to be determined viable.

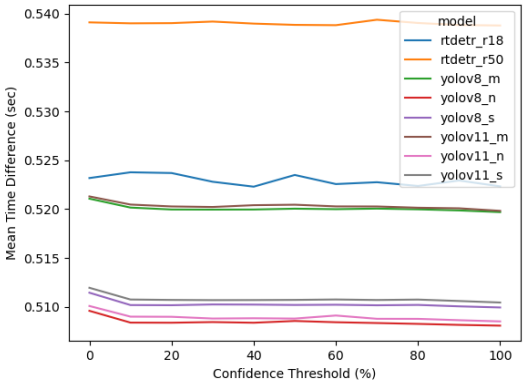

Figure 2: Time Between IUs

The first metric collected was the time between IUs in seconds. By logging the time elapsed between each new IU for a given model, it is possible to quantify that model's speed and compare it against the others. Interestingly, both RT-DETR models performed noticeably slower than the YOLO models, with the smaller 'r18' version barely keeping up with the YOLO 'm' models. Figure 2 shows the results of each model's timing at different thresholds.

The second metric collected was the number of detections for each given IU. The more detections each model can make at a given threshold, the better that model's utility at that threshold. Figure 3 shows a comparison of each model's detection abilities at different thresholds. After analysis, both RT-DETR models were able to produce more detections at each threshold than the YOLOv8 and YOLOv11 models, especially at confidence thresholds between 10-30%. At threshold values above 60% and below 10%, all models were nearly indifferentiable.

The third metric was a composition of the other two metrics and was the number of detections per second a model could produce. Assuming that some IUs might not produce any detections at all, the number of detections per second a model is capable of outputting should give an estimate for how often and how consistently the model will actually make detections at the given threshold [2]. Similar to

[2]It is important to note that a model being able to reliably produce detections does not mean that the detections produced are reliable. While the RT-DETR models were able to produce significantly more detections in the 10-20% threshold range

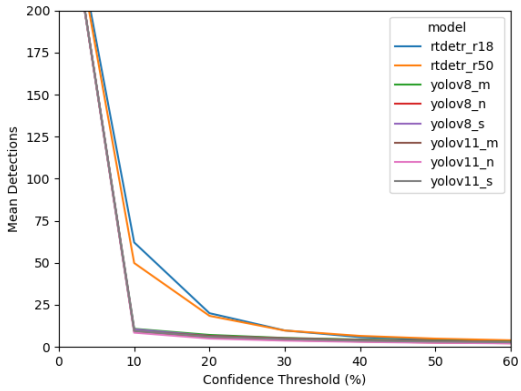

Figure 3: Detections per IU

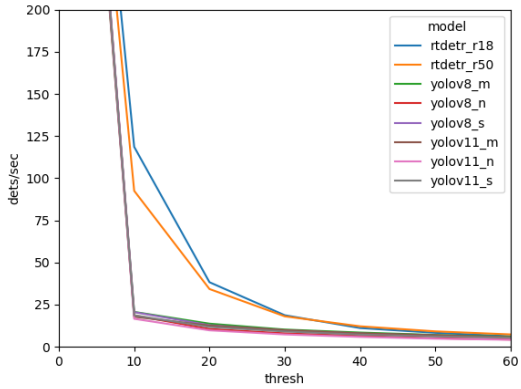

Figure 4: Detections per Second

the previous metric, both RT-DETR models were able to produce more detections per second than the YOLO models at low confidence thresholds, with the RT-DETR 'r50' model outperforming every other model at every threshold. Figure 4 shows a full comparison of the models.

4.1.3 Results

The data collected in experiment 1 indicates that the RT-DETR module, including the different model sizes available to it, is quantifiably slower than the YOLO modules as currently implemented. At the same time, however, the RT-DETR module was able to output more detections per IU and per second on average than the other modules, implying that there are still potential advantages to using it over the YOLOv8 and YOLOv11 options. The results suggest that the RT-DETR module can perform noticeably better in situations and applications that allow for more relaxed confidence thresholds. Additionally, it is consistently able to output more detections than the other modules at all thresholds, making it more suitable for all detection tasks when some speed can be sacrificed.

compared to the YOLO models, these additional detections are also only 10-20% confident.

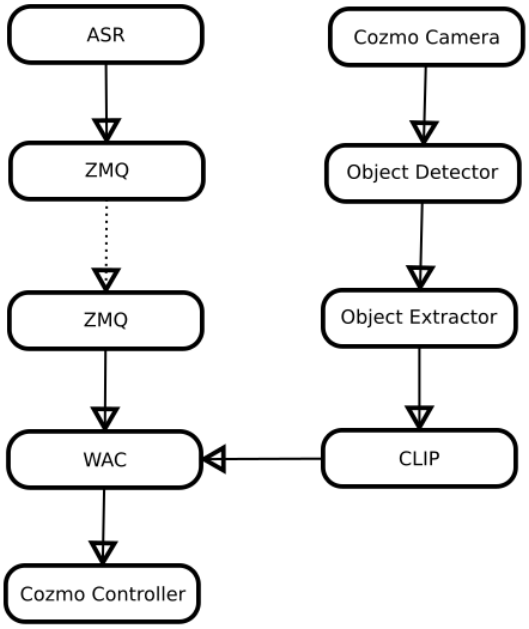

Figure 5: Symbol Grounding Pipeline

4.2 Experiment 2

4.2.1 Task & Procedure

Experiment 2 involved conducting a survey that utilized each object detection module in a more complex ReTiCo pipeline for a symbol grounding task in an attempt to determine which modules, if any, had a more positive impact on human-robot interactions. The pipeline consisted of a Microphone module, an ASR module, a custom module for communicating with Anki's Cozmo robot, an Object Extraction module, a Clip Object Features module, and a Words as Classifiers module. The ZMQ module was also used to distribute the computational cost across machines. A total of four participants took part in the survey. Figure 5 shows a visual representation of the pipeline that was used.

Participants were given a symbol grounding task requiring them to "teach" the Anki Cozmo robot the names of different objects, once for each of the three modules. The task involved positioning the Cozmo robot to face one of two objects and telling Cozmo the name of the object in view. The image from the Cozmo camera would then be fed into the pipeline and used to train the WAC model.

Figure 6: Symbol Grounding Environment

Once this action was performed for both objects, participants were instructed to perform inference using the pipeline, telling Cozmo what object to try and identify. Cozmo would then alternate between looking at both objects, indicating when it predicted the object in view as being the same type as instructed to look for. Once the task was done for each of the three modules, participants were then asked to rate the modules (using aliases to ensure ratings were blind) based on different attributes.

The survey was conducted in a single-blind manner, where participants did not know which object detection modules they were interacting with at any given moment. Due to time limitations, only one model size per module was tested during the survey. Both the YOLOv8 and YOLOv11 modules were set to use the 's' model size, with the RT-DETR module being set to use the 'r18' model size. These sizes were chosen to highlight each module's speed while retaining good precision. The survey environment used for experiment 2 is shown in figure 6.

4.2.2 Metrics & Baseline

Survey participants were asked to share their impressions of each module they interacted with (without knowing the true identity of each module) by ranking them on integer scales from -2 to 2 for responsiveness, quickness, and correctness. Additionally, participants were asked to indicate which module they thought contributed to the best interaction overall. While the correctness rating is straightforward to understand (being related to model precision), the difference between the responsiveness and quickness ratings is less straightforward. The purpose of asking for both responsiveness and quickness is that while quickness accounts for a module's perceived speed, it might not account for a module's perceived consistency. If a module can consistently output good predictions at a decent speed, it could be seen as more responsive than a module that inconsistently outputs good predictions at a fast speed. Thus the need for both responsiveness and quickness as ratings is deemed important and relevant. As previously, the collected data and metrics were compared against those gathered for the YOLOv8 module, as that serves as the baseline for what must be beaten to justify using other modules in new ReTiCo pipelines.

	Responsiveness	Quickness	Correctness	Votes for Best
RTDETR	6	3	6	2
YOLOv8	2	0	2	0
YOLOv11	4	5	5	2

Figure 7: Module Ratings Matrix

4.2.3 Results

While the survey sample size for this experiment was incredibly small and should be cause for caution, a pattern did start to appear showing preference towards both the RT-DETR and YOLOv11 modules in interactions. According to survey participants, the RT-DETR module showed more responsiveness and correctness on average than all other modules, with a middling quickness rating. The YOLOv11 module showed middling responsiveness and correctness slightly behind the RT-DETR module, but with the highest quickness rating by a more noticeable margin. YOLOv8 was rated significantly worse than both other modules on average. Figure 7 shows a full performance rating matrix constructed using the module ratings collected from survey participants. All individual module ratings were summed to produce the matrix, with higher values reflecting better performance in a given category and lower values reflecting worse performance. The RT-DETR and YOLOv11 modules were equally favored by participants as performing the best during interactions.

5 Conclusion

5.1 Discussion

The research done in this paper, motivated by the findings of Zhao et al. (Zhao et al., 2024), explored the efficacy of two new object detection modules for the ReTiCo framework: one based on the RT-DETR object detection model and the other based on the eleventh version of the YOLO object detection model. The performed experiments show that although the RT-DETR module (and to a lesser extent the YOLOv11 module) are quantifiably slower than the existing baseline (YOLOv8) module when tested on the COCO 2017 test dataset, they both performed qualitatively better than the baseline in a more complex system under the hardware limitations of the Anki Cozmo robot. While more survey data must be gathered before drawing stronger conclusions, the results presented in this paper show that at the cost of some quickness, the proposed RT-DETR module contributes more to the perception of better responsiveness and correctness in human-robot interactions than any of the other three modules. Similarly, the YOLOv11 module is shown to contribute the most to the perception of quicker interactions while maintaining similar responsiveness and nearly the same level of correctness as the RT-DETR module. Qualitatively, both the RT-DETR and YOLOv11 modules performed significantly better than the existing baseline and can justifiably be used over the YOLOv8 module in pipelines relying on the Anki Cozmo camera, or other similar hardware configurations.

5.2 Limitations

Although a clear pattern seems visible in the data shown in this paper, several limitations confine the provided results and could potentially alter the conclusions presented. The biggest limitation that must be mentioned is the scope of experiment 2, and the survey sample size. The sample size was extremely small with only 4 participants, meaning that the results as a whole are highly dependent on each individual response. Additionally, only one model size was tested for each module due to the small sample pool and restricted amount of time for each participant. Because of this, the inferences made are only representative of each respective module under the configurations used for experiment 2. Further testing needs to be done before conclusive inferences can be made regarding the modules under their full array of configurations. Experiment 2 also took place in one environment with static settings used for Cozmo's camera gain and exposure, leaving the possibility that the combination of environment and camera settings naturally favored specific models.

Finally, it is worth noting that there could be optimization differences between how the YOLO modules are implemented and how the RT-DETR module is implemented. Both YOLO modules utilize the Ultralytics Python package for handling inference, while the RT-DETR module is implemented using the Hugging Face Transformers package.

5.3 Future Work

Based on the issues presented under section 5.2, future work would involve a more comprehensive and thorough evaluation of each module with a focus on three core improvements to the experiments performed in this paper:

- Increasing the scope of experiment 2 to be more comprehensive, including testing more model sizes for each module. Modules would be evaluated under varying environments and the camera settings for Cozmo would be comprehensively tested rather than being statically set.
- Experiment 2 would be conducted for a longer period of time with significantly more participants. Significantly more data would be collected for accurate results, particularly if the scope of the experiment is performed with a wider range of model sizes and under differing environments.
- More time would be allocated towards investigating module optimization and the underlying implementation of the Ultralytics and Transformers libraries. Potential differences between module implementations could have an important impact on the conclusions drawn from both experiments.

References

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers.

Thilo Michael. 2020. Retico: An incremental framework for spoken dialogue systems. In *Proceedings of the 21th Annual Meeting of the Special Interest*

Group on Discourse and Dialogue, pages 49–52, 1st virtual meeting. Association for Computational Linguistics.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

Nicole Robinson, Brendan Tidd, Dylan Campbell, Dana Kulić, and Peter Corke. 2023. Robotic vision for human-robot interaction and collaboration: A survey and systematic review. *ACM Transactions on Human-Robot Interaction*, 12(1):1–66.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. 2024. Detrs beat yolos on real-time object detection.