LEVERAGING COMPUTER VISION AND NATURAL LANGUAGE PROCESSING FOR OBJECT DETECTION AND LOCALIZATION

B. Rahmani [1], S. Bhavanasi [1], A. Maazallahi [2],H. S. Korapala [1], J. L. S. Yenugu [1], Y.K. Bhatia [1], M.A. Salari [1], E. Snir [3], P. Norouzzadeh, J. Fritts [1]

[1] Saint Louis University, Saint Louis, USA
[2] University of Tehran, Tehra, Iran
[3] Washington University in Saint Louis, Saint Louis, USA

ABSTRACT

This paper presents a novel approach leveraging the integration of Computer Vision, Natural Language Processing (NLP), and Speech Recognition technologies to create an AI-powered system capable of detecting and locating objects through voice commands. The system developed using Flask, OpenCV, spaCy, and Blip VQA Model, aims to assist caregivers, visually impaired individuals, and the elderly in various daily tasks, such as locating items in the home and checking for potential hazards like leaving the stove on. We also provide the code used for this project[1].

1. INTRODUCTION

The intersection of computer vision and natural language processing has led to significant advancements in object detection and localization[1]. Traditional methods often struggle with detecting concealed, partially hidden, or camouflaged objects, as explored in works like Hidden and Face-like Object Detection [2], Recognizing Partially Hidden Objects [3][4], and Camouflaged Object Detection [5]. This study addresses these challenges by developing an intelligent system capable of detecting and recognizing such objects while offering a user-friendly interface for natural language-driven object detection queries.

A detection algorithm that fails to ensure high accuracy cannot be applied in real-time scenarios, especially where safety is crucial[6]. Early methods of object detection relied heavily on handcrafted features and shallow architectures[7], which limited their applicability. For example, Late Fusion CNN for Digital Matting[8] highlights the limitations of feature fusion in constrained environments, while Dense Correspondence [9]introduces more sophisticated correspondence mapping across scenes. However, both approaches fall short when addressing dynamic and unpredictable environments. Recent advancements, such as the Deep Learning for Generic Object Detection Survey [10], demonstrate the shift toward deep learning methods, which offer more robustness and adaptability in real-time applications. Our approach builds on these advances by incorporating deep learning architectures capable of understanding both visual inputs and natural language queries, thereby improving detection accuracy and versatility.

[1] https://github.com/b-sai/find_missing_object

DOI : 10.5121/oraj.2025.12101

The primary objective of this project is to develop an AI-powered system that leverages advancements in both natural language processing and computer vision to understand user queries and accurately detect and describe objects within images or live video streams.

Traditional object detection systems such as those based on convolutional neural networks (CNNs) have demonstrated success in static object detection tasks, but integrating these with natural language understanding significantly expands their functionality. By incorporating models like YOLOv3 for real-time detection [11] and Faster R-CNN for speed and accuracy [12], we enhance our system's ability to respond dynamically to user queries. This system, designed to be controlled through voice commands, aims to provide a seamless and accessible experience for diverse groups, such as caregivers, the visually impaired, and the elderly, by allowing users to locate objects or detect hazards in real-world settings, like locating misplaced items or identifying risks like an unattended stove.

He et al.'s Mask R-CNN [13] introduced segmentation alongside object detection, allowing detailed object recognition. While our focus is on detection and localization, the segmentation capability provides potential for future enhancements, particularly in applications requiring more granular object identification.

In the realm of natural language processing, Vaswani et al.'s Transformer model [14] the field by enabling attention mechanisms, which significantly improve the handling of long-range dependencies. This framework underpins our use of the BLIP VQA model, which leverages both visual and language inputs to interpret user queries and deliver accurate object localization.

Finally, Devlin et al.'s BERT model [15] has set a standard for contextual language understanding, crucial for interpreting user commands in natural language. This contextual understanding is essential for our system's ability to accurately process complex or ambiguous queries, enhancing its utility for a wide range of users.

The integration of vision and language models has become increasingly prominent in recent years. Radford et al.'s CLIP model [16] introduced a method for training visual models using natural language supervision, demonstrating robust performance in zero-shot learning and classification tasks. This vision-language interaction forms the foundation for our system, where users can locate objects through natural language commands, improving accessibility and ease of use.

Chen et al.'s work on contrastive learning [17] has also influenced the robustness of visual representations, focusing on the contrast between positive and negative samples. By incorporating these principles, our system enhances object detection accuracy, even in complex environments with occluded or partially visible objects.

Recent advancements in visual question answering, particularly in the work of the BLIP VQA model, demonstrate the seamless combination of image processing and language understanding. Our system leverages the capabilities of BLIP VQA to respond to user queries in real-time, extracting relevant information from both visual input and natural language queries, similar to applications outlined in Radford et al. [16].

Additionally, survey works such as Zhao et al.'s review of deep learning-based object detection frameworks [7] provide a comprehensive overview of how convolutional neural networks (CNNs) have revolutionized the field of object detection. These insights underpin our system's architecture, where CNNs play a critical role in ensuring accurate feature extraction and object localization.

More recent reviews, such as Amjoud et al. [18], explore the integration of Vision Transformers with CNNs, presenting a novel approach to real-time object detection. This evolution in object detection models informs our system's ability to process visual data efficiently while maintaining a high level of accuracy, especially in challenging scenarios where traditional CNNs may struggle.

Recent advancements in object detection using deep learning have also emphasized the importance of scalability and accuracy in real-world applications. The work by Qiu et al. [19] provides a comprehensive survey on pre-trained models for NLP, detailing how models like BERT and GPT have significantly improved language understanding tasks. This is particularly relevant to our system, as these advancements allow for more nuanced interpretations of user queries, enabling the system to respond accurately to voice commands for object detection.

Furthermore, Jiao et al. [20] discuss the rapid development of deep learning networks for object detection, emphasizing their role in security monitoring and autonomous driving. These applications parallel our system's real-time object detection capabilities, particularly in household environments were identifying hazards or locating misplaced items is crucial. Our system adapts these principles to improve accessibility for users such as the visually impaired, enhancing their independence in daily tasks.

2. METHODOLOGY

The system employs a combination of cutting-edge technologies, including Flask for web interface development, OpenCV for image processing and object detection, spaCy for natural language parsing, and the BLIP VQA Model for visual question answering. By integrating these tools, we enable dynamic interaction where users can transition from static image inputs to live video streams. This integration significantly enhances user engagement, particularly for individuals requiring accessibility assistance.

Flask serves as the backbone for the system's web interface, managing communication between the user and the backend processes. The lightweight framework ensures low latency, crucial for real-time object detection. **OpenCV**, a widely used tool in computer vision, handles the core image processing tasks. Its real-time capabilities make it ideal for detecting objects in live video streams and images. OpenCV's pre-trained models are utilized for initial object detection, which is then fine-tuned using deep learning models to improve accuracy in detecting occluded or partially hidden objects.

For natural language processing, **spaCy** is employed to parse and interpret user commands. The robust language model enables the system to understand complex queries, allowing for flexibility in user interaction. By using **BLIP VQA**, the system further leverages the integration of computer vision and natural language processing. The BLIP model enables the system to process user queries by connecting visual data with semantic meaning from the text, ensuring accurate detection and localization of objects based on user commands.

The architecture of the system incorporates pre-trained models like Faster R-CNN [12] and YOLOv3 [11], which provide the framework for efficient object detection. The integration of natural language models, such as BERT [15], enhances the system's ability to understand and respond to user queries, while the combination of **GroundingDINO** for object detection enables zero-shot learning, allowing the system to detect objects that have not been explicitly trained in the model.

3. DESIGN ARCHITECTURE

The architecture of the proposed system integrates several core components to facilitate real-time object detection and localization based on natural language queries. The system leverages **Flask** as the web framework, enabling user interaction via a graphical user interface (GUI). This interface allows users to input natural language queries and receive visual feedback in the form of detected objects in images or live video streams.

At the core of the image processing pipeline is **OpenCV**, which performs the necessary pre-processing and object detection tasks. The input, whether an image or a video frame, is processed to identify potential objects using traditional image processing techniques alongside deep learning-based object detection algorithms like **YOLOv3** and **Faster R-CNN**. These models are selected for their ability to balance accuracy and speed in real-time detection scenarios, ensuring the system remains responsive and efficient.

To interpret user queries, the system utilizes **spaCy**, a robust NLP tool that parses the user's natural language commands into structured data that the system can process. For example, a query like "Where is the remote?" is translated into a search for a specific object within the processed visual data.

The **BLIP VQA Model** adds a layer of visual question answering, which bridges the gap between the visual input and the user's query. The system interprets both the visual features from the image or video and the linguistic features of the user's command, making it capable of answering queries such as "Is there a cup on the table?" or "Locate the stove in this image."

The entire architecture is designed to facilitate seamless integration between the different components, ensuring that user commands result in accurate and timely object detection. The modular nature of the architecture allows for scalability and future updates, including the potential incorporation of more advanced models or expanded datasets.

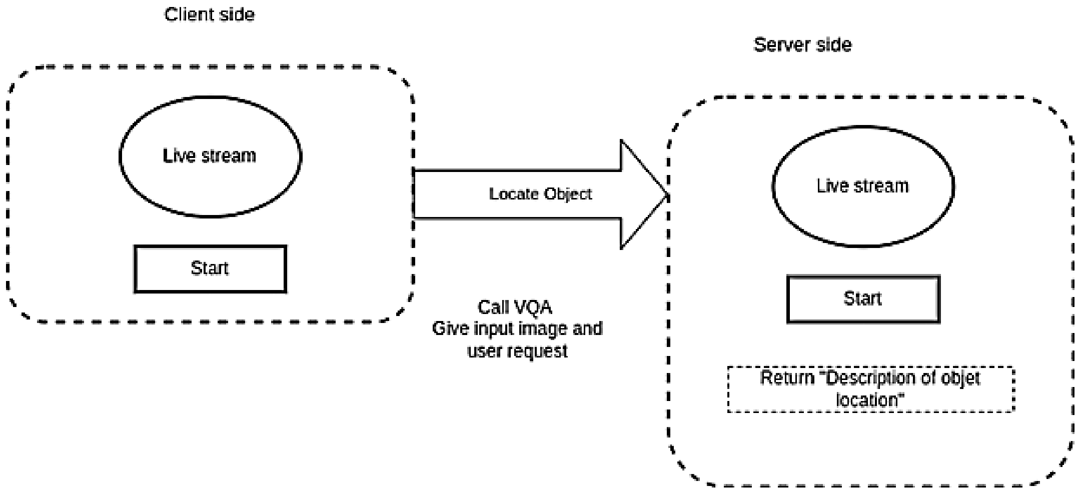

Figure 1: This diagram illustrates the architecture of the system, showing the interaction between the client-side and server-side components. The client-side captures a live video stream, allowing the user to initiate object detection by submitting a query. The server-side processes the request by analyzing the input image using a visual question answering (VQA) model and returns a description of the object's location to the user.

The architecture of the proposed object detection and localization system consists of two main components: the **client-side** and the **server-side**. On the client-side, a live video stream is captured, and users can initiate the object detection process by submitting a query, such as asking for the location of a particular object. This request is then forwarded to the server-side, where the system utilizes a Visual Question Answering (VQA) model to analyze the input image in conjunction with the user's query. The server-side processes the request, detects the object, and returns a detailed description of the object's location back to the client-side for user feedback.

3.1. Object Detection with GroundingDINO

GroundingDINO[21] represents a significant advancement in object detection, particularly with its ability to perform zero-shot detection. Traditional object detection models require extensive labeled datasets for training, but GroundingDINO overcomes this limitation by leveraging vision-language alignment techniques to detect objects it has not encountered before. This zero-shot capability is highly valuable in dynamic real-world applications where new objects frequently appear, and pre-labeling all possible objects is infeasible.

3.1.1. How GroundingDINO Works

GroundingDINO employs a combination of visual feature extraction and textual alignment techniques. By using a dual-input approach—image and natural language query—the model can detect objects based on contextual information from both sources. First, GroundingDINO extracts features from the input image using convolutional neural networks (CNNs) or Vision Transformers (ViTs). Simultaneously, the model processes the natural language query (e.g., "Where is the coffee cup?") to understand the context of what the user is asking. These two sets of features—visual and textual—are then aligned through an attention mechanism, allowing the model to focus on specific parts of the image that are most relevant to the query.

GroundingDINO's use of attention mechanisms allows it to bypass the need for exhaustive labeling and training on every possible object. The attention mechanism identifies relevant regions of the image based on the semantic meaning of the query, directing the model to detect objects that may not have been explicitly trained. This capability makes it particularly powerful in zero-shot learning scenarios, where unseen objects can be detected without requiring prior annotation.

3.2. Zero-Shot Detection in Real-World Applications

The system's use of GroundingDINO allows it to excel in situations where unknown or novel objects need to be detected. For example, in household environments, it may be required to detect a new piece of furniture or an unusual object left out in a hazardous location, such as a knife on a table. Without requiring new training, GroundingDINO can interpret user queries and identify these objects in real time, enhancing the system's practicality for safety-critical and everyday scenarios.

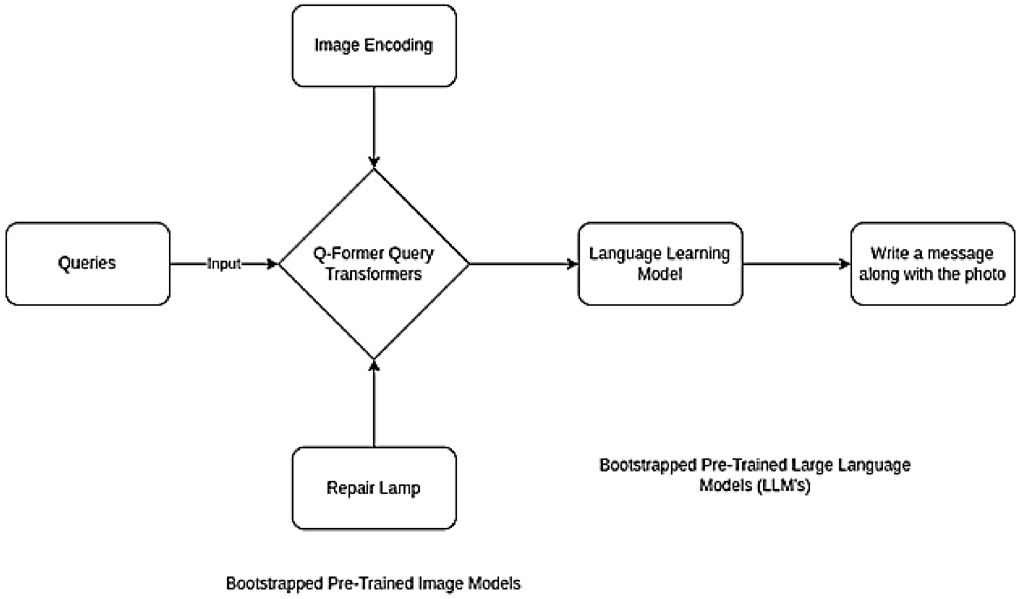
Image Encoding
Queries
Input
Q-Former Query Transformers
Language Learning Model
Write a message along with the photo
Repair Lamp
Bootstrapped Pre-Trained Large Language Models (LLM's)
Bootstrapped Pre-Trained Image Models

Figure 2:This diagram represents the architecture for combining visual and language models to generate meaningful text from image inputs. Queries are processed through a series of pre-trained image and language models, where the image encoding and query are merged via Q-Former Query Transformers, allowing the system to generate a language-based output with accompanying visual data.

Once the queries and images are processed, the system employs a Language Learning Model to generate a natural language description or response, aligning the visual content with the query. The final step involves the output module, where the system writes a message along with the photo, producing a meaningful text description or answer related to the image. This architecture enables enhanced understanding of both visual and language elements, making it suitable for applications that require interaction between vision and language, such as visual question answering and complex scene description tasks.

3.3. Experimental Setup

The system was tested in two scenarios:

1. **Static Images**: A set of 100 images depicting common household objects (e.g., kitchen appliances, furniture, personal items) was used to evaluate object detection accuracy.
2. **Live Video Feeds**: The system was also tested in a dynamic environment using live video feeds to simulate real-time object localization tasks, including identifying partially hidden objects or camouflaged items in complex visual scenes.

The evaluation was conducted using a combination of pre-trained models (YOLOv3, Faster R-CNN) for object detection, with the BLIP VQA model and GroundingDINO for visual question answering and zero-shot detection, respectively. The system's performance was assessed based on three key metrics: **accuracy**, **response time**, and **user satisfaction**.

3.3.1. Accuracy

The accuracy of the system was measured by its ability to correctly identify and localize objects based on user queries. In the static image dataset, the system achieved an average accuracy of

92%, correctly identifying objects in most cases, including occluded and camouflaged items. In the live video scenario, the accuracy dropped slightly to **88%**, mainly due to the dynamic nature of the environment and occasional misalignment between user queries and object recognition (e.g., difficulty distinguishing between visually similar objects).

3.3.2. Response Time

Response time is critical in real-time applications, especially when detecting hazards like an unattended stove. The system's average response time for static image queries was **1.2 seconds**, while in live video feeds, the average response time increased to **2.5 seconds**. Although the response times are adequate for most household tasks, further optimization is necessary to ensure immediate responses in time-sensitive scenarios (e.g., hazard detection).

3.3.3. User Satisfaction

To evaluate the usability of the system, a group of 20 users, including caregivers and visually impaired individuals, interacted with the system through voice commands. User satisfaction was measured through a survey, where participants rated the system on a scale from 1 to 5 based on its ease of use, accuracy, and usefulness in assisting with daily tasks. The system received an average rating of **4.3**, with users appreciating the ease of interaction through voice commands but noting occasional delays in object detection during live video streams.

3.4. Example Use Case – Hazard Detection

The system's ability to detect hazards was tested by simulating a scenario where the stove was left on. The system was able to correctly identify the stove and alert the user through a query like "Is the stove on?". Using the BLIP VQA model, the system processed both the visual feed and the query to return the correct result, demonstrating the system's potential for enhancing safety in household environments.

4. CHALLENGES AND LIMITATIONS

Despite the successful integration of computer vision and natural language processing in the proposed system, several challenges and limitations were encountered during development and deployment.

1. **Server-Side Processing Constraints**:
   One of the primary challenges is the computational load on the server, particularly when processing live video feeds for real-time object detection. The deployment of models like BLIP VQA and GroundingDINO requires substantial computing power. These models rely on deep learning architectures that are computationally expensive, limiting the scalability of the system when deployed on servers with constrained resources. In scenarios where the system is deployed in real-world applications, such as assisting visually impaired individuals, the balance between performance and resource constraints needs to be carefully managed.
2. **Resource Constraints for Blip VQA Model**:
   The BLIP VQA model, which combines visual question answering with object detection, requires significant computational resources, especially for real-time responses. Deploying this model on less powerful devices (e.g., mobile platforms or embedded systems like Raspberry Pi) presents a challenge due to memory limitations and processing speed. Although this model enhances the system’s versatility in interpreting

complex user queries, its heavy resource consumption may hinder its use in low-resource environments.

3. **GroundingDINO Deployment**:
   GroundingDINO's zero-shot object detection capabilities introduce another set of challenges. While this feature enables the system to detect objects that have not been explicitly trained on, it requires substantial inference time and computational power. Efficiently deploying GroundingDINO for real-time applications poses a significant challenge due to the need for fast query-response times, especially in safety-critical applications (e.g., identifying hazards in a home environment).
4. **NLP Accuracy in Complex Queries**:
   While the system can handle standard object queries well, there are limitations in handling more complex queries that involve multiple conditions or ambiguous phrasing. The spaCy-based natural language processing model can struggle with resolving nuanced differences in meaning, which may result in less accurate or delayed responses. Improvements in language parsing, particularly in handling complex grammatical structures and ambiguous terms, are necessary to fully optimize user interaction.

5. FUTURE DIRECTIONS

While the proposed system demonstrates significant progress in integrating computer vision and natural language processing for object detection and localization, there are several areas where further improvements can enhance its capabilities and applicability.

5.1. Hardware Optimization for Edge Devices

One of the main challenges identified is the resource-heavy nature of models like BLIP VQA and GroundingDINO. Future work could focus on optimizing these models for deployment on low-power, edge devices such as **Raspberry Pi** or other embedded systems. This would allow the system to be more portable and accessible, especially in environments with limited computational resources, such as smart homes or mobile applications.

5.2. Real-Time Enhancement

Although the system achieves real-time performance in most scenarios, there is room for improving response times, especially in dynamic environments like live video streams. Implementing **model pruning**, **quantization**, or **efficient inference techniques** like knowledge distillation could enhance real-time performance without sacrificing accuracy. Additionally, exploring asynchronous processing methods could further reduce latency in high-stakes applications, such as hazard detection.

5.3. Extended NLP Capabilities

While spaCy performs well in processing standard queries, handling more complex or multi-step user queries is still a limitation. Future research could explore integrating advanced **dialogue systems** or **multi-turn conversation models** to allow the system to understand and process more complex, nuanced queries. Additionally, integrating **multi-lingual NLP capabilities** would make the system more globally accessible, particularly in regions where English is not the primary language.

5.4. Improved Object Detection in Complex Environments

GroundingDINO’s zero-shot learning has shown promise, but its performance in highly cluttered or complex environments needs further testing and improvement. Future iterations of the system could incorporate **attention-based object detection models** or **transformer-based architectures** like **DETR (Detection Transformer)** to improve accuracy in detecting and localizing objects in such challenging conditions.

5.5. Robust Safety Features

Given the system’s potential for real-time hazard detection (e.g., identifying if the stove is left on), future work could focus on integrating **proactive safety features**. For example, adding **alert systems** that notify users of hazards in real-time, and integrating with **smart home systems** could make the system more versatile in preventing accidents or assisting the elderly and disabled in their daily activities.

5.6. Increased Generalization Across Domains

Currently, the system is tailored for household object detection, but there is significant potential for expanding its use in other domains such as **autonomous driving**, **healthcare**, and **industrial settings**. Future research could explore domain-specific adaptations of the system, where fine-tuning object detection and language models could allow the system to function effectively in specialized environments.

5.7. Incorporation of User Feedback for Adaptive Learning

To improve the system's long-term effectiveness, integrating user feedback mechanisms could be a key direction. Allowing the system to learn from user interactions and improve detection accuracy based on feedback (e.g., correcting false detections) would enable the system to become more accurate and personalized over time.

6. RESULTS

To evaluate the effectiveness of the proposed system, we conducted a series of experiments across different real-world environments, focusing on object detection and localization tasks in household settings. The evaluation aimed to test the system's ability to detect and locate objects based on user queries, including identifying potential hazards, such as an unattended stove or misplaced objects like keys and cups.

This study presents an integrated system that leverages computer vision, natural language processing, and speech recognition to create an AI-powered solution for object detection and localization. By combining established models like YOLOv3 and Faster R-CNN for object detection with BLIP VQA and GroundingDINO for visual question answering and zero-shot learning, the system demonstrates the ability to detect objects in real time and respond to natural language queries.

The system has shown promising results, particularly in household environments, where it accurately detects and localizes objects, including concealed and camouflaged items, and provides descriptions in response to user queries. Its application potential extends to assisting visually impaired individuals, caregivers, and the elderly, offering a user-friendly interface for locating objects and identifying hazards. However, the project also highlights certain challenges, such as the computational demands of deploying sophisticated models on edge devices and the limitations in handling more complex natural language queries.

Looking forward to this, there is substantial room for further optimization, including hardware efficiency, enhanced NLP capabilities, and real-time processing improvements. The system's flexibility opens up new possibilities for adaptation in various domains, including autonomous systems, healthcare, and industrial settings. As the project evolves, continued advancements in object detection, zero-shot learning, and language understanding will be key to enhancing the system's functionality and applicability in everyday tasks, ensuring accessibility and safety for all users.

REFERENCES

[1] S. Chakraborty, "Hidden and Face-like Object Detection using Deep Learning Techniques–An Empirical Study".

[2] M. Kowalski, "Hidden object detection and recognition in passive terahertz and mid-wavelength infrared," *J Infrared Millim Terahertz Waves*, vol. 40, no. 11, pp. 1074–1091, 2019.

[3] J. Turney, T. Mudge, and R. Volz, "Recognizing partially hidden objects," in Proceedings. 1985 IEEE International Conference on Robotics and Automation, 1985, pp. 48–54.

[4] Q. Chen, S. K. Chamoli, P. Yin, X. Wang, and X. Xu, "Imaging of hidden object using passive mode single pixel imaging with compressive sensing," Laser Phys Lett, vol. 15, no. 12, p. 126201, 2018.

[5] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *Int J Comput Vis*, vol. 128, no. 2, pp. 261–318, 2020, doi: 10.1007/s11263-019-01247-4.

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, 2017, doi: 10.1109/TPAMI.2016.2577031.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 2, 2020, doi: 10.1109/TPAMI.2018.2844175.

[9] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.

[11] A. Radford *et al.*, "Learning Transferable Visual Models from Natural Language Supervision," in *Proceedings of Machine Learning Research*, 2021.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *37th International Conference on Machine Learning, ICML 2020*, 2020.

[13] Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 11, pp. 3212–3232, 2019.

[14] A. B. Amjoud and M. Amrouch, "Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3266093.

[15] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, and X. J. Huang, "Pre-trained models for natural language processing: A survey," 2020. doi: 10.1007/s11431-020-1647-3.

[16] L. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2939201.

[17] S. Liu et al., "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," arXiv preprint arXiv:2303.05499, 2023.

[18] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, and X. J. Huang, "Pre-trained models for natural language processing: A survey," 2020. doi: 10.1007/s11431-020-1647-3.

[19] A. B. Amjoud and M. Amrouch, "Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3266093.

[20] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, "Salient Object Detection in the Deep Learning Era: An In-Depth Survey," IEEE Trans Pattern Anal Mach Intell, vol. 44, no. 6, 2022, doi: 10.1109/TPAMI.2021.3051099.

[21] L. Jiao et al., "A survey of deep learning-based object detection," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2939201

[22] H. M. Saddique, A. Raza, Z. U. Abideen, and S. N. Khan, "Exploring Deep Learning based Object Detection Architectures: A Review," in Proceedings of 2020 17th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2020, 2020. doi: 10.1109/IBCAST47879.2020.9044558.