Pathway to Perception: A Smart Navigation Approach for Visually Impaired Individuals Leveraging YOLO, Faster R-CNN, and LLaMA

Tahsin Ashrafee Susmit[1], Isratul Hasan[1], Maliha Mehejabin[1],
Azmain Ibn Kausar[1], Suraiya Binte Akbar[1], Dr. Golam Rabiul Alam[1], MD. Saiful Islam[1]

[1]*Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh*

{tahsin.ashrafee.susmit, isratul.hasan, maliha.mehejabin, azmain.ibn.kausar,
suraiya.binte.akbar, rabiul.alam, md.saiful.islam}@g.bracu.ac.bd

***Abstract*—The purpose of our study is to create new technology that will provide a revolutionary navigation system with significant improvement of mobility and independence for visually impaired people. We utilize YOLOv11 and Faster R-CNN to detect an object which is used in combination with Llama 3.2–3B Instruct for context-aware navigation by providing helpful guidance of our current essential location. Our paper tackles the failure points in today's technologies with lack of flexibility for dynamic and unfamiliar environments, unreliable performance under changes in lighting conditions and inefficient obstacle detection. By training these models together and selecting the one with the highest confidence score, we enhance spatial awareness, identifying obstacles in key areas like the left, right, or center. This approach, complemented by personalized navigation instructions, ensures improved decision-making and safety in real-world scenarios. Using advanced locational technologies available today and imagining those of tomorrow, we aspire to render current navigation methods obsolete by fostering more efficient, real-time and autonomous tools for visually impaired people as they become part of the familiar or unfamiliar environments. After fine-tuning the Llama 3.2-3B-Instruct model, BLEU-4 increased from 0.0442 to 0.1175, and ROUGE-L improved from 0.2102 to 0.3204, indicating enhanced text generation fluency and coherence.**

***Index Terms*—YOLOv11, Faster R-CNN, Llama 3.2-3B Instruct, Object Detection, Navigation System, Visually Impaired, Location Detection.**

I. Introduction

Advancement in technology and our focus on inclusivity have made assistive devices invaluable for improving the daily quality of life for people with disabilities. Using the most recent studies, the World Health Organization (WHO) estimates that the number of people with visual impairment is 285 million. Of these, 246 million have low vision and 39 million are estimated to be fully blind [1]. Although assistive technology and medical treatments have improved, many visually impaired individuals continue to encounter difficulties in navigation without some support from others, especially in more challenging and unfamiliar situations. Their dependence affects their independence and mobility. Few existing navigation systems offer sufficient support, often proving to be too adaptive and not functional in real-time. Current systems face challenges in dealing with varying landscapes and uneven areas. One of the main limitations in current navigation technologies is that they combine object recognition and customized suggestions in ways that are not always effective, making everyday use difficult [2]. We are now using YOLOv11 with Faster R-CNN to assist the visually impaired individuals in a better way. All models are trained together, and the one with better confidence score is selected. Which allows us to know if some obstacles are on the left, right or center. This object detection and spatial awareness combination enables better navigation decisions. We have added Llama 3.2-3B Instruct, an improved version of the Large Language Model (LLM) to increase flexibility in dynamic environments with look-up tables for codebook bindings. This potentially has an effect on the system decision-making which in turn increases its ability to deploy these types of smart interconnected systems.

II. Related works

Smart navigation systems for visually impaired people have made attention over the years. Technology has made major progress, with advancement in object detection models and (LLMs)-Large Language Models enabling new approaches to enhance navigation assistance. For instance:

The paper, "Embedded Implementation of an Obstacle Detection System for Blind and Visually Impaired Persons' Assistance Navigation", proposes a system using an improved version of the YOLO v5 neural network to solve navigation problems encountered by individuals with visual impairment. By integrating DenseNet in YOLO structure and improving the full-network both speed wise as well detection 6 accuracy, this system is able to run at 43 frames per second (FPS) and achieve an impressive accuracy of 83.42%. The pruning and quantization of the channels were used to make it possible as embedded implementation in a ZCU 102 board's system. The authors point out limitations such as the difficulty of detecting a set of diverse obstacles within dynamic-cluttered scenes which might affect the system's general robustness and adaptability [3].

In the paper, You Only Look Once: Unified, Real-Time Object Detection, the author doesn't follow traditional methods which often rely on classifiers for detection tasks. When implemented for object detection framework YOLO attempts to frame the problem of detecting objects in images as a single regression process that predicts bounding boxes and class probabilities directly from full image in one evaluation. By this new approach, a single neural network runs at full speed on the images in 45 frames per second for the entire base model and up to 155 frames per second with Fast YOLO. Although the architecture of YOLO enables end-to-end optimization that is directly associated with detection performance, it produces more localization errors than methods like R-CNNs. Additionally, YOLO presents a lower false positive rate on non-object regions than the conventional object detection algorithms which makes it more practical [4].

In the paper Real Time Object Detection using YOLO Algorithm, authors made use of the You Only Look Once (YOLO) method for object detection and checked its efficiency compared to the leading models in speed and performance. Whereas traditional algorithms might scan regions of an image through multiple forward and backward propagations, YOLO architecture performs a single evaluation that first predicts bounding boxes and associated class probabilities from features maps using logistic regression itself. This has the benefit of speeding up the detection process drastically which can be applied to real-time applications. Thus the research correctly showcases benefits of YOLO which is quick and accurate, thereby proving it to be a good alternative solution in scenarios like autonomous navigation etc. However, it limits its effectiveness and needs more work to enhance the robustness of YOLO under a general case like localization error in complex environments [5].

In the paper "An Improved Faster R-CNN for Small Object Detection", authors introduce methods to overcome difficulties in detecting small objects under complex scenes by using convolution neural networks(CNN). This paper proposes a refined Faster R-CNN based algorithm for small object detection. In this method, a twostage detection strategy is used. Improved loss function based on Intersection over Union (IoU) is introduced for bounding box regression during the positioning stage. In the recognition stage, multiscale convolutional feature fusion helps to supplement the feature map with more information and a modified Non-Maximum Suppression (NMS) algorithm is utilized in order to reduce overlap object losses. The results show that in the (0, 32], proposed method has a recall rate as high as 90% and an accuracy rate of up to 87%, which is far better than the original Faster R-CNN. The effectiveness of these enhancements motivates the further study on object detection frameworks and provides us a valuable way to consider handling small objects for them [6].

In the paper "A Closer Look at Faster R-CNN for Vehicle Detection", the authors investigate the application of the Faster R-CNN algorithm to vehicle detection, noting its initial unimpressive performance when directly applied to large vehicle datasets. After much trial and error, the authors detail their study on model architecture as well as tweaks to parameters and algorithms. Due to their modification the model performance gets greatly enhanced; the most competitive result is achieved on the KITTI vehicle dataset. However, using parameter tuning to reach for this additional performance may also indicate difficulties in achieving robust performance across diverse datasets without the level of customization needed [7].

In their paper "Intelligent LiDAR Navigation: Leveraging External Information and Semantic Maps with LLM as Copilot", Xie Zhang Schwertfeger introduce a new take on robot navigation which combines Large Language Models (LLMs) with traditional occupancy grid maps and laser-based sensing approaches. The research aims to improve robotic navigation systems with some latent contextual understanding similar to human cognition by exploiting osmAG and a fabulous topometric hierarchical map representation. With this integration, robots can leverage external information and experiential knowledge from requests to other robotic services like elevator maintenance updates for better navigation efficiency. But as useful and receptive as these maps are, a big hindrance is that the need to use it for the robot to get anywhere which brings up potential obstacles when in real-world situations where a rather dated map can be practically worthless. Authors argue that addressing these problems by using LLMs in path planning can prevent the system from being too careful, and ensure all available passages are recognized to improve navigation results [8].

III. Dataset Overview

This section outlines the datasets used in our work, how they were prepared, and the methodologies employed for annotating and enhancing object detection for the visually impaired. For object detection and localization, our dataset is made up of two major sources: the MSCOCO dataset and a primary dataset with object classes that are crucial for navigation. Also, We created two separate datasets from scratch and compiled them together for fine-tuning LLaMA 3.2-3B Instruct to generate navigational sentences.

A. *MSCOCO Dataset*

The MSCOCO dataset is a widely recognized dataset for object detection. It comprises 80 object classes. Three of the classes were removed for this paper as they are not relevant to the task of guiding a visually impaired person. This dataset provided around 116081 training and 4900 validation images, each with annotated bounding boxes and object locations for multiple objects.

B. *Primary Dataset*

To be more precise and enhance the model's capability we introduced 10 new classes. They are, Pole, Zebra Crossing, Pedestrian Green Light, Pedestrian Red Light, Red Traffic Lights, Yellow Traffic Lights, Manhole, Stairs and Bus stop. We collect nearly 11490 training images and 2878 validation images from different resources and annotated them manually with the help of roboflow.

C. VQA Dataset

We created two separate datasets on our own and compiled them together for fine-tuning LLaMA 3.2-3B Instruct to generate navigational sentences. These datasets are crucial to help the model learn how to tackle navigating through real-world spaces. The datasets are: Navigating Sentence Generation Dataset with 948 entries and Description Generation Dataset with 2,269 entries.

1) Navigating Sentence Generation Dataset: The Navigating Sentence Generation Dataset is targeted for training the model to provide real-time navigational instruction. The instructions rely on object detection in the world, as well as the poses of objects relative to each other (e.g., people; zebra crossings, pedestrian signals etc).

Dataset structure:
Instruction: Directs the model to generate a navigational sentence.
Input: A description of the scene, including objects and their spatial positions (e.g., left, right, center).
Output: The corresponding navigation command, which directs the user on what actions to take based on the input.

2) Description Generation Dataset: The Description Generation Dataset focuses on training the model to output elaborate descriptions regarding either objects or interactions in the environment. The navigating sentence dataset just gives guidance, whereas this one describes objects and what users can do with them.

Dataset Structure:
Instruction: Directs the model to generate a description.
Input: A detailed description of the environment, including objects and their spatial positions.
Output: The generated description of the object interactions, often focusing on how the user can interact with the object.

IV. THE PROPOSED METHODOLOGY

The goal of this research was to create a method of navigational guidance for the blind people by objectifying classes and localization using an ensemble model of YOLOv11, Faster R-CNN and then from class and location LLaMa will generate a navigational sentence and this sentence will be converted into audio through GTTS.

A. YOLOv11 for Object Detection

Initially, we trained YOLOv11 on the MSCOCO dataset, but it struggled with critical classes like zebra crossings and stairs, which were absent. To address this, we created a primary dataset of a decent amount of images with 10 key classes then trained YOLOV11 on that dataset. After the processing steps of YOLOV11 like image Preprocessing, feature extraction from 53 convolutional layer, Anchor free bounding box regression it give the output of classified objects name with a bounding box and a confident score [9] [10]

B. Faster R-CNN for Object Detection

We also trained Faster R-CNN with the primary dataset and changed the layers to get more fine result. Here after feature extraction we used PCA (Principle Component Analysis) to reduce the amount of feature and selected smaller but the most useful features which is efficient for the model [11] [12].

C. Ensembling YOLOv11 and Faster R-CNN

For the 10 classes defined by us, accuracy is very critical as any error could lead to dangerous situations. For example, misclassifying a red pedestrian light as a red traffic light could cause someone to cross the road unsafely. To overcome this drawback and ensure reliability, we trained both YOLOv11 and Faster R-CNN on the same dataset and used an ensembling method to select the best object detection and localization. To ensure our output detections are clean and precise, the best way to obtain this is to remove duplicates. For this reason, Intersection-over-Union (IoU) calculation is done. If two detections have an IoU score that shows high overlap between the pair of detections, then one is removed. The detection with the lower confidence score is removed. After such calculations, our data has only the non-overlapping detections between both models with the higher confidence detection. We kept the threshold IoU value at 0.5 to detect overlap.

IoU Calculation:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{i}$$

D. Post-Processing and Combining Detections

After getting the localized objects along with confidence scores for each objects from YOLOV11 trained with MSCOCO Dataset and from the ensembling of YOLOV11 and Faster R-CNN with our primary dataset we convert it into a consistent format and normalize based on the image width. Using the normalized center, objects are classified into regions: the left region if the normalized x-center is less than or equal to 0.33, the center region if it falls between 0.33 and 0.66, and the right region if it exceeds 0.66. As each object is classified into these regions, our system counts how many objects of each type are found in each regions of the image and creating a summary that works as a input for the finetuned Llama that will generate the navigating or descriptive sentence [13].

Left Region:

$$\text{Region} = \text{Left}, \quad \text{if} \quad x_{\text{center}} \leq 0.33 \tag{ii}$$

Center Region:

$$\text{Region} = \text{Center}, \quad \text{if} \quad 0.33 < x_{\text{center}} \leq 0.66 \tag{iii}$$

Right Region:

$$\text{Region} = \text{Right}, \quad \text{if} \quad x_{\text{center}} > 0.66 \tag{iv}$$

E. Fine tuning LLaMA-3.2-3B-Instruct

After that, we finetuned LLaMA-3.2-3B-Instruct using Low-Rank Adaptation (LoRA) to generate both navigating and descriptive sentences from detected localized objects [14]. LoRA applies low-rank decomposition matrices to the attention layers of our model and instead of updating the full weights, this model inserts additional matrices that are lower in rank. These matrices are responsible for catching the task or context-specific adaptations of a model and can update during the fine-tuning process without the need for full re-training or a substantial degree of memory usage [15] [16] [17].

F. Text-to-Speech conversation using gTTS

Finally, when we had the navigation sentences generated by detected objects and spatial locations, they were with Google Text to Speech (gTTS) in text form converted into audio representation. Google Text-to-Speech is by far the most reliable and efficient way to synthesize text into human-like speech necessary for guiding visually impaired individuals [18].

V. EXPERIMENTAL RESULTS ANALYSIS

The experimental findings underscore the efficacy of our method in producing navigation instruction for visually impaired individuals.

A. Experimental Setup

Specifications for YOLOv11 and Faster R-CNN Training

- **Processor:** AMD Ryzen 9 5950X 16-Core
- **GPU:** NVIDIA GeForce RTX 3080 Ti with 12 GB GDDR6X Memory
- **RAM:** 64 GB

Specifications for Fine-tuning LLM

- **GPU:** NVIDIA A100 with 40 GB HBM2 Memory
- **RAM:** 84 GB System Memory

B. Result Analysis

The following part evaluates the experimental findings and assesses the model's performance using metrics such as BLEU, ROUGE, F1 score, Precision, and Recall. By providing information on the effects of fine-tuning and dataset modifications, these metrics aid in evaluating the model's object identification accuracy and the quality of navigation directions.

This table uses important metrics including Precision, Recall, F1 score, and mean Average Precision (mAP) at two IoU thresholds (mAP50 and mAP50-95) to evaluate the performance of YOLOv11 with Faster-RCNN across several datasets. With a precision of 0.9849, recall of 0.9862, and an F1 score of 0.9855, the Faster R-CNN on the primary dataset performs the best overall, exhibiting outstanding precision across all criteria. With an accuracy of 0.9620, recall of 0.9116, and an F1 score of 0.9361, YOLOv11 likewise does well on the primary dataset, but marginally worse than F-RCNN. It continues to provide good results, nevertheless, especially in mAP, earning 0.9422 for mAP50 and 0.7706 for mAP50-95. However, when evaluated using the COCO dataset, YOLOv11 demonstrates a significant drop in efficiency, with accuracy of 0.6275, recall of 0.4803, and an F1 score of 0.5442.

TABLE I
MODEL PERFORMANCE METRICS OF YOLOV11 AND FASTER-RCNN

Model	Precision	Recall	F1 Score	mAP50	mAP50-95
F-RCNN (Primary Dataset)	0.9849	0.9862	0.9855	-	-
YOLOv11 (Primary Dataset)	0.9620	0.9116	0.9361	0.9422	0.7706
YoloV11(Coco Dataset)	0.6373	0.4649	0.5376	0.5115	0.3610

Following fine-tuning, the Llama3.2-3B-Instruct model's BLEU and ROUGE scores show an impressive rise in performance. At first, the model's BLEU scores (BLEU-1 was 0.1858 and BLEU-4 was 0.0442) showed that it was less accurate at producing meaningful and contextually correct phrases. After fine-tuning, however, BLEU-1 rose to 0.2982 and BLEU-4 improved to 0.1175, indicating improved sentence construction and word choice accuracy. Significant improvements were also observed in the ROUGE scores, which evaluate recall and the overlap between produced and reference material. The model's ROUGE-1, ROUGE-2, and ROUGE-L scores were 0.2572, 0.0827, and 0.2102, respectively, prior to fine-tuning. These scores increased to 0.3668 for ROUGE-1, 0.1618 for ROUGE-2, and 0.3204 for ROUGE-L after fine-tuning, demonstrating a significant improvement in the model's ability to match sentence structure and content with the reference, hence raising the produced text's general quality.

TABLE II
LLAMA3.2-3B-INSTRUCT BLEU SCORE

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Llama3.2-3B-Instruct(Before Finetune)	0.1858	0.1053	0.066	0.0442
Llama3.2-3B-Instruct (After Finetune)	0.2982	0.196	0.1473	0.1175

TABLE III
LLAMA3.2-3B-INSTRUCT ROUGE SCORE

Model	ROUGE-1	ROUGE-2	ROUGE-L
Llama3.2-3B-Instruct (Before Finetune)	0.2572144958	0.08267532265	0.2102495145
Llama3.2-3B-Instruct (After Finetune)	0.3668390408	0.1617590308	0.3204431557

C. Implications and Results

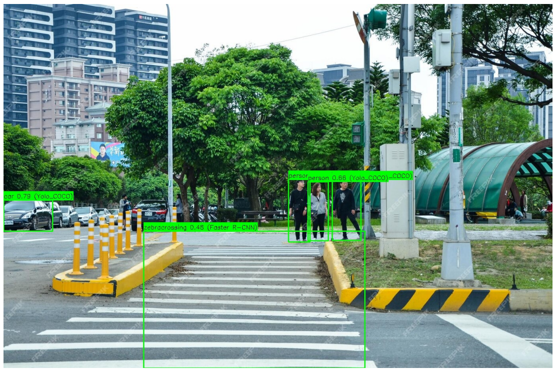

Fig. 1. Output of YOLO and Faster R-CNN Model for Zebracrossing

TABLE IV
GENERATED SENTENCE FOR ZEBRACROSSING FROM LLAMA

Detection from Yolo and Faster RCNN	1 car on the left, 1 zebracrossing on the center, 3 person on the center.
Response from LLAMA	You can cross safely at the zebra crossing on the center.

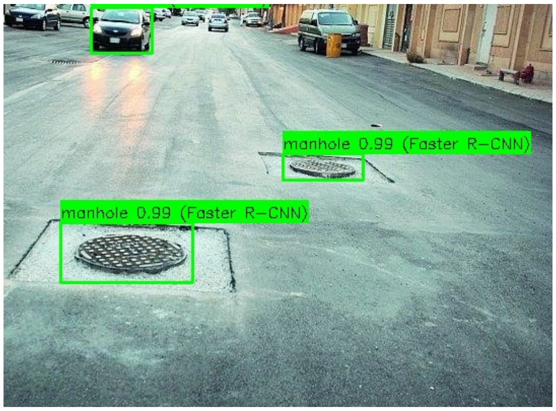

Fig. 2. Output of YOLO and Faster R-CNN Model for Manhole without Person

TABLE V
GENERATED SENTENCE FOR MANHOLE WITHOUT PERSON FROM LLAMA

Detection from Yolo and Faster RCNN	1 manhole on the left, 1 car on the left, 1 manhole on the center.
Response from LLAMA	A manhole is on your left and another manhole is in front of you so, stay right.

VI. DISCUSSION

YOLOv11, Faster R-CNN, and LLaMA 3.2-3B Instruct are used in our innovative navigation system for the blind and visually handicapped. In contrast to earlier models, our approach offers exact position, spatial information, and auditory source analysis for improved navigation in addition to obstacle detection. YOLOv11 enables quick object recognition, while Faster R-CNN increases the precision of categorization. Context-aware navigation instructions are produced by LLaMA 3.2-3B, allowing for secure and secure mobility. The clear, accurate, and dependable instructions provided by our system greatly increase user autonomy and mobility in contrast to models, which provided inaccurate navigation assistance.

VII. CONCLUSION AND FUTURE WORK

In our study, we made a navigation system for the visually impaired people by using object detection models YOLOv11, Faster R-CNN with Llama 3.2-3B Instruct as well. This combination greatly provides the obstacle detection and awareness that people need to better move through their environment without any human assistance. The fine-tuned Llama 3.2–3B Instruct model achieved competitive performance metrics, a BLEU-4 score of 0.1175 and ROUGE-L of 0.3204. Integrating YOLOv11 helped in fast detection of obstacles, which is an important factor as timely decisions need to be taken in dynamic environments. Faster R-CNN improves this accuracy of class based recognition and hence delivers dependable information to the users about its surroundings. This capability is essential for enhancing user confidence and safety while navigating complex and unfamiliar environments. Our results show the promise of such a combination to provide effective and practical solutions for visually impaired individuals. This work is particularly well placed to highlight how AI technologies are revolutionizing improving navigation accuracy, cost constraint and user confidence for a better living experience in persons who are blind or visually impaired.

Furthermore, to maximize technology needs and make the whole thing more portable, we also want to create lighter versions of the models. Last but not least, we intend to increase training in a variety of environmental settings and strive toward creating a unique navigation tool that can include distance measurement and object identification for even higher accuracy. Our goal is to develop a wearable gadget that can be worn in the face as well as has a camera that can identify things. It will also be able to produce and return audible text to the user.

References

[1] W. H. Organization, "Who releases new global estimates on visual impairment," 2012. Accessed: Oct. 22, 2024.

[2] M. A. Rahman, S. Siddika, M. A. Al-Baky, and M. J. Mia, "An automated navigation system for blind people," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 201–212, 2022.

[3] A. B. Atitallah, Y. Said, M. A. B. Atitallah, and et al., "Embedded implementation of an obstacle detection system for blind and visually impaired persons' assistance navigation," *Computers and Electrical Engineering*, vol. 108, p. 108714, 2023.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[5] I. V. S. L. Haritha, M. Harshini, S. Patil, and J. Philip, "Real time object detection using yolo algorithm," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1465–1468, 2022.

[6] C. Cao and et al., "An improved faster r-cnn for small object detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019.

[7] Q. Fan, L. Brown, and J. Smith, "A closer look at faster r-cnn for vehicle detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 124–129, 2016.

[8] F. Xie, J. Zhang, and S. Schwertfeger, "Intelligent lidar navigation: Leveraging external information and semantic maps with llm as copilot," *arXiv*, 2024.

[9] T. Delleji, F. Slimeni, H. Fekih, A. Jarray, W. Boughanmi, A. Kallel, and Z. Chtourou, "An upgraded-yolo with object augmentation: Mini-uav detection under low-visibility conditions by improving deep neural networks," in *Operations Research Forum*, vol. 3, p. 60, Springer, 2022.

[10] W. Feng, Y. Zhu, J. Zheng, and H. Wang, "Embedded yolo: A real-time object detector for small intelligent trajectory cars," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 6555513, 2021.

[11] A. F. Gad and J. Skelton, "Faster r-cnn explained for object detection tasks," 2024. Accessed: Oct. 22, 2024.

[12] T. Grel, "Region of interest pooling explained," February 2017. Accessed: Oct. 22, 2024.

[13] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021. Accessed: Oct. 22, 2024.

[15] "Llama 3.2: Handling both text and images," 2024. Accessed: Oct. 22, 2024.

[16] H. Touvron and et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023. Accessed: Oct. 22, 2024.

[17] R. Zhang and et al., "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv preprint arXiv:2303.16199*, 2023. Accessed: Oct. 22, 2024.

[18] C. Van Lieshout and W. Cardoso, "Google translate as a tool for self-directed language learning," 2022.