Evaluation of Vision-Language Models

Sumanth Manduru
G01380318
George Mason University
smanduru@gmu.edu

Swathi Guptha
G01393328
George Mason University
sguptha@gmu.edu

Krishna Preetham Rachakonda
G01386104
George Mason University
krachak@gmu.edu

Abstract

The vision-language models (VLM) are trained on extensive image-language datasets, but there's uncertainty regarding whether these models effectively ground objects based on their corresponding text inputs. Our study focuses on assessing the performance of these models. With particular attention to CLIP and BLIP. We evaluate Detic, based on CLIP, for object detection and recognition tasks using both PascalVOC and custom datasets. Furthermore, we assess the VQA downstream task on BLIP architecture on the fine-grained datasets. Our approach involves leveraging parameter-efficient fine-tuning for the BLIP VQA model. To assess the grounding capability, we evaluated the VLM model across various datasets. This examination aimed to determine if additional text input enhances object identification efficiency. Our experiments led us to conclude that the V-L model performs well with generalized images. However, its efficacy diminishes significantly with domain-specific datasets. Furthermore, fine-tuning the model for specific downstream domains proves to be challenging. Results indicate a twofold improvement compared to zero shot performance

1. Introduction

Vision-Language involves merging natural language processing (NLP) and computer vision methods to help machines grasp, create, and convey information using both text and images. This combination of AI fields allows vision language models to explain and produce descriptions for pictures, videos, and settings, linking visual and textual data. This interdisciplinary technique has exciting potential in various areas like describing images and answering visual questions, as well as creating content and developing tools for visually impaired individuals, pushing AI to better understand and communicate with more complexity. By training models to understand both visuals and text at the same time, this method has seen impressive success in various tasks that involve analyzing and creating content from different kinds of data. By exposing these models to many examples of paired visual and textual data during pre-training, they learn complex connections between images and their descriptions, allowing them to generate detailed and contextually relevant responses. This advancement not only pushes forward the field of natural language processing and computer vision but also greatly enhances the abilities of machines to grasp and communicate in a world with multiple modes of input, opening doors for more advanced applications in areas like understanding images, creating content, and developing tools for people with sensory disabilities.

Vision-Language Pre-training, models have been demonstrating impressive capabilities across a spectrum of tasks that involve both images and text. In tasks like Visual Question Answering (VQA) and visual reasoning, the machine excels at understanding images and providing accurate answers to questions about them. Additionally, it demonstrates proficiency in tasks like image captioning and image-text retrieval by generating captions for images and retrieving relevant images based on textual queries. Moreover, VLP models excel in visual grounding, which involves associating specific textual descriptions with corresponding regions in an image. Additionally, they can even generate images from textual descriptions, showcasing their versatility in tasks like text-to-image generation. In the realm of computer vision, these models tackle traditional tasks like image classification, object detection, and segmentation as variations of vision-language problems.

In our project, we aim to assess the performance of Vision-Language Pre-training (VLP) models, particularly focusing on CLIP (Contrastive Language-Image Pre-Training) and BLIP (Bootstrapping Language-Image Pre-training). To evaluate CLIP, we utilize a model called Detic, which is based on CLIP, for object detection and recognition tasks using the PascalVOC dataset as well as our own custom dataset. Through various experiments, we analyze the performance of Detic in different scenarios and evaluate its accuracy and robustness in object detection and recognition. Additionally, we evaluate BLIP VQA architecture on fine-grained datasets. We start by conducting zero-shot

evaluations to understand its performance without any fine-tuning and then proceed to fine-tune the model on the selected datasets. We employ a range of evaluation metrics commonly used in both vision and natural language processing domains to ensure a comprehensive assessment and obtain accurate inferences from the models. Through these evaluations, we aim to gain insights into the capabilities and limitations of CLIP and BLIP models across various tasks and datasets, contributing to the advancement of vision-language understanding.

2. Related Work

CLIP [7], which stands for Contrastive Language-Image Pre-training, marks a major advancement in integrating vision and language understanding. Created by OpenAI, CLIP functions on a transformer-based design and is trained to connect images with their written descriptions through contrastive learning. Unlike conventional models that concentrate solely on images, CLIP utilizes a dual encoder framework where images and text undergo separate encoding processes, with interaction between modalities achieved through a straightforward cosine similarity calculation of their feature vectors. This architecture proves successful in tasks like image retrieval, and when scaled up, it can even establish a robust image encoder through extensive contrastive pre-training. This extensive pre-training also enables CLIP to encode both modalities into a shared space, allowing it to perform tasks like image classification, object detection, without requiring task-specific fine-tuning. However, CLIP does have its limitations. Because it lacks sophisticated multimodal fusion mechanisms, CLIP exhibits subpar performance in tasks like Visual Question Answering (VQA) and visual reasoning. It demands significant computational resources for training and inference and may struggle with grasping nuanced contextual understanding in images and text, leading to occasional inaccuracies. Additionally, biases in the large scale pre-training data could affect its performance on certain tasks or datasets. Addressing these limitations is crucial for improving the effectiveness of CLIP and reliability in practical applications.

BLIP [6] Image Captioning is a big leap forward in how we add captions to images. It mixes together different methods to make captions more accurate and detailed. Unlike older ways that only focus on either the picture or the words, BLIP uses both. It starts by looking at the picture to pick out possible words, then it polishes them up with a language model to make sure they're just right. It even cleans up any extra stuff in the picture that might distract from the caption. All this careful work means we get captions that are really spot on, which is a big deal for image captioning.

BLIP Visual Question Answering (VQA), which takes things up a notch by allowing computers to understand and respond to questions about images. Just like with BLIP Image Captioning, it uses a mix of visual features and language processing to come up with answers. So, if you ask it a question like "What color is the car?" about an image, it can analyze the picture and give you the right answer like Blue. BLIP VQA opens up exciting possibilities for all sorts of applications, from helping visually impaired individuals to enhancing search engines and beyond.

3. Datasets

In the evaluation of Vision Language Models, we have selected two downstream tasks: Object Recognition and Visual Question Answering (VQA). Here, we will provide descriptions of the datasets utilized for each specific task.

3.1. Object Recognition

For Object Recognition, we have utilized two datasets, outlined as follows:

3.1.1 Pascal-VOC-2012

The evaluation of the vision-language model on object recognition involved utilizing the standard dataset known as Pascal VOC 2012 [1]. With every image in this dataset, there is an associated XML file that provides comprehensive details about the bounding box, class, and, a few times segmentation of the object. As shown in 2, the dataset consists of 20 different classes.Pixel-level annotations from the XML files were ignored because the task was centered on object detection and recognition.In order to evaluate the DETIC [12] model, a total of 17,125 validation images were used for the assessment. While the annotations were initially provided in XML format, DETIC utilizes Detectron2 [10] to register the dataset, resulting in a conversion to the COCO format. Furthermore, it is important to take into account that dimensions of each image in the validation dataset vary from (300, 500) to (400, 500). Due to this, the dataloader set up for evaluation was limited to processing batches of size 1.

3.1.2 Custom Dataset

To address the limitations of the Pascal VOC 2012 dataset, a custom dataset was curated by capturing images around the campus using a standard iPhone camera. This dataset was designed to cover classes that the model might find challenging, classes that did not perform well on Pascal VOC, and regular classes. A total of 114 pictures in the .HEIC format were gathered. The original dimensions of each image were (4032 * 3024), but they were resized to (800 * 800) in order to comply with the input size requirement of the DETIC model [12].

Following the data formatting step, the images underwent annotation using a tool called Roboflow. In total, 76

Figure 1. Visualization of BLIP model's predictions using GradCAM for ZeroShot learning, showcasing its performance on the color attribute-based question.

- *Person:* person
- *Animal:* bird, cat, cow, dog, horse, sheep
- *Vehicle:* aeroplane, bicycle, boat, bus, car, motorbike, train
- *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor

Figure 2. Class distribution of PascalVOC-2012 dataset [1]

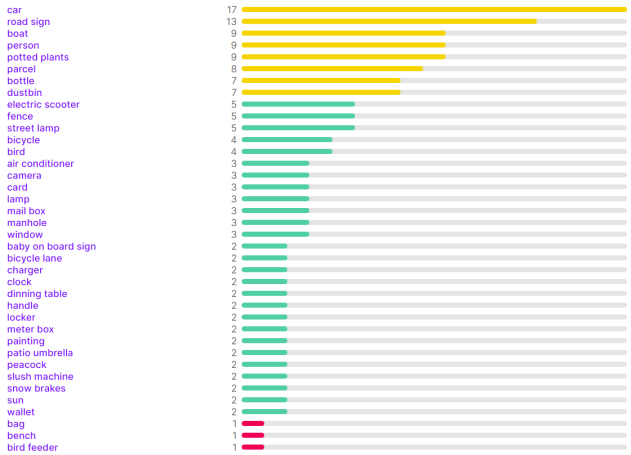


Figure 3. Selected classes from the custom dataset along with their distributions

classes were identified during the annotation process. Additionally, as Fig 3 illustrates, there are not many pictures for each class. The generalized training approach of DETIC Model, which asserts to be able to perform object detection and recognition without specific prior training data, is the reason for the large number of classes in the images. The purpose of this sparse class distribution was to evaluate the generalizability of the model.

3.2. Visual Question Answering

During the evaluation of Vision Language models for Visual Question Answering, we utilize the Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset [9]. This dataset is an upgraded version of the CUB-200 dataset, boasting a doubled amount of images per category and additional annotations for part locations. It comprises 200 categories, totaling 11,788 images, with each image annotated with 15 part locations, 312 binary attributes, and 1 bounding box. The dataset includes a comprehensive set of attributes describing various visual characteristics of birds, ranging from colors and patterns to shapes and sizes. Among the attributes listed, our focus lies on color, shape, and pattern. Colors encompass a spectrum including Blue, Brown, Iridescent, Purple, Rufous, Grey, Yellow, Olive, Green, Pink, Orange, Black, White, Red, and Buff. Similarly, for shape and pattern attributes, we consider values like Solid, Spotted, Striped, and Multi-colored for patterns, and Dagger, Hooked, Needle, Hooked seabird, and Spatulate for shape attributes.

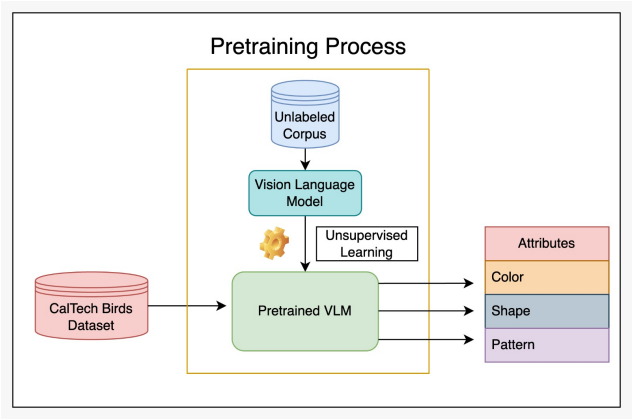


Figure 4. Zeroshot Technical Approach

We established a pipeline to prepare a Visual Question Answering (VQA) dataset using the Caltech-UCSD Birds-200-2011 (CUB-200-2011) dataset. The pipeline prepares the data by filtering attributes, constructing a DataFrame, and extracting bounding box information to facilitate subsequent analyses. During preprocessing, images are cropped based on bounding boxes and resized to a target shape of (224, 224, 3). We define a custom PyTorch dataset class,

VQADataset, which loads an image, crops it according to its bounding box, resizes it, and constructs a textual question based on the queried attribute. We obtained approximately 45,000 samples each for color and shape attributes, while for the pattern attribute, we acquired around 65,000 samples. These samples underwent an 80-20 train-test split for training purposes. This comprehensive pipeline laid the groundwork for training a VQA model, facilitating the processing and fusion of both image and text inputs for visual question answering tasks. The question follows the format: **What is the {attribute_part} of the bird?**.

4. Approach

In this section, we will outline the methodology and approach employed for each task, providing detailed insights into our strategies and techniques.

4.1. Object Recognition

Object detection comprises two main tasks: Object Localization and Object Classification. Traditional approaches tightly integrate both tasks, leading to annotation-intensive processes. This limitation restricts object detection tasks to a maximum of 1000 classes in datasets like LVIS [2], unlike the classification tasks on ImageNet [8], which supported 21k classes a decade ago. The DETIC paper addresses this challenge by decoupling localization and classification sub-problems. It primarily focuses on the classification sub-problem, training the classifier using image-level labels to expand the vocabulary of detector. This open-vocabulary detector is achieved by replacing classification weights (W) with fixed language embeddings of class names obtained using CLIP embeddings [7]. By utilizing CLIP text embeddings, the proposed detector (Fig 5) can be transferred to new datasets and vocabularies without the need for fine-tuning. The pre-trained DETIC model [12] is applied for evaluation on both the Pascal VOC 2012 and custom datasets. To adapt the model to an evaluation dataset, text embeddings for all classes in the evaluation dataset are generated using a pre-trained CLIP model. The analysis was conducted using the evaluation code provided in the GitHub repository of DETIC model. The pascal-voc-2012 dataset is evaluated on threshold of 0.8 and the custom dataset is evaluated on the threshold of 0.6.

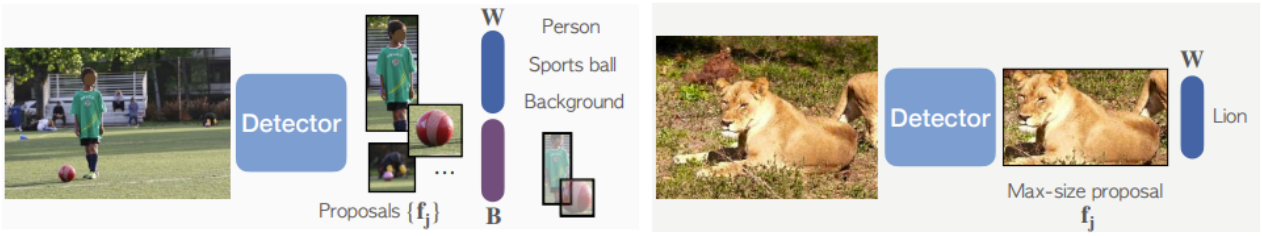
4.2. Visual Question Answering

In the VQA task, our examination focused on the pre-trained BLIP base version, considering three primary characteristics: color, shape, and pattern. For color-related inquiries, we categorized various body parts such as Forehead, Wing, Belly, and Leg together, while Eye, Crown, and Primary formed another distinct category. The characteristic values encompass a range of colors including Blue, Brown, Iridescent, and others. The rationale behind categorizing colors was based on our belief that certain body parts are inherently more recognizable than others. For instance, features like eyes, crowns, and the entire body are typically easier to discern due to their distinct visual characteristics. Eyes, in particular, stand out as they often possess clear and recognizable features. Crowns, although they may vary in appearance, generally have distinguishable shapes that aid in recognition. Conversely, body parts like the forehead, wings, and belly present greater challenges for recognition. Foreheads may lack distinctiveness compared to other features, while wings exhibit varying shapes and sizes across species. Additionally, the belly tends to be less visually prominent and its recognition may rely more on contextual cues. By categorizing colors in this manner, we aimed to evaluate the ability of Visual Language Model (VLM) to recognize and respond to different levels of visual complexity within the context of color-based inquiries. This approach allowed us to assess the performance of model across a spectrum of visual characteristics, shedding light on its strengths and limitations in visual understanding tasks. Similarly, We followed a systematic procedure to assess the performance of BLIP in discerning shape and pattern attributes.

Surprisingly, our ZeroShot inference on the Caltech dataset yielded notably low exact accuracies, with approximately 20% for all attributes: 22.14% for color, 19.67% for shape, and 18.54% for pattern. Grad-CAM, short for Gradient-weighted Class Activation Mapping, is a technique employed to elucidate and interpret the decisions made by deep learning models. It offers a visual representation of the pivotal regions within an input image crucial for the prediction of model. By calculating the gradients of the classification score pertaining to a specific class relative to the feature maps of the final convolutional layer, Grad-CAM [11] facilitates comprehension of the model decision-making process. In our examination, Figure 1, we observed that the inquiry "What is the belly color of the bird?" not only emphasizes the belly of the bird but also its beak, resulting in diminished informativeness and impact. Consequently, this observation motivated us to pursue fine-tuning of the BLIP Model specifically on the Caltech Bird dataset.

5. Results

We conducted a series of experiments to evaluate the Visual Language Model (VLM) across two distinct tasks: Visual Question Answering (VQA) and Object Recognition. These experiments aimed to gauge the effectiveness and capabilities of the Vision Language model in comprehending and responding within the context of visual understanding tasks.



(a) Detection data (b) Image-labeled data

Figure 5. Approach overview of the DETIC model [12]

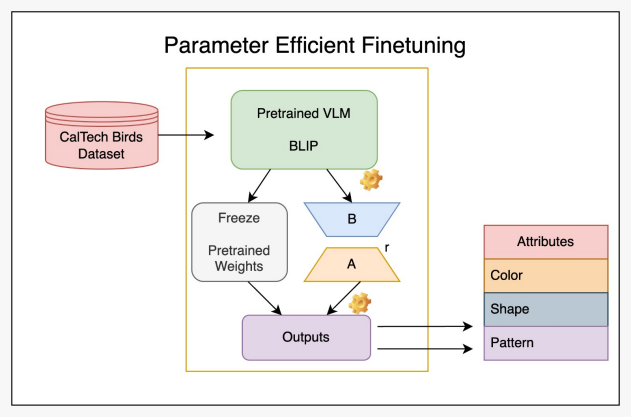


Figure 6. Parameter Efficient Finetuning Approach - Low Rank Adaptation (LoRA)

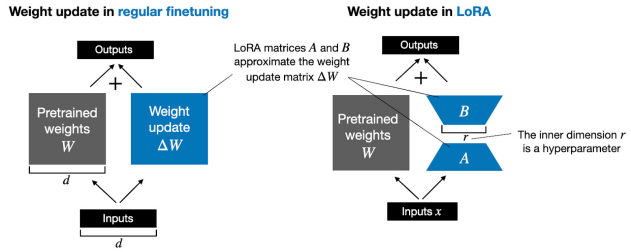


Figure 7. Difference between regular finetuning and PEFT Finetuning (LoRA).

5.1. Object Recognition

The results of the Object Recognition downstream task will be discussed in the subsequent sections.

5.1.1 Pascal-VOC-2012

As Table 1 shows, the DETIC model performs better at identifying large objects than small ones. In 8, this tendency is demonstrated by the detection of the object when it is big enough but undetected when it is too small. In object recognition, the model demonstrates satisfactory performance in identifying classes such as person, cow, sheep, bird, dog, airplane, and cat 9. However, it struggles when these objects are densely packed in an image, making detection challenging. Moreover, classes like pottedplant, tvmonitor, and diningtable exhibit poor performance, which may be caused by imprecise class definitions affecting the alignment between CLIP-generated text embeddings and object detection-generated image embeddings. This could be because CLIP generates text embeddings for each class in a generic way; the generated embedding does not have to correspond to that particular category (e.g., vintage vs. modern cars). The embeddings in the two subcategories will be very different. Another possibility is the image clarity; a network may find it challenging to extract features such as trains since the images are hardly readable with the human eye. The results showcased in the table **??** indicate that the DETIC model demonstrates superior performance when detecting larger-sized objects compared to smaller ones.

Dataset	APs	APm	APl	AP
Pascal-voc	11.97	28.61	**46.40**	38.94
Custom Dataset	nan	3.932	**12.15**	10.91

Table 1. Evaluation of DETIC model on object detection

5.1.2 Custom Dataset

The custom dataset includes common objects as well as objects that Pascal VOC detects poorly. Since the DETIC model requires input images to have a size of (800, 800) minimum, incorporating these classes might enhance performance. The model performs noticeably poorly in regular classes, as Table 1 illustrates its precision in object detection. The overall performance of the model seems satisfactory for some classes based on the classification results 11. Notably, classes that were previously missed in Pascal VOC—such as potted plants, dining tables, TVs, monitors, cars, and boats, are now being correctly identified 12. This indicates that the model heavily relies on the CLIP embeddings interpretation of the class. If a class is unknown to the CLIP model, the DETIC model will likely fail. Additionally, image clarity plays a significant role in detect-

Figure 8. In the pascal-voc dataset, the DETIC model effectively detects objects of significant size but struggles to recognize them when they are small

Figure 9. The DETIC model is effectively detecting few of the pascal-voc dataset images

category	AP	category	AP	category	AP
person	60.956	bird	53.833	cat	82.505
cow	69.117	dog	75.846	horse	37.994
sheep	60.260	aeroplane	47.936	bicycle	43.573
boat	41.114	bus	11.678	car	0.990
motorbike	55.118	train	0.000	bottle	28.837
chair	35.833	diningtable	17.247	pottedplant	0.000
sofa	56.142	tvmonitor	0.000		

Figure 10. Evaluation of DETIC model on pascal-voc dataset for object recognition

ing classes like cars and boat. In contrast to the classes found in Pascal VOC, the model is able to identify new classes 9. Usually, these classes are only detectable after the model has been trained on relevant input images. On the other hand, DETIC allows these classes to be classified without requiring prior training. Examples of such new classes in custom dataset includes sun, road sign, patio umbrella, wireless controller, bench, golf club, clock, and various other objects. However, some classes are being incorrectly interpreted. For instance, the class "graduation gown" is misinterpreted in a few images, as depicted in 15. While the open vocabulary setup may work well for certain classes, it can also result in misclassification. The DETIC model performs well in general object detection tasks, but performs quite poorly when encountered with unknown objects. It must be fine-tuned using domain-specific data in order to improve its performance in particular domains. If this fine-tuning is not done, misclassifications may occur, which could have serious consequences, especially in fields where accuracy is crucial.

5.2. Visual Question Answering

During the fine-tuning process of the BLIP model, we focused on optimizing the Language Model (LM) loss, which serves as a crucial objective function for training. To achieve this, we utilized the AdamW optimizer with a learning rate of $2e^{-5}$ and a weight decay of 0.05. In our experimentation, we explored the effectiveness of different learning rate schedulers to dynamically adjust the learning rate during training. Specifically, we tested two schedulers: the Cosine Annealing Learning rate scheduler with a maximum number of epochs (T_max) set to 10, and the Exponential Learning rate scheduler with a decay factor (gamma) of 0.9. We conducted training for a total of $\{10, 15\}$ epochs, with a patience value of 5 for early stopping to prevent overfitting. Additionally, to enhance training stability and efficiency, we employed gradient scaling using

category	AP	category	AP	category	AP
handle	0.000	vacuum cleaner	0.000	toilet cleaner	0.000
camera	0.000	water meter	0.000	hair clip	0.000
charger	0.000	ghost	0.000	dustbin	8.911
sketch	0.000	chandlier	0.000	painting	12.624
fire place	0.000	moon	0.000	window	0.000
bird	13.168	ceiling fan	50.000	locker	0.000
coffee machine	0.000	fire hydrant	0.000	wallet	40.396
dinning table	5.050	snow brakes	0.000	mail box	0.000
bicycle lane	0.000	boat	13.304	parcel	0.000
fire alarm	0.000	street lamp	8.317	pipe	0.000
person	28.175	golf club	40.000	tv	0.000
baby on board sign	0.000	lip balm	0.000	bench	35.000
patio umbrella	6.733	detergent	0.000	potted plants	48.133
sign	0.000	racket	0.000	graduation gown	20.000
manhole	39.967	cat	0.000	sun	0.000
tree roots	0.000	train	0.000	pen	0.000
toast machine	0.000	electric scooter	34.604	clock	0.000
slush machine	0.000	cooker	0.000	bottle	24.858
bicycle	64.356	wireless controller	60.000	road sign	4.256
meter box	15.149	hanger	80.000	bag	0.000
stool	0.000	monitor	0.000	twig	0.000
peacock	0.000	air conditioner	20.198	fence	0.000
laptop	50.000	socket	0.000	card	0.000
lamp	20.198	car	6.238	spoon	0.000
hose pipe	0.000	coins	0.000	bird feeder	80.000
cap	0.000				

Figure 11. Evaluation of DETIC model on custom dataset for object recognition

the GradScaler. These training details were chosen and experimented with to optimize the performance of the BLIP model in downstream tasks. Figure 7 showcases the disparity between regular fine-tuning and Parameter-Efficient Fine-Tuning (PEFT), while Figure 6 portrays the adaptation of LoRA [3] to the VQA use case.

Let us say If the weight matrix, W contains 7B parameters, then the weight update matrix ΔW also contains 7B parameters, and computing the matrix ΔW can be very compute and memory intensive. The decision to utilize Low Rank Adaptation (LoRA) stems from the sheer size of BLIP, boasting 387,031,868 parameters. Training such an extensive model is computationally prohibitive. By selecting a mere 0.6%, approximately 2,359,296 parameters, we circumvent this issue. Each epoch of training these parameters spans approximately 20 minutes. Let us delve into the configuration of LoRA. We have opted for 16 attention heads i.e rank, a number carefully chosen to avoid memory issues on the GPU cluster. Our choice of lora_alpha, set to 32, dictates the scaling factor, crucial for fine-tuning. Additionally, we apply dropout to LoRA attention scores with a 0.05 probability, aiding in regularization to prevent overfitting. Furthermore, LoRA offers numerous benefits. Notably, it mitigates catastrophic forgetting, a phenomenon wherein LLMs lose previously acquired knowledge during fine-tuning. Moreover, LoRA facilitates superior generalization to out-of-domain scenarios. Its streamlined fine-tuning process, targeting only a fraction of model parameters, significantly expedites training compared to full fine-tuning approaches.

In our evaluation process, we employed several metrics to assess the performance of the model. Each of these metrics offers unique insights into the performance of the model, enabling a comprehensive assessment of its capabilities across different dimensions of evaluation. **Exact Accuracy:** This straightforward metric evaluates the exact string match between the model's predicted answer and the ground truth. However, it may be overly strict since it does not consider semantically similar answers. **Wu-Palmer Similarity (WUPS):** This metric gauges the semantic similarity between the predicted answer and the ground truth by leveraging WordNet to quantify the distance within the semantic tree. This metric generates a score ranging from 0 to 1, with 1.0 signifying a flawless alignment between the predicted and actual answers in terms of their semantic meaning. **Levenshtein Distance (LD):** This metric calculates the edit distance between the predicted and reference answers, providing insight into the degree of dissimilarity while normalizing for answer length. It quantifies the similarity between two strings by counting the minimum number of single-character edits required to transform one into the other. From Table 2, it is evident that fine-tuning results in a twofold increase from the zero-shot performance.

Attributes	WUPS	Accuracy	LD
color	48.97	43.47	2.67
shape	43.35	37.2	2.91
pattern	38.62	36.9	3.67

Table 2. Evaluation Metrics on different attributes of the Bird

6. Resources

We will share the resources that assisted us in completing the project.

1. LAVIS for GradCAM - LAVIS, a Vision Language Library from Salesforce, offers a variety of advanced VLT models, along with GradCAM activation mapping. [5]

2. HuggingFace - We relied on Hugging Face for model training and leveraged pre-trained models like BLIP, and CLIP. The question-answer pairs were prepared for input into a Vision Language model using the **BlipProcessor**. Subsequently, the dataset was partitioned into training and testing subsets using **datasets** library.

3. PEFT: Parameter-Efficient Fine-Tuning is a library designed for efficiently adapting large pretrained models to various downstream applications by fine-tuning only a small number of additional model parameters. This approach significantly reduces computational and storage costs while maintaining performance levels comparable to fully fine-tuned models.

4. DETIC: The proposed approach utilizes both box labels from detection datasets (Ddet) and image-level labels from classification datasets (Dcls) during training.

It follows a two-stage detector training strategy for images with box labels, and for those with image-level labels, it solely trains features from fixed region proposals for classification. https://github.com/facebookresearch/Detic

5. Detectron2: Detectron2 serves not just as a platform for object detection exploration, but also furnishes a range of pre-trained models, simplifying the initiation of object detection endeavors. In essence, Detectron2 appears to be a robust and adaptable toolkit for object detection investigations.https://detectron2.readthedocs.io/en/latest/index.html#

6. RoboFlow: Roboflow simplifies the process by automating image labeling and segmentation tasks, which are typically labor-intensive. https://roboflow.com/

7. What We have Learned

In this section, we will share the insights and lessons learned from this course project.

7.1. Sumanth Manduru

Engaging in this project has been both enjoyable and highly educational, providing me with a valuable opportunity to delve into the realm of multimodality. Throughout this journey, I have gained insights into various concepts, including Parameter Efficient Finetuning (PEFT), which involves adapting large pretrained models to downstream applications without fine-tuning all parameters. One notable PEFT technique I explored is LoRA, which offers efficient adaptation for different tasks. While exploring the Visual Language Model (VLM), I focused on examining its performance across different characteristics of birds. This exploration allowed me to compare the effectiveness of Regular Finetuning with PEFT techniques like LoRA. Additionally, I encountered new evaluation metrics like WUPS [4] and Levenshtein Distance, which offer alternative measures of model performance in the era of Large Language Models (LLMs). Throughout the project, I extensively referenced online sources to implement these techniques and metrics effectively for our project's objectives. Overall, this project has been a rewarding experience, providing valuable insights into multimodal learning and cutting-edge techniques in natural language processing and computer vision.

7.2. Swathi Guptha

I have used a variety of pre-trained vision-language models in my past projects without going too deeply into their training procedures. It was not until this project that I realized how to train a model in a zero-shot fashion. This project helped me understand how data from various modalities is combined to complete downstream tasks. I discovered that this integration is accomplished, though with different architectural strategies, by concatenating embeddings from various modalities and using the combined data for downstream tasks. Furthermore, before putting a pretrained model into practice, it is crucial to understand how it was trained and whether it is appropriate for the particular downstream task at hand. This newfound knowledge extended the perspective on assessing a precision of the model and emphasized the importance of large dataset utilization in the success of the models. The thorough analysis emphasized the fact that a model that has been trained on a sizable dataset and is performing well on some datasets does not ensure that it will perform similarly on a dataset that is specific to a given domain. It did, however, draw attention to the possibility of fine-tuning the same model to obtain good results on domain-specific datasets, a procedure which typically requires substantial data and computational resources.

7.3. Krishna Preetham

In this project, I have acquired a diverse set of skills and knowledge. Firstly, I delved into various Vision Language Models, gaining insights into their architectures and applications. Additionally, I thoroughly understood Vision Transformers and their limitations, which provided valuable context for model selection. Through extensive reading of papers, I stayed updated on the latest trends in image segmentation and detection and image-text matching, enriching my understanding of these domains. Creating a custom dataset and annotating it using Roboflow was a significant accomplishment, enabling me to tailor data to the project's needs. I grasped the crucial relationship between model size, data quality, and performance, recognizing the importance of both factors in achieving optimal results. Overall, this project has equipped me with a comprehensive skill set in vision-based machine learning, from model selection to dataset creation and fine-tuning techniques.

8. Conclusions

The realm of multimodality presents exciting opportunities for exploration. VLMs excel in generic tasks but struggle with domain-specific knowledge and related downstream tasks. We noticed instances of hallucinations in VLMs, particularly when answering questions related to image content. Responses tend to be repetitive and sometimes inappropriate. Additionally, the textual entailment available for our bird dataset is relatively short in length, posing challenges for multi-modality models to accurately align image-text matching pairs.

To mention some challenges we faced during the course project, We have evaluated the Visual Language Model

(VLM) in the context of Visual Question Answering (VQA), focusing primarily on "Wh" questions. However, one can explore its performance in handling "yes/no" type questions. Secondly, While we have successfully utilized bounding boxes (bbox) of birds to crop images for our experiments, we believe that fine-tuning the VLM using bbox information specific to different body parts could enhance its robustness and reliability. In our exploration of the LoRA (Local Relation Attention) approach, we encountered challenges when attempting to experiment with different hyperparameters, particularly the rank (r) value. While the current implementation of LoRA focuses on target layers of query and value, we envision the potential for further enhancing the capabilities of the model by incorporating additional components such as linear layers, MLPs (Multi-Layer Perceptrons), or projection layers into the target layers.

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[2] A. Gupta, P. Dollar, and R. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

[4] K. Kafle and C. Kanan. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20, Oct. 2017.

[5] D. Li, J. Li, H. Le, G. Wang, S. Savarese, and S. C. Hoi. LAVIS: A one-stop library for language-vision intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 31–41, Toronto, Canada, July 2023. Association for Computational Linguistics.

[6] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[8] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[9] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. *The Caltech-UCSD Birds-200-2011 Dataset*. Jul 2011.

[10] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[11] Z. Yang, K. Kafle, F. Dernoncourt, and V. Ordonez. Improving visual grounding by encouraging consistent gradient-based explanations, 2024.

[12] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022.

Figure 12. Notably, classes like car, train, and bus are expected to perform well compared to previous object detection models, but as depicted in Figure 9, the model fails to detect any instances of these classes across the entire dataset

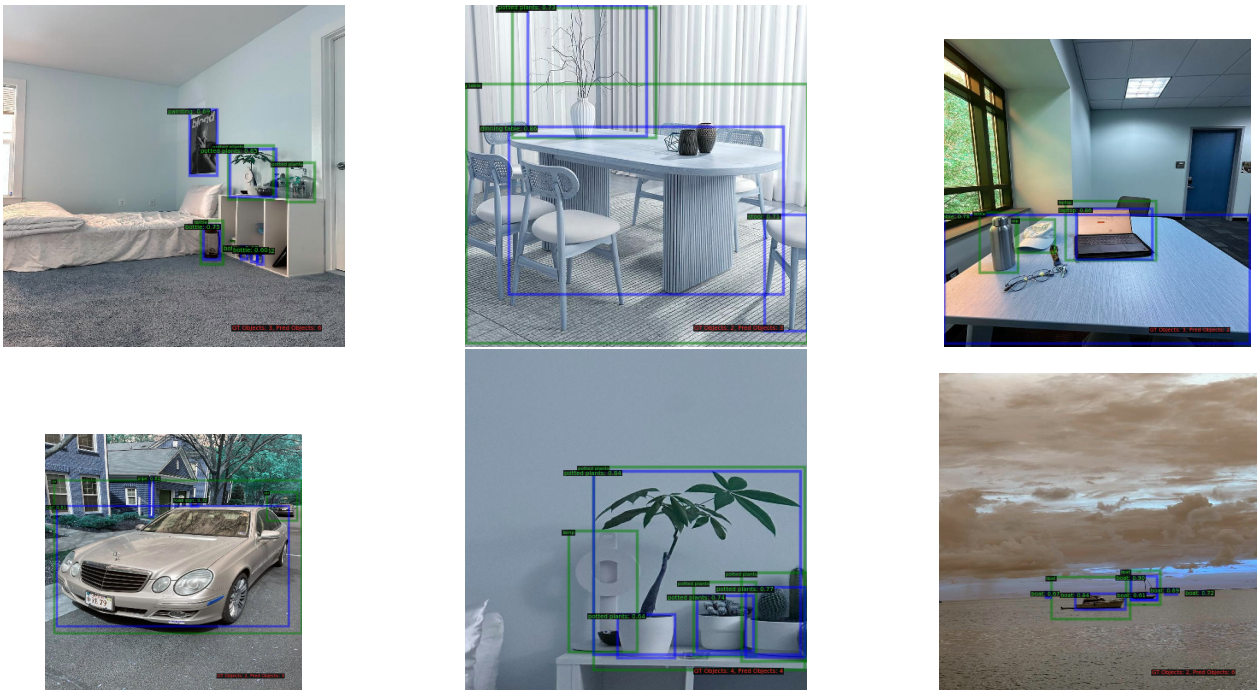


Figure 13. The classes that were previously missed in Pascal VOC—such as potted plants, dining tables, TVs, monitors, cars, and boats—are being correctly detected (Fig 10) in custom dataset.This indicates that the model heavily relies on the CLIP embeddings interpretation of the class. If a class is unknown to the CLIP model, the DETIC model will likely fail

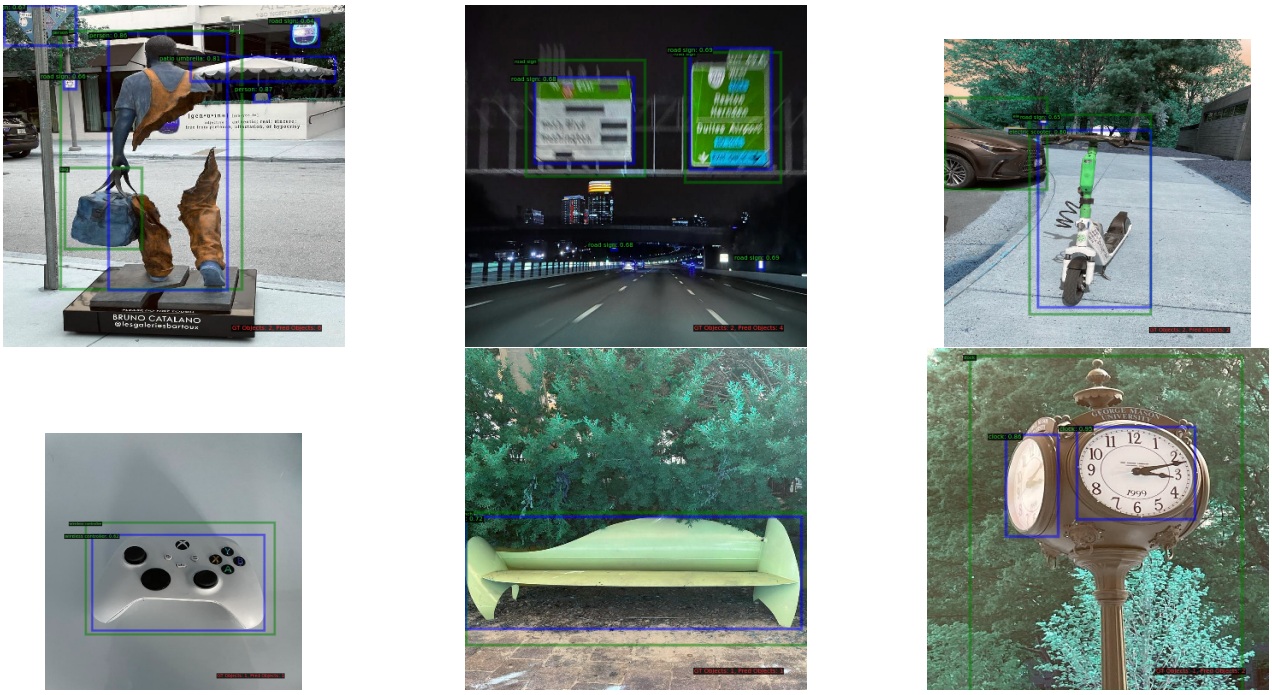


Figure 14. The model is also able detect few very different classes from custom dataset without any finetuning

Figure 15. The model exhibits promising capabilities in detecting novel objects; however, due to the absence of domain-specific training, certain features are misinterpreted, resulting in erroneous object detections