

GeoLLaVA: Efficient Fine-Tuned Vision-Language Models for Temporal Change Detection in Remote Sensing

Hosam Elgendi Ahmed Sharshar Ahmed Aboeitta Yasser Ashraf Mohsen Guizani

Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE
 {hosam.elgendi, ahmed.sharshar, ahmed.aboeitta, yasser.attia, mohsen.guizani}@mbzuai.ac.ae

Abstract

Detecting temporal changes in geographical landscapes is critical for applications like environmental monitoring and urban planning. While remote sensing data is abundant, existing vision-language models (VLMs) often fail to capture temporal dynamics effectively. This paper addresses these limitations by introducing an annotated dataset of video frame pairs to track evolving geographical patterns over time. Using fine-tuning techniques like Low-Rank Adaptation (LoRA), quantized LoRA (QLoRA), and model pruning on models such as Video-LLaVA and LLaVA-NeXT-Video, we significantly enhance VLM performance in processing remote sensing temporal changes. Results show significant improvements, with the best performance achieving a BERT score of 0.864 and ROUGE-1 score of 0.576, demonstrating superior accuracy in describing land-use transformations.

1 Introduction

Understanding temporal changes in remote sensing data is critical for numerous applications, particularly in environmental monitoring, urban planning, and geographical information systems (GIS) (Li et al., 2024). Observing, analysing, and interpreting how geographical features evolve over time can provide valuable insights into environmental trends, land use changes, and the impacts of human activity on the earth’s surface (Statuto et al., 2017; Whig et al., 2024; Siabato et al., 2018). The advancement of Vision language models (VLMs), has made it possible to automate and enhance the detection and interpretation of such temporal changes (Cheng et al., 2024; Zhang et al., 2024).

However, despite significant advancements in VLMs that process both visual and textual data, they still face several key limitations. A major challenge is their computational demand, as training

and fine-tuning large-scale models require substantial resources, making them inefficient for many practical applications, especially when handling large datasets or modeling complex temporal dynamics. Furthermore, many VLMs are optimized for static images and struggle to capture temporal changes (Wang et al., 2023), which are critical in geographical contexts like deforestation, urban sprawl, or seasonal variation. This issue is compounded by the lack of annotated remote sensing datasets that effectively capture temporal changes over time, highlighting a critical research gap in applying VLMs for temporal geographical analysis (Varma et al., 2023).

This paper addresses the aforementioned research gap by introducing an annotated remote sensing dataset of video frame pairs, specifically designed to capture evolving patterns in the data while adapting VLMs for sequential captioning tasks. Using video frames spaced across different time intervals, the task prompts the video language model to generate descriptions explaining changes between two specific moments (Liu et al., 2024d). This approach aims to articulate transitions, actions, or events happening over time in the video frames, enabling the model to capture and describe temporal dynamics more effectively.¹

To enhance the ability of video-language models, including Video-LLaVA and LLaVA-Next (Liu et al., 2023, 2024c), to capture and describe temporal changes, we employ efficient fine-tuning techniques such as Low-Rank Adaptation (LoRA) (Hu et al., 2022) and Quantized LoRA (QLoRA) (Dettmers et al., 2023). This fine-tuning is applied to both few-shot and full datasets, striking a balance between accuracy and computational efficiency. Additionally, model pruning is utilized to further optimize resource usage while maintaining

¹The annotated data and code are available at <https://github.com/HosamGen/GeoLLaVA>

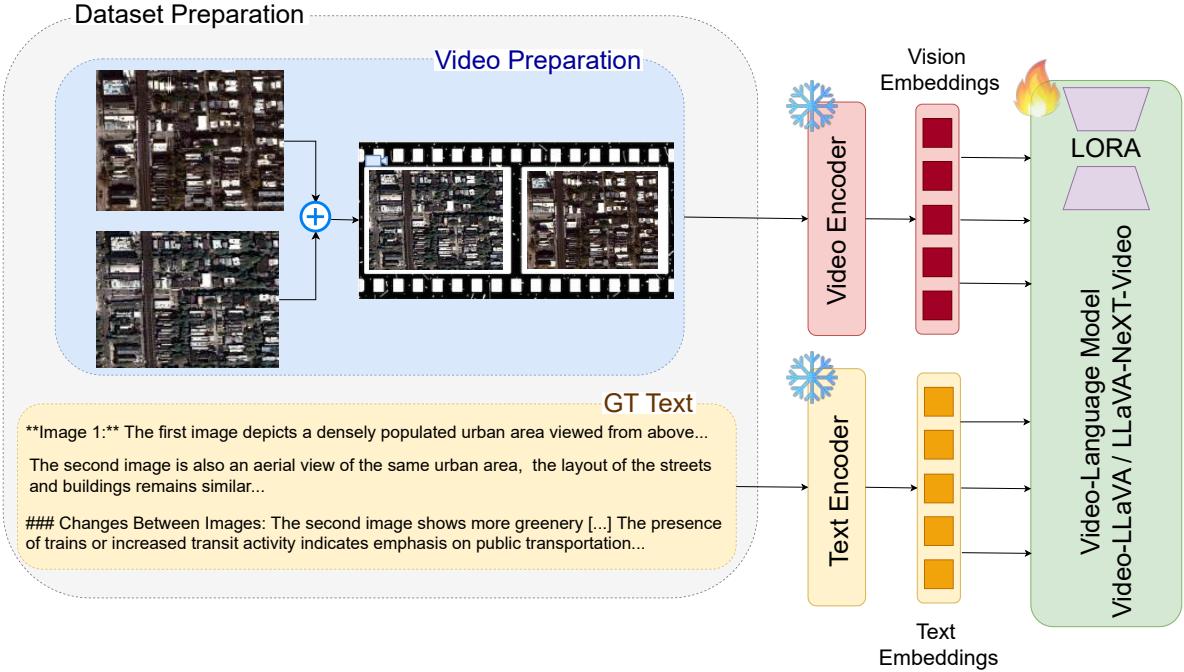


Figure 1: Overview of Our System

performance, ensuring the models are well-suited for real-time applications. Our contributions are summarized as follows:

- **Introduction of a Novel Dataset:** We created an annotated dataset consisting of video frame pairs that track temporal changes in geographical landscapes, particularly focused on urban and environmental transformations over time.
- **Optimized Fine-tuned Model:** By employing techniques like LoRA, QLoRA, and model pruning, we enhanced the efficiency and accuracy of video-language models, specifically Video-LLaVA and LLaVA-NeXT-Video, for detecting temporal changes. These models are evaluated using different metrics including, ROUGE, BLEU, and BERT.
- **Comprehensive Ablation Study:** We conducted an extensive ablation study to assess the impact of different configurations, including LoRA parameters (scale (α), rank (r)), quantization, and pruning ratios.

2 Related Work

Remote sensing datasets are essential for the detailed analysis of temporal and spatial changes within dynamic environments. Foundational datasets such as LEVIR-CD (Chen and Shi, 2020)

and FloodNet (Rahnemoonfar et al., 2021) have significantly advanced the field of change detection. LEVIR-CD primarily focuses on bi-temporal imagery from Google Earth to monitor urban development (Chen and Shi, 2020), while FloodNet utilizes UAV-based data for assessing disaster impacts (Rahnemoonfar et al., 2021). Additionally, datasets like SpaceNet and ERA have contributed to the domains of feature extraction and event recognition, respectively, whereas ISBDA offers granular disaster impact assessments (Etten et al., 2019; Mou et al., 2020; Zhu et al., 2021).

The limitations of current datasets are clear when considering LEVIR-CD’s restricted scope of 637 image pairs and RSICap’s focus on static scene descriptions, which fail to support studying temporal changes (Chen and Shi, 2020; Hu et al., 2023). Although SkyScript boasts a substantial corpus of 2.6 million image-text pairs, it focuses on static imagery rather than evolving visual data (Wang et al., 2024). Additionally, methods like RemoteCLIP highlight the challenges of combining visual and textual features without explicitly incorporating temporal dynamics (Liu et al., 2024a).

Recent advancements in VLMs have substantially impacted remote sensing, particularly by enabling the integration of visual data with linguistic descriptions. This progress has facilitated tasks such as image captioning, zero-shot classification,

and visual question answering (Zia et al., 2022; Li et al., 2023; Chappuis et al., 2022; Yuan et al., 2022; Kuckreja et al., 2024). Notable models like RemoteCLIP and GeoChat have endeavored to merge visual and textual data through training on extensive image-text datasets (Liu et al., 2024a; Kuckreja et al., 2024). However, their applicability in remote sensing remains constrained due to a predominant focus on static image datasets, which neglect the temporal dependencies critical to multi-temporal data analysis. For instance, RSGPT primarily enhances image captioning and visual question answering without addressing sequential data analysis (Hu et al., 2023).

Despite efforts to enhance these models using specialized datasets like RSICap, VLMs continue to exhibit limitations in effectively capturing and analyzing temporal changes, a capability fundamental to environmental monitoring and urban development applications (Wang et al., 2024). GeoChat, while improving multitask conversational capabilities within remote sensing, still lacks the necessary capabilities to evaluate image evolution over time (Kuckreja et al., 2024). Additionally, RemoteCLIP has successfully integrated multi-modality for various computational tasks, their functions remain predominantly limited to zero-shot classification, without extending it to temporal scene analysis (Liu et al., 2024a).

3 Dataset Introduction and Processing

To enable visual-language models (VLMs) to process temporal information, we propose a large-scale dataset comprising scene descriptions and change detection for training VLM architectures. This dataset includes visual interpretations of each image and summaries of changes between image pairs, providing insights into transformations in nature and civilization over time.

3.1 fMoW Dataset

The fMoW RGB dataset, introduced in (Christie et al., 2018), is a high-resolution satellite imagery dataset targeting the classification of 62 categories (Cong et al., 2022). It consists of 363,571 training images and 53,041 validation images, with all multi-spectral imagery converted to JPEG format for ease of use. Images were acquired globally between 2002 and 2017, and the dataset was published in 2018. Its high spatial resolution, ranging from 0.3 to 3.7 meters, enhances the accuracy of

change detection and descriptions of natural and urban environments compared to other sources.

3.2 Creating Image Pairs

Using metadata, we sorted images by location and timestamp to create an ordered list based on the unique "*location_id*" identifiers. For each location, we selected image pairs that are at least 12 months apart. For example, starting with *image_1*, we find *image_2* as the next image satisfying the time difference, and this process continues sequentially from *image_2*, as illustrated in Figure 2.

Due to size and processing limitations of OpenAI's ChatGPT (OpenAI, 2024), images over 1MB were excluded from the training and validation datasets, resulting in the removal of 5,379 training images (1.4%) and 785 validation images (1.4%).

3.3 Data Splits

After filtering and annotating, we created a dataset of 100,000 image pairs from 173,348 images for training and 6,042 pairs from 11,349 images for testing. The test dataset was derived from the original validation dataset of fMoW, with some images randomly selected for inclusion in the training dataset to achieve the complete set of 100,000 pairs. These splits are made available for the reproducibility of the model results.

While the annotation costs with ChatGPT were a consideration, this dataset sufficiently meets our project's objectives. The final dataset maintains most original classes and includes a range of image resolutions between (293,230) and (4766,4634) pixels. It is worth noting that we manually reviewed the testing dataset to verify the authenticity and accuracy of the GPT model's annotations prior to model assessment.

3.4 Temporal Annotations

With image pairs established, annotations are necessary to describe each satellite image and summarize the changes between them relative to the time difference. This was achieved using OpenAI's API, where each image was processed through the "GPT-4o mini" model with the prompt:

"Briefly describe each image independently, then explain the changes happening between them."

The API responses typically provide descriptions of objects and landscapes, including water bodies, green ecosystems, and urban areas. The

image_name	timestamp
airport_hangar_23_6_rgb.jpg	07/06/2008
airport_hangar_23_2_rgb.jpg	21/11/2010
airport_hangar_23_4_rgb.jpg	13/06/2012
airport_hangar_23_1_rgb.jpg	06/01/2014

image_1	image_2	Video name
airport_hangar_23_6_rgb.jpg	airport_hangar_23_2_rgb.jpg	airport_hangar_23_6-airport_hangar_23_2.mp4
airport_hangar_23_2_rgb.jpg	airport_hangar_23_4_rgb.jpg	airport_hangar_23_2-airport_hangar_23_4.mp4
airport_hangar_23_4_rgb.jpg	airport_hangar_23_1_rgb.jpg	airport_hangar_23_4-airport_hangar_23_1.mp4

Figure 2: Overview of the video creation process from the original fMoW dataset images.

second image description often references the first, highlighting structural and environmental changes. The final paragraph explicitly details the changes, focusing on seasonal variations, vegetation dynamics, and alterations in urban landscapes.

Annotations for training and testing were generated in the standard LLaVA format (Liu et al., 2023), including a unique ID, video name, and conversational data between humans and AI. This data is structured from multiple templates (in Appendix A for the human prompts, with API responses serving as the "GPT" replies. The conversational format is structured as:

HUMAN: <Question/Prompt> <Video-tokens>
GPT: <Image Descriptions and Changes Summary>

In conclusion, our dataset provides sufficient training samples for fine-tuning VLMs and is derived from open-source data. The annotations created contribute to making the largest available dataset for grounded image captioning of satellite images while remaining open-source and accessible.

4 Experimental Setup

This work presents an efficient and optimized fine-tuning pipeline aimed at enhancing the temporal understanding of geographical landscapes while highlighting significant land-use changes. We propose an architecture that integrates video processing, custom prompt construction, and fine-tuning tailored for state-of-the-art (SOTA) VLMs. Each segment of the architecture contributes uniquely to the overall system's functionality.

Initially, pairs of images are transformed into videos, with each image serving as an individual frame. These videos are then processed through a video encoder, which uniformly samples the frames and outputs a tensor array of visual information. Simultaneously, the corresponding text inputs are passed to the VLM for textual encoding, enabling the model to align the textual and visual data effec-

tively. Through fine-tuning, the model parameters are updated, transitioning from a general-purpose model to a specialized domain-specific model capable of accurately describing the input frames and detecting changes in accordance with the provided prompts. An overview of this complete system is illustrated in Figure 1.

The fine-tuning process facilitates efficient learning and optimization of the model parameters. To assess the model's capability within the specific domain, we conducted zero-shot tests using the chosen base models, demonstrating their ability to perform well on unseen data. Additionally, a 10k sampled sub-dataset was used for few-shot tuning, allowing for targeted adjustments based on specific examples of land-use changes.

4.1 Model Fine-tuning

Pre-training VLMs is typically computationally intensive and time-consuming. Consequently, fine-tuning presents an effective alternative that preserves most of the model's parameters while enhancing performance on downstream tasks. Fine-tuned models can often outperform the original general models, utilizing fewer computing resources and requiring less training time (Patil and Gudavada, 2024). This advantage motivates the use of Parameter-Efficient Fine-Tuning (PEFT) methods for tasks involving geographical change detection.

In our work, we focus on fine-tuning two distinct models that have demonstrated a robust understanding of temporal data through video processing within the VLM framework for question-answering and captioning. The first model, LLaVA-NeXT (Liu et al., 2024c), was introduced in early 2024, offering improved reasoning and world knowledge compared to other large models. It exhibits data efficiency comparable to SOTA models such as LLaVA-1.5 (Liu et al., 2024b), while also delivering higher image resolution and enhanced visual conversation capabilities. Shortly after the release of LLaVA-NeXT, a video variant was introduced, named LLaVA-NeXT-Video, which has demon-

strated strong performance in zero-shot video tasks.

The second model utilized for comparison is Video-LLaVA (Lin et al., 2023), which excels in understanding visual language for downstream tasks and surpasses many existing video language models across various benchmarks. Both projects have multiple variations based on the number of parameters for the models. For simplicity, we have chosen to use the 7B parameter variation from both models. The 7B variations can be fine-tuned with PEFT techniques on a single GPU, making them particularly well-suited for our dataset.

4.2 Low Rank Adaptation

Low Rank Adaptation (LoRA) is based on a pivotal insight that the disparity between the fine-tuned weights for a specific task and the original pre-trained weights often exhibits ‘‘low intrinsic rank’’, which implies that the disparity can be approximated by a matrix of low rank (Hu et al., 2022).

For an initial pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, LoRA limits its update through a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and the rank $r \ll \min(d, k)$. Throughout the training process, W_0 remains unchanged and does not receive gradient updates, whereas A and B are endowed with trainable parameters. It is noteworthy that both W_0 and $\Delta W = BA$ are applied to the same input, with their outputs being aggregated coordinate-wise. For an output $h = W_0x$, the modified forward pass is:

$$h = W_0x + \Delta Wx = W_0x + BAx \quad (1)$$

Typically, A is initialized with a random Gaussian distribution, and B with zero, ensuring that $\Delta W = BA$ starts from zero at the inception of training. The scaling of ΔWx by α/r . The rank of the low rank matrices is denoted by r , whereas α is the scaling factor that controls the magnitude of updates to the matrices.

4.3 Evaluation Metrics

Many evaluation metrics are taken into consideration to evaluate the performance of the fine-tuned models and evaluate the model’s generated text against the ground truth text. In similar works about fine-tuned models in domain-specific tasks, metrics such as ROUGE, BLEU, and METEOR are used. All three metrics compare the overlap of n-grams or phrases between the generated output and the reference text.

ROUGE Score (Lin, 2004) measures the N-gram overlap between a candidate’s output and a set of reference outputs, ROUGE-1, ROUGE-2, and ROUGE-L were used. Where 1 and 2 are the n-grams, and ROUGE-L is based on the Longest common subsequence. The equations for this metric are available in Appendix B.

BLEU (Papineni et al., 2002) is commonly employed for generation tasks to measure n-gram similarities between machine-generated outputs and reference translations. Although our work is not focused on translation, we utilize this metric to assess our generated outputs. It is calculated as follows:

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right) \quad (2)$$

where p_n is the precision for n-grams of length n , w_n are weights (50% for each n-gram in this work), and N is the maximum n-gram length considered, $N = 2$ in this work.

Although these two metrics give a good idea of the model’s performance by comparing words and sentences and matching them against the reference text, they both have limited contextual understanding or capture the semantic coherence of the generated or reference texts. Therefore, the BERT metric is also utilized to provide performance indicators between texts by generating contextual embedding to capture the semantic similarity between words.

BERT Score (Zhang et al., 2019) employs BERT (Devlin, 2018) to assess the similarity between two sentences by aligning each token in the reference sentence with the closest token in the candidate sentence. This similarity is determined through the cosine similarity of the token embeddings. Precision is calculated by comparing the candidate tokens with those in the reference, while recall involves matching reference tokens to those in the candidate. The F1 score is subsequently derived from both precision and recall. The formulas for recall, precision, and F1 are:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{x_j \in \hat{x}} x_i^T \hat{x}_j \quad (3)$$

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^T \hat{x}_j \quad (4)$$

$$F_{\text{BERT}} = \frac{2 \cdot P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \quad (5)$$

	Video LLaVA 7B					LLaVA NeXT Video 7B				
	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERT	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERT
Base	0.211	0.041	0.122	0.039	0.456	0.197	0.037	0.113	0.042	0.404
10K LORA	0.563	0.214	0.313	0.243	0.849	0.554	0.198	0.300	0.232	0.856
100K LORA	0.576	0.226	0.325	0.250	0.863	0.562	0.199	0.300	0.239	0.864
10K-QLORA	0.565	0.212	0.310	0.243	0.845	0.543	0.193	0.283	0.213	0.836
100K-QLORA	0.571	0.220	0.316	0.250	0.854	0.561	0.202	0.302	0.229	0.858
10K Pruning 5%	0.031	0.007	0.024	0.010	0.265	0.532	0.178	0.278	0.209	0.829
100K Pruning 5%	0.125	0.034	0.110	0.043	0.359	0.541	0.183	0.284	0.210	0.840
Final Model	-	-	-	-	-	0.556	0.202	0.290	0.227	0.850

Table 1: Table comparing different variations of Video LLaVA and LLaVA NeXT Video models (Base, LoRA, QLoRA, and Pruning) using ROUGE-1, ROUGE-2, ROUGE-L, BLEU, and BERTScore metrics.

where x and \hat{x} denote the reference and candidate sentences, respectively, and $x_i^T \hat{x}_j$ represents the cosine similarity between token embeddings x_i and \hat{x}_j .

4.4 Model Optimization

Pruning: We implement magnitude-based fine-grained pruning, an unstructured pruning method that selectively removes individual weights based on their magnitude. Precisely, the weights with the smallest absolute values are pruned for each layer according to a predefined sparsity level. The pruning process uses a binary mask that retains important weights (those with larger magnitudes) and sets the others to zero. A global sparsity target is applied to ensure consistent pruning across the model, although certain layers, such as embeddings and critical vision model components, are excluded to preserve model performance. This fine-grained approach allows for more granular control over which weights are pruned, resulting in reduced model size and computational overhead with minimal impact on accuracy. During inference, the pruning masks are re-applied to maintain the enforced sparsity, optimizing the model for efficiency without sacrificing performance.

In magnitude-based fine-grained pruning, each weight $W_{i,j}$ in the weight matrix W is pruned based on its absolute value $|W_{i,j}|$. The pruning threshold τ is determined such that the smallest $s \times 100\%$ of weights, where s is the sparsity level, are pruned. A binary mask M is created, where

$$M_{i,j} = \begin{cases} 1 & \text{if } |W_{i,j}| > \tau \\ 0 & \text{if } |W_{i,j}| \leq \tau \end{cases} \quad (6)$$

The pruned weights are then obtained by element-wise multiplication of the original weight matrix W with the mask M , yielding

$$W^{\text{pruned}} = W \odot M \quad (7)$$

This process reduces the model’s parameter count while retaining the most significant weights.

QLORA: Although LoRA reduces the overall number of parameters to be modified from the original model, it is still challenging to fine-tune the total number on a single device. Therefore, quantization for LLMs was introduced in (Dettmers et al., 2023) to optimize the computation process for reduced memory usage while maintaining model accuracy, referred to as QLORA. The quantization (q) is calculated by:

$$q = \text{round} \left(\frac{(2^b - 1)}{\text{absmax}(X)} \cdot X \right) = \text{round}(c \cdot X) \quad (8)$$

where c is the quantization constant or quantization scale.

Ultimately, all models were fine-tuned with a single 48GB A6000 GPU, for one epoch, taking on average between 2 hours and 24 hours with batch size 3 for Video-LLaVA 7B for the 10k and 100k datasets respectively. As for LLaVA-NeXT Video 7B, the batch size was 2, tuning for 3 hours to 27 hours for the 10k and 100k datasets respectively. The full hyper-parameters and training configurations can be found in Appendix B.

5 Results & Discussion

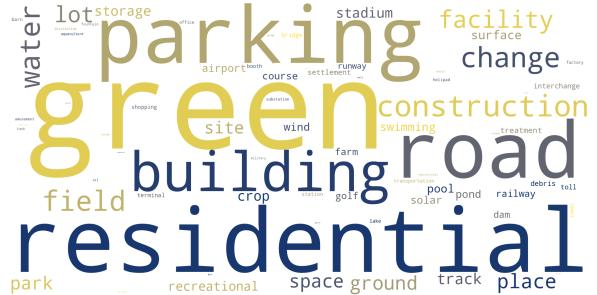
Table 1 presents the experimental results across different models and scoring criteria. Initially, we evaluated both models without any fine-tuning to assess their baseline capabilities. The base models struggled to generate meaningful outputs, performing poorly across all metrics. Even using the BERT score for semantic evaluation, the models

	r	α	Video LLaVA 7B				LLaVA NeXT Video 7B					
			ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERT	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERT
10K QLORA-4bit	640	1280	0.561	0.211	0.313	0.236	0.855	0.531	0.177	0.281	0.196	0.834
100K QLORA-4bit	640	1280	0.570	0.219	0.318	0.245	0.860	0.545	0.187	0.287	0.205	0.844
10K QLORA-4bit	64	256	0.556	0.203	0.307	0.233	0.854	0.539	0.185	0.288	0.213	0.834
100K QLORA-4bit	64	256	0.567	0.212	0.311	0.240	0.863	0.555	0.196	0.296	0.225	0.848
10K QLORA-8bit	64	128	0.566	0.221	0.319	0.248	0.844	0.542	0.183	0.288	0.211	0.842
100K QLORA-8bit	64	128	0.578	0.224	0.320	0.252	0.863	0.557	0.195	0.294	0.224	0.852
10K Pruning_10%	64	128	0.025	0.005	0.018	0.008	0.250	0.479	0.130	0.224	0.171	0.747
100K Pruning_10%	64	128	0.063	0.018	0.052	0.021	0.289	0.529	0.175	0.274	0.204	0.823

Table 2: Table comparing various configurations of LLaVA and LLaVA NeXT models (including QLoRA with 4-bit and 8-bit precision and Pruning at 10%) across evaluation metrics: ROUGE-1, ROUGE-2, ROUGE-L, BLEU, and BERTScore. The variations are analyzed using different ranks (r) and alpha values (α).



(a) Ground truth captions.



(b) Final model's generated captions.

Figure 3: Word clouds comparing the ground truth (left) and the final model’s generated annotations (right).

demonstrated minimal ability to capture meaningful changes between objects in the input images.

We then applied few-shot fine-tuning using 10% of the data (10K samples) and, subsequently, the full dataset (100K samples). The first fine-tuning approach employed LoRA with $r = 64$ and $\alpha = 128$, following the recommended 2:1 ratio between α and r . This configuration required tuning 178M parameters. The performance improved significantly, especially with the 100K sample, yielding the highest BERT score of 0.864.

To enhance efficiency, we utilized QLoRA with 4-bit quantization, significantly reducing memory requirements by around 75% without compromising performance. Despite the reduced precision, the model achieved a BLEU score of 0.250, matching that of the LoRA-based approach.

We optimized the model by pruning 5% of the parameters to reduce its size while maintaining accuracy. However, pruning amplified the performance gap between the 10K and 100K datasets, suggesting that more data is needed to mitigate the degradation caused by pruning. Notably, the LLaVA-Next video model outperformed Video-LLaVA, thanks to its sparse structure, achieving a BERT score of 0.823—only 0.03 lower than the best result. In contrast, Video-LLaVA faced significant challenges

with pruning due to its dense architecture, making it unsuitable for pruning.

Table 2 summarizes our ablation study to validate the chosen hyperparameters. We explored several α and r ratios to determine their impact on performance. Increasing the fine-tuned parameters to 1.7B by setting $r = 640$ and $\alpha = 1280$ did not yield significant performance gains, highlighting diminishing returns at higher parameter counts. Modifying the α and r ratio to 4:1 also resulted in negligible improvements or degraded performance. Therefore, we adopted the $r = 64$ and $\alpha = 128$ configuration for subsequent experiments.

We also tested 8-bit quantization but found that the slight performance improvement came at the cost of increased memory usage, reducing efficiency. The 8-bit quantization uses 150% more GPU memory and requires 174% more time to fine-tune compared to the 4-bit model. As a result, we retained the 4-bit quantization configuration. Similarly, increasing pruning to 10% further reduced the model size but significantly harmed accuracy. While greater pruning might suit applications prioritizing efficiency over accuracy, we chose the intermediate solution of 5% pruning to balance performance and efficiency.

Using 4-bit QLoRA, 5% pruning, and fine-



Question: Describe the images in the video frames, and elaborate on the changes between them.



Image 1: This aerial view shows a densely built urban area with a grid layout of streets and buildings. The buildings have reddish roofs and are interspersed with patches of greenery. There are several parking areas visible, and the overall scene appears somewhat organized with clear roadways.

Image 2: This aerial view also depicts a similar urban environment, but with noticeable differences. The layout remains consistent with a grid pattern, but there are more visible green spaces and trees. The buildings still have reddish roofs, but some areas appear less developed or have been altered. The parking areas are more pronounced, and there are additional structures or features that suggest ongoing development or changes in land use.

Changes Between the Images: The transition between the two images indicates urban development and changes in land use. In the second image, there is an increase in green spaces and trees, suggesting either new landscaping or the preservation of natural areas. The overall density of buildings appears to remain consistent, but the arrangement of parking and possibly new structures indicates ongoing urban development.



Figure 4: Qualitative output showcasing a sample video of two frames inputted with a question followed with the model output describing the two images and summarizing differences and changes.

tuning on 100K samples, our final model demonstrated competitive performance, achieving a BERT score of 0.850. The results show that although this setup does not yield the highest possible scores, it strikes an optimal balance between accuracy and efficiency. Specifically, the model is 5% less resource-intensive while still performing well.

For qualitative analysis, Figure 3 presents word cloud representations comparing the ground truth and the model-generated captions. The visual overlap highlights the alignment between key descriptive terms in both the ground truth and the model outputs, showcasing the model’s ability to capture salient features. Figure 4 provides an additional qualitative example, where the model describes two input images with high attention to key objects. The model accurately identifies changes between the images, using smooth and coherent language to summarize differences.

Our results align closely with SOTA models such as GPT-4, and the generated annotations demon-

strate strong consistency with human descriptions.

6 Conclusion

In this paper, we introduced a novel dataset and applied fine-tuning techniques with it to enhance VLMs for detecting temporal changes in geographical landscapes. By employing methods such as LoRA and QLoRA, we enhanced models like Video-LLaVA and LLaVA-NeXT-Video. Our fine-tuned models surpassed the performance of the base models, with the final model achieving a BERT score of 0.864 and a ROUGE-1 score of 0.576. Furthermore, the use of quantization and pruning improved computational efficiency without degrading accuracy, making the models more suitable for real-time applications. This work addresses key limitations in remote sensing, providing a scalable and efficient solution for tracking temporal changes in environmental and urban landscapes.

7 Limitation & Future Work

Reliance on GPT-4o mini for Annotations: We relied on GPT-4o mini for captioning and annotations, which provided reliable ground truth. However, incorporating annotations from other models could lead to a more comprehensive comparison and capture a wider range of nuances in temporal changes. Future work could explore integrating multiple models to enhance annotation diversity and improve the robustness of temporal change detection.

Limited Dataset Due to High Labeling Costs: The high cost of labeling with GPT-4o mini limited us to 100,000 image pairs for training and 6,000 pairs for testing. Larger datasets would improve model performance by providing more diverse examples for better fine-tuning and generalization. Future efforts should focus on creating larger datasets to enhance model training and allow for better generalization across diverse conditions.

Single Dataset: Although our dataset is diverse in terms of classes, locations, and image characteristics, it is still a single dataset. Using multiple datasets with different collection schemas would improve the model’s generalization across various environments and tasks. In the future, expanding the dataset to include multiple data sources with varying temporal resolutions will be crucial to improving model generalization.

Manual Evaluation and Crowdsourcing: We were only able to manually evaluate the test data due to the large volume of training data. Crowdsourcing annotations could enhance captioning verification and quality, though it would come with increased costs. Future work could explore the feasibility of crowdsourcing annotations to further validate and improve the quality of the dataset at scale.

Hardware Limitations: Hardware limitations restricted us to a single GPU. Access to larger, more advanced GPUs would improve model performance by allowing faster processing, larger batch sizes, and the ability to train models with more parameters. Future research could benefit from leveraging more powerful hardware to expedite training processes and accommodate more complex model architectures.

Supervised Learning Only: We only employed supervised learning, which made it easier for the model to learn. While this approach helped achieve reliable results, it limited the model’s abil-

ity to generalize to unseen data. Future work could explore the integration of unsupervised or semi-supervised learning approaches, which, though more challenging, could lead to more robust and generalized results, particularly in low-resource settings.

Model Distillation: Techniques like distillation via the teacher-student model were not explored in this work. Distillation could significantly reduce the model’s complexity and computational resource requirements while maintaining performance levels. Future research could focus on applying model distillation techniques to streamline the model, making it more efficient and suitable for real-world applications with limited computational resources.

References

- Christel Chappuis, Valérie Zermatten, Sylvain Lobry, Bertrand Le Saux, and Devis Tuia. 2022. [Prompt-rsvqa: Prompting visual context to a language model for remote sensing visual question answering](#). In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1371–1380.
- Hao Chen and Zhenwei Shi. 2020. [A spatial-temporal attention-based method and a new dataset for remote sensing image change detection](#). *Remote Sensing*, 12(10).
- Guangliang Cheng, Yunmeng Huang, Xiangtai Li, Shuchang Lyu, Zhaoyang Xu, Hongbo Zhao, Qi Zhao, and Shiming Xiang. 2024. [Change detection methods for remote sensing in the last decade: A comprehensive review](#). *Remote Sensing*, 16(13).
- Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. 2018. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180.
- Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. 2022. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, 35:197–211.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [QLoRA: Efficient finetuning of quantized LLMs](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Adam Van Etten, Dave Lindenbaum, and Todd M. Baccastow. 2019. [Spacenet: A remote sensing dataset and challenge series](#). *Preprint, arXiv:1807.01232*.

- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. **LoRA: Low-rank adaptation of large language models**. In *International Conference on Learning Representations*.
- Yuan Hu, Jianlong Yuan, Congcong Wen, Xiaonan Lu, and Xiang Li. 2023. **Rsgpt: A remote sensing vision language model and benchmark**. *Preprint*, arXiv:2307.15266.
- Kartik Kuckreja, Muhammad S. Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad S. Khan. 2024. Geochat: Grounded large vision-language model for remote sensing. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Xiang Li, Congcong Wen, Yuan Hu, Zhenghang Yuan, and Xiao Xiang Zhu. 2024. **Vision-language models in remote sensing: Current progress and future trends**. *IEEE Geoscience and Remote Sensing Magazine*, 12(2):32–66.
- Xiang Li, Congcong Wen, Yuan Hu, and Nan Zhou. 2023. **Rs-clip: Zero shot remote sensing scene classification via contrastive vision-language supervision**. *International Journal of Applied Earth Observation and Geoinformation*, 124:103497.
- Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. 2024a. **Remoteclip: A vision language foundation model for remote sensing**. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024c. **Llava-next: Improved reasoning, ocr, and world knowledge**.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. **Visual instruction tuning**. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Shihong Liu, Samuel Yu, Zhiqiu Lin, Deepak Pathak, and Deva Ramanan. 2024d. Language models as black-box optimizers for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12687–12697.
- Lichao Mou, Yuansheng Hua, Pu Jin, and Xiao Xiang Zhu. 2020. **Era: A data set and deep learning benchmark for event recognition in aerial videos [software and data sets]**. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):125–133.
- OpenAI. 2024. Chatgpt: A large language model. <https://chat.openai.com/>. Accessed: 2024-10-07.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Rajvardhan Patil and Venkat Gudivada. 2024. A review of current trends, techniques, and challenges in large language models (llms). *Applied Sciences*, 14(5):2074.
- Maryam Rahnamoonfar, Tashnim Chowdhury, Argho Sarkar, Debraj Varshney, Masoud Yari, and Robin Roberson Murphy. 2021. **Floodnet: A high resolution aerial imagery dataset for post flood scene understanding**. *IEEE Access*, 9:89644–89654.
- Willington Siabato, Christophe Claramunt, Sergio Ilarri, and Miguel Angel Manso-Callejo. 2018. **A survey of modelling trends in temporal gis**. *ACM Comput. Surv.*, 51(2).
- Dina Statuto, Giuseppe Cillis, and Pietro Picuno. 2017. **Using historical maps within a gis to analyze two centuries of rural landscape changes in southern italy**. *Land*, 6(3).
- Maya Varma, Jean-Benoit Delbrouck, Sarah Hooper, Akshay Chaudhari, and Curtis Langlotz. 2023. Villa: Fine-grained vision-language representation learning from real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22225–22235.
- Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. 2023. **EfficientVLM: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13899–13913, Toronto, Canada. Association for Computational Linguistics.
- Zhecheng Wang, Rajanie Prabha, Tianyuan Huang, Ji-jun Wu, and Ram Rajagopal. 2024. **Skyscript: A large and semantically diverse vision-language dataset for remote sensing**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5805–5813.
- P. Whig, A.B. Bhatia, R.R. Nadikatu, Y. Alkali, and P. Sharma. 2024. **Gis and remote sensing application for vegetation mapping**. In T. Choudhury, B. Koley, A. Nath, JS. Um, and A.K. Patidar, editors, *Geo-Environmental Hazards using AI-enabled Geospatial Techniques and Earth Observation Systems*, Advances in Geographic Information Science. Springer, Cham.

Zhenghang Yuan, Lichao Mou, Qi Wang, and Xiao Xiang Zhu. 2022. From easy to hard: Learning language-guided curriculum for visual question answering on remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 60.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. [Vision-language models for vision tasks: A survey](#). *Preprint*, arXiv:2304.00685.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiaoyu Zhu, Junwei Liang, and Alexander Hauptmann. 2021. Msnet: A multilevel instance segmentation network for natural disaster damage assessment in aerial videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2710–2720. IEEE.

Usman Zia, M. Mohsin Riaz, and Abdul Ghafoor. 2022. [Transforming remote sensing images to textual descriptions](#). *International Journal of Applied Earth Observation and Geoinformation*, 108:102741.

A Dataset Supplementary Details

Distribution of the classes in the dataset, split between train and test splits in Figure 5.

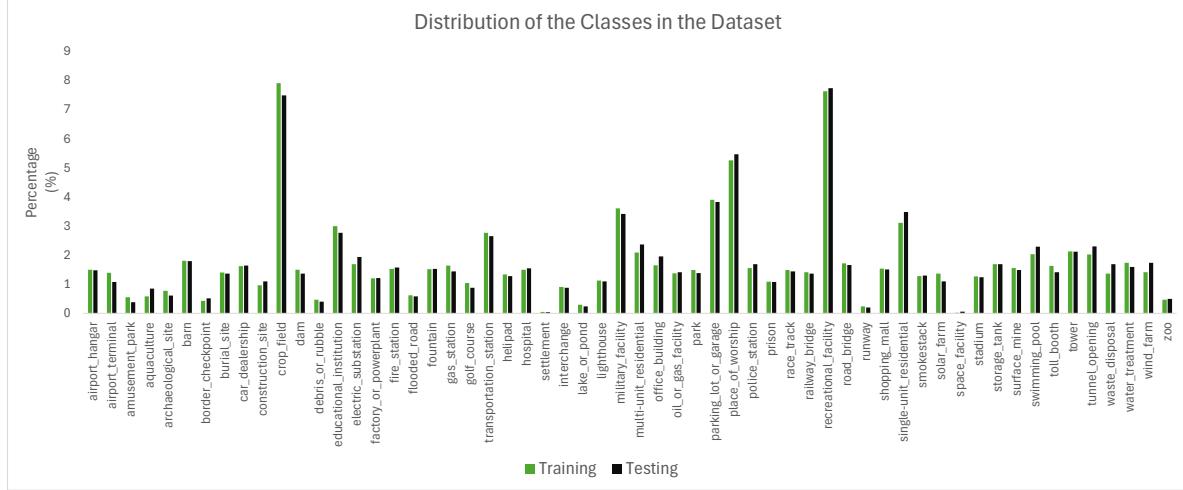


Figure 5: Distribution of the classes in the training and testing dataset.

List of templates used to create prompts for fine-tuning the models:

- "Provide a detailed description of the satellite video, where each frame corresponds to a different time but the same location."
- "Describe the satellite video thoroughly, noting that each frame shows the same location at a different time.", "Give a detailed account of the satellite video, with each frame depicting the same location at distinct points in time."
- "Offer an elaborate explanation of the satellite video, where every frame captures the same location but at different times."
- "Elaborate on the changes in the location as seen in the satellite video, where each frame is a snapshot of the same place at different times."
- "Provide a report describing the satellite video, where each frame shows the same location at different time points."

For each entry, a random template is used as a prompt for the model, while the API response is used as the answer, thus creating annotations in the required format.

B Results and Evaluation Supplementary Details

Equations for the ROUGE evaluation metrics are listed here in this appendix.

For ROUGE-1:

- Precision = $\frac{\sum \text{Count}_{\text{match unigrams in output}}}{\sum \text{Count}_{\text{unigrams in output}}}$
- Recall = $\frac{\sum \text{Count}_{\text{match unigrams in output}}}{\sum \text{Count}_{\text{unigrams in reference}}}$
- F1 = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

For ROUGE-2:

- Precision = $\frac{\sum \text{Count}_{\text{match bigrams in output}}}{\sum \text{Count}_{\text{bigrams in output}}}$
- Recall = $\frac{\sum \text{Count}_{\text{match bigrams in output}}}{\sum \text{Count}_{\text{bigrams in reference}}}$
- F1 = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

For ROUGE-L:

the equations use the concept of the Longest Common Subsequence (LCS). The precision, recall, and F1-score can be defined similarly, but are specifically based on the LCS length. While explicit formulas vary by implementation, the fundamental concept involves calculating the LCS between the system-generated output and the reference text, then applying similar formulas for precision, recall, and F1-score as with ROUGE-1 and ROUGE-2 (Lin, 2004).

- Precision_{LCS} = $\frac{\text{Length of LCS}}{\text{Length of the system-generated output}}$
- Recall_{LCS} = $\frac{\text{Length of LCS}}{\text{Length of the reference text}}$
- F1_{LCS} = $\frac{2 \times \text{Precision}_{\text{LCS}} \times \text{Recall}_{\text{LCS}}}{\text{Precision}_{\text{LCS}} + \text{Recall}_{\text{LCS}}}$

Parameter	Value
MAX_LENGTH	400
MODEL	LLaVA-NeXT-Video-7B-hf
USE_QLORA	True (4-Bit)
batch_size	2
lora_r	64
lora_alpha	128
Training Configuration	
max_epochs	1
val_check_interval	0.2
check_val_every_n_epoch	1
gradient_clip_val	1.0
accumulate_grad_batches	1
learning_rate	1e-4
num_nodes	1
warmup_steps	50

Table 3: Full hyper-parameters used for fine-tuning the final model.