Adapting TinyLLaVA for describing image differences

Megha Chakravorty
UMass Amherst
mchakravorty@umass.edu

Peter Tran
UMass Amherst
mdtran@umass.edu

Abstract

In this project, we introduce a practical approach for transforming the subtle distinctions between pairs of images from video surveillance footage into descriptive text. The primary challenge is the minimal visual differences between these image pairs, which traditionally makes manual differentiation both tedious and time-consuming. To address this, we leverage the capabilities of Vision-and-Language Models (VLMs) to automate the detection and facilitate logging or documenting of these differences. Specifically, we utilize a lightweight variant of the LLaVA model, TinyLLaVA, which we fine-tune to enhance its adaptability to this nuanced task. Furthermore, LLaVA's interactive capabilities allow for the extraction of more detailed information about the observed differences through conversational engagement with an AI assistant. This feature is particularly valuable for probing and pinpointing potential security threats, thereby augmenting the utility of surveillance systems. This paper details the implementation of this approach and evaluates its efficacy in automating the meticulous task of identifying and documenting subtle visual differences. Our approach also provides an opportunity to explore how VLMs align language with vision and capture visual salience. We have employed parameter-efficient fine-tuning techniques like LoRA to achieve improved results over the traditional encoder-decoder based approaches. This paper details our utilitarian approach and its effectiveness in streamlining the process of capturing and documenting subtle visual differences.

1. Introduction

In this paper, we investigate the application of Vision-and-Language Models (VLMs) for security tasks, focusing specifically on detecting differences between image frames extracted from surveillance footage. The growing prominence of foundational models in diverse domains inspires our exploration into their potential for enhancing security systems, where accurately identifying changes in object positions is crucial.

Moreover, this technology has potential applications in assisting visually impaired individuals, akin to existing systems in image captioning and visual question answering that help such individuals interact with their surroundings. By generating intuitive and accessible descriptions of visual data, our model not only advances security monitoring but also contributes to assistive technologies.

We are motivated by the methodologies established in the "Spot-the-Difference" research, which utilizes a neural encoder-decoder architecture with a latent alignment variable to directly align clusters of visual differences with corresponding output sentences. This model also integrates a learned prior that effectively captures the visual salience of these difference clusters. By leveraging alignment through discrete latent variables, this approach significantly outperforms models that rely solely on attention mechanisms. However, the use of the latent variables limits the fluidity and granularity of the alignment between visual differences and text descriptions. The simplification inherent in using discrete choices for alignment may not capture the complex and subtle nuances of visual data as effectively as the continuous representations often employed in more sophisticated VLMs. Additionally, the learned prior, although useful for highlighting salient features, might oversimplify the diverse and context-dependent nature of visual salience, potentially leading to less accurate or overly generalized descriptions in certain scenarios.

In our work, we employ TinyLLaVA [6], a powerful yet lightweight variant of the Large Language and Vision Assistant (LLaVA). Upon fine-tuning, TinyLLaVA can detect and describe a single difference between image frames in a conversation thread, allowing for a thorough examination of each identified change. This also enhances the flexibility by limiting the output generation by a fixed token size. The TinyLLaVA assistant can be probed to elaborate on changes that appear potentially hazardous while providing succinct summaries for those that seem inconsequential or benign.

To validate our model's effectiveness, we propose experiments to examine how it prioritizes changes based on their potential security impact and differentiates between salient and non-salient visual discrepancies. Enhancing the

model with Reinforcement Learning from Human Feedback (RLHF) could further refine its ability to assess and respond to changes, making it an even more valuable tool for security applications.

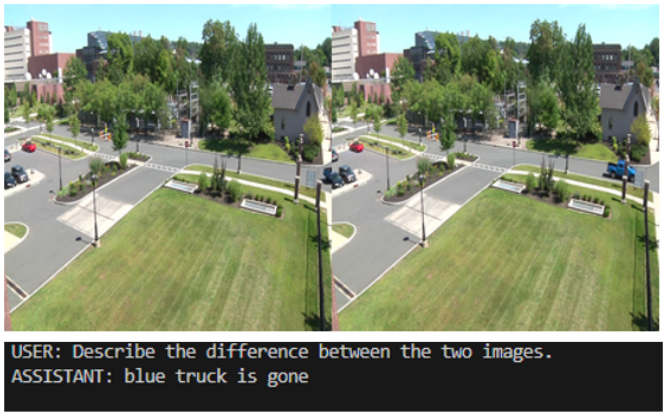


Figure 1. Difference between images of the same scene taken at different times captured by TinyLLaVA Assitant

2. Related Works

The paper "Learning to Describe Differences Between Pairs of Similar Images"[3] presents a methodological advancement in automatically generating text descriptions for the differences between similar images. A principal contribution is the creation of a novel dataset, curated through crowd-sourcing, that features pairs of image frames with associated difference descriptions, crafted to specifically enhance model training in visual salience and language alignment. Their proposed model employs a latent variable to align clusters of differing pixels with generated textual descriptions.

"Expressing Visual Relationships via Language", [5], describes the development of a language-guided image editing dataset and a neural speaker model designed to articulate the visual relationships between pairs of images. This dataset is curated by gathering image pairs from real image editing websites, annotating them with editing instructions via Amazon Mechanical Turk, and refining these annotations with expert input. The neural speaker model begins with an attentive encoder-decoder framework and incorporates two attention-based components: static relational attention and sequential multi-head attention. These components address the complexity of depicting diverse, detailed, and implicit visual transformations between images. The model is further enhanced by a dynamic relational attention module that synthesizes the strengths of the previous components to effectively model visual relationships during the decoding phase, representing a comprehensive method for directly interpreting visual relationships through language.

The "Learning a Recurrent Visual Representation for Image Caption Generation" [2] paper presents a bi-directional model using Recurrent Neural Networks (RNNs) that dynamically generates both descriptions from images and visual representations from descriptions. Central to this model is a unique representation that dynamically updates to capture the visual elements of a scene as they are described. As each word is generated or read, the visual representation is adjusted to incorporate the new information the word conveys. This is particularly significant as traditional RNNs often struggle with retaining concepts after several iterations, making it challenging to maintain long-distance relationships within the text. In this model, the dynamically updated visual representation serves as a long-term memory for retaining relevant information, enabling the network to selectively emphasize salient but as yet unmentioned aspects of the image. This functionality enhances sentence generation and also allows the model to construct visual representations based on textual descriptions, thus integrating visual and textual information seamlessly.

The paper "Video Storytelling: Textual Summaries for Events" [4] focuses on visual storytelling for long videos to compose coherent and succinct textual stories for extended events. This task differs from dense-captioning and photostream storytelling by focusing on extracting and narrating the pivotal scenes from complex video sequences, rather than describing every detail or bridging visual gaps with imagination. The approach presented in the paper addresses outlines a context-aware framework for multimodal semantic embedding learning, utilizing a two-step local-to-global process. This begins with modelling individual clip-sentence pairs to establish a local embedding, then integrating these into a global sequence of clips, utilizing a Residual Bidirectional RNN (ResBRNN) that captures the temporal dynamics and context, enhancing the diversity and coherence of the embeddings. Then a Narrator model is proposed to generate stories by selecting key clips from the video, generating a narrative by retrieving and concatenating sentences that align with these clips in the embedding space. This selection process is guided by reinforcement learning, optimizing a policy to choose clips that maximize narrative coherence and relevance, as measured against human-written reference stories.

The Visual Text Comparison Network (VIXEN)[1] introduces a novel framework for image difference captioning that effectively highlights manipulations in image content by synthesizing descriptive captions of visual differences between image pairs. A key contribution of VIXEN is its integration of a pre-trained image encoder and a large language model, which together process synthetically generated image pairs to produce precise change summaries. This methodology not only addresses the challenges posed by the limited volume and diversity of training data in existing datasets but also enhances the network's capability to generate accurate and comprehensible captions across varied

image edits. This approach leverages advancements such as stable diffusion for image generation and employs the GPT-3 model in a few-shot learning setting to ensure the captions succinctly summarize the depicted changes, thereby offering a potential tool against the spread of misinformation through manipulated images.

3. Dataset

We have utilised the "Spot-the-Diff" dataset, which comprises 13,192 pairs of images with human-generated textual annotations that capture the differences between each pair. The image pairs are extracted from video surveillance footage, ensuring that some objects have changed positions, and have entered or left in the second image compared to the first image. To enhance the robustness of our model, some image pairs included in the dataset exhibit no detectable differences, addressing potential corner cases. The annotations, which are natural language differences between images, were crowd-sourced via Amazon Mechanical Turk. While the images are captured using ostensibly static cameras, slight shifts in camera alignment may occur due to factors such as wind or manual adjustments. In our analysis, however, we disregard these minor misalignments and focus solely on changes in object positions.

The dataset is segmented into training, validation, and test splits, totalling 13,192 annotations. These annotations range from single sentences to multi-sentence descriptions that identify the differences between the images. Given that all data is derived exclusively from surveillance footage, it is ensured that the domain of the data is quite specific, thus leading to a relatively limited vocabulary. On average, the dataset features approximately 1.86 reported differences per image pair. Each of the images in the dataset have a fixed dimension of 224x224 pixels.

4. Approach

We chose to use a TinyLLaVA model, which projects the images into the embedding space of the backbone LLM, then predicts the caption. Our intuition is that VLMs are better able to extract information from images and produce coherent response from similarly sized networks, especially with the instruction pre-training. Therefore, we speculate that it will perform better than the original, which used a combination of LSTM and Resnet.

We chose the dataset because it was small enough to perform multiple training runs, allowing us to test multiple changes to our pipeline. The dataset was also constrained to differences on fixed cameras in parking lots, which should be easier as it isolates from differences in camera angle, lighting, etc... making it feasible that the model can learn the differences with the small number of images.

4.1. Model Selection

TinyLLaVA, as a streamlined version of the more extensive LLaVA model, was a clear choice due to its lower computational demands for fine-tuning and inference, while still maintaining high-quality text generation capabilities.

TinyLlaVA leverages the power of small-scale Large Language Models (LLM), F_θ, and Vision Encoders, V_ϕ, and a connector P_ϕ, where θ, ϕ and ϕ are the (learnable) parameters respectively, to accomplish multimodal tasks. Its architecture integrates TinyLLaMA, a light-weight variant of the popular LLaMa model, with a vision encoder, OpenAI's CLIP and a connector module. Specifically, TinyLLaVA utilizes CLIP as its vision encoder to process images into a sequence of visual patch features, which are crucial for encoding relevant features for identifying the changes in visual data.

- **Small-scale LLM** The small-scale LLM, F_θ, receives a sequence of vectors $\{h_j\}_{i=0}^{N-1}$, each within a d-dimensional text embedding space, and produces the corresponding sequence of next predictions $\{h_j\}_{i=1}^{N}$. This LLM is integrated with a tokenizer and embedding module that maps text input sequences $\{y_j\}_{i=0}^{N-1}$ to the embedding space and converts the embeddings back into text output sequences $\{y_j\}_{i=1}^{N}$.
- **Vision Encoder** The vision encoder V_ϕ processes an input image X to output a sequence of visual patch features $V = V_\phi(X)$. This encoder may utilize Vision Transformers that directly produce a sequence of patch features, or it might use CNNs that generate a grid of features, which are then reshaped to form patch features.
- **Connector** The role of the connector P_ϕ is to map the sequence of visual patch features $\{v_j\}_{j=1}^{M}$ into the text embedding space $\{h_j\}_{j=1}^{M}$. The design of P_ϕ maximizes the capabilities of both the pre-trained LLM and the vision encoder.

The language model component includes a tokenizer and embedding module that maps text input sequences into an embedding space, facilitating seamless transition from textual to visual modalities and vice versa. The vision encoder can either be a Vision Transformer, which directly produces a sequence of patch features, or a CNN that outputs grid features. In the case of CNNs, a subsequent reshape operation is employed to transform these grid features into the required sequence of patch features. The connector maps the visual patch sequences to the text embedding space. The connector is designed for effectively leveraging the capability of both the pre-trained LLM and vision encoder.

Overall, TinyLLaVA's architecture not only ensures efficient processing but also supports detailed and contextually relevant text generation, making it highly effective for applications in surveillance and monitoring where accuracy and speed are paramount.

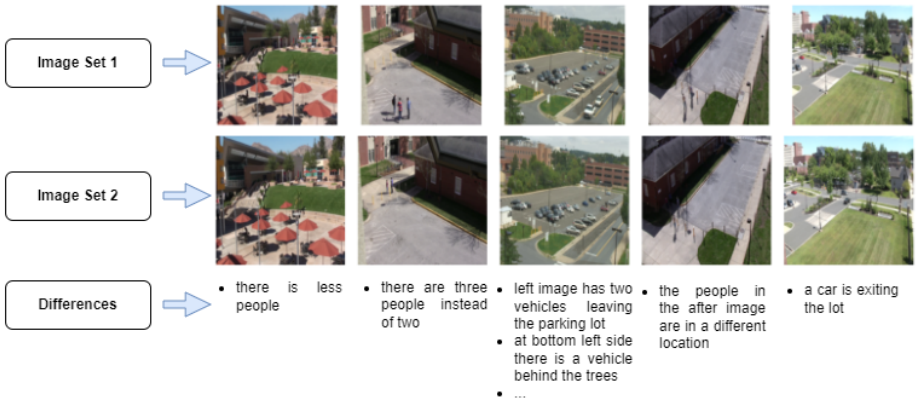


Figure 2. Sample image pair with the corresponding caption stating their differences

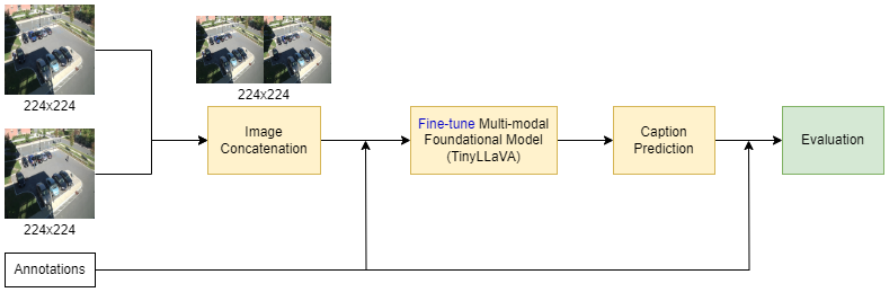


Figure 3. Spot-the-difference pipeline

Type	Name	Abb.	HF path	Size
Small-scale LLM	TinyLlama	TL	TinyLlama/TinyLlama-1.1B-Chat-v1.0	1.1B
	StableLM-2	SLM	stabilityai/stablelm-2-zephyr-1_6b	1.6B
	Phi-2	Phi	microsoft/phi-2	2.7B
Vison encoder	CLIP	C	openai/clip-vit-large-patch14-336	0.3B
	SigLIP	Sig	google/siglip-so400m-patch14-384	0.4B

Figure 4. TinyLLaVA Architecture

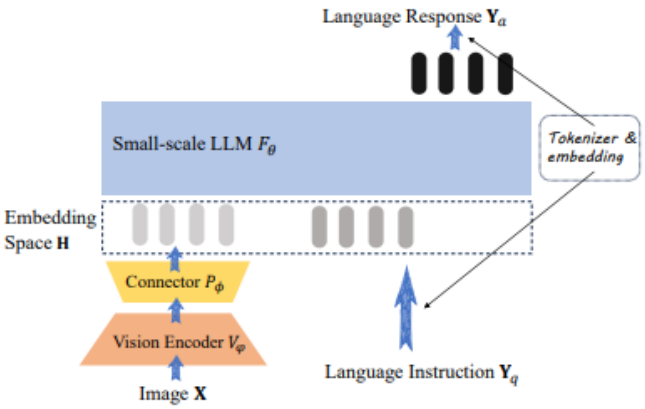


Figure 5. TinyLLaVA Framework

4.2. Fine-tuning

Fine-tuning is key to ensure model adaptation for our task of identifying differences between two video surveillance footage frames and transforming that into textual descriptions. In this regard, full fine-tuning, though the best approach, is resource-intensive and costly, as it requires modification of millions of parameters as part of training. It can also result in a large model tailored to a specific task, but lacking modularity.

Therefore, we employed Parameter Efficient Fine-tuning, PEFT, a technique designed to fine-tune models while minimizing the need for extensive resources and cost. Using PEFT, we modify only a minimal set of model parameters, while keeping the majority of the pre-trained model parameters unchanged, thus significantly reducing both computational and storage requirements. Additionally, PEFT helps prevent the problem of catastrophic forgetting. These approaches are particularly effective in scenarios with limited data and demonstrate superior generalization in out-of-domain conditions. It also enhances model portability by generating lightweight checkpoints. PEFT can be adapted across different modalities, enabling it to align perfectly with our objectives. Among PEFT tech-

niques, Low-Rank Parameter or LoRA & QLoRA are the most widely used and effective.

4.2.1 Low Rank Adaptation

LoRA can be implemented as an adapter module that augments existing neural network layers. It introduces an extra layer of trainable parameters (weights) while keeping the original parameters frozen, effectively preserving the learned parameters of the pre-trained model. The key feature of LoRA is that these additional trainable parameters are of a lower rank (dimension) compared to the original network dimensions. This structure allows LoRA to streamline and accelerate the adaptation of models for specific domain tasks.

The weights of the original model (W) are kept frozen during training, ensuring that the foundational knowledge embedded in the pre-trained model remains intact. Concurrently, a new set of parameters, W_A and W_B, are introduced. These networks employ low-rank weight vectors, with dxr and rxd dimensions, respectively. Here, 'd' denotes the dimension of the original, now-frozen network parameters, while 'r' represents the chosen low-rank or reduced dimension. The value of 'r' is crucially smaller than 'd', optimizing the training process by reducing complexity and expediting computations. The choice of 'r' is critical - a smaller 'r' makes the training process quicker and less resource-intensive, albeit at the potential cost of model performance. Conversely, a larger 'r' may slow down the training and increase costs but can improve the model's ability to manage more complex tasks. The outputs from the original network and the low-rank network are combined using a dot product, resulting in a new weight matrix of dimension 'n'. This matrix is crucial for generating the final output of the adapted model. During training, this result is compared with the expected outcomes to compute the loss function. Adjustments to W_A and W_B are made based on this loss, following standard backpropagation techniques used in neural networks. Additionally, LoRA supports scaling up AI models without a proportional increase in computational resources, making the management of growing model sizes more practical.

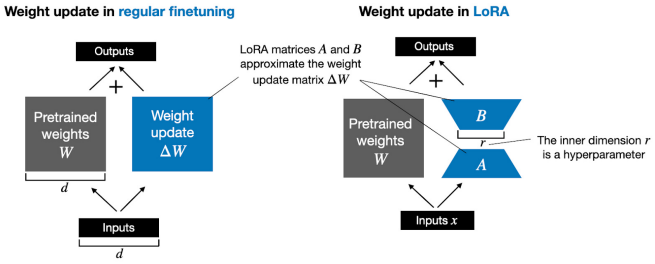


Figure 6. LoRA Mechanism

4.3. Dataset preparation

We experimented with multiple ways of formatting the training data for TinyLLaVA. Each entry in the original dataset contains 2 images, before and after, and potentially multiple sentences by different annotators describing the differences. This includes concatenating every sentence description into one, creating multi-step conversations using sentences, and creating multiple conversations based on each sentence. For the images, we tested ways of extending TinyLLaVA from single to multiple images. Our first method was concatenating images side by side into 1 large image, our second was to encode the images into tokens separately using the encoder and projector, then adding the tokens to the prompt.

While the dataset included pixel-level masks of the differences between images, we decided against using them, as that would not be generalizable to other datasets that didn't use fixed cameras.

4.4. Training pipeline

The original dataset consists of pairs images $\mathbf{X} = [(x_{0,1}, x_{0,2}), ..., (x_{n,1}, x_{n,2})]$ with accompanying captions $\mathbf{Y} = [(y_{0,1}, ..., y_{0,n}), ...]$. Using the above dataset preparation techniques, we get pairs of input **(x,y)**.

Denote the LoRA adapter L_σ and model F_θ. We then use supervised finetuning to maximize the log likelihood of the model output $\hat{\mathbf{y}}$ for each input **(x,y)**.

$$\arg\max_{\sigma} \sum_{i}^{N} I(y_i \in \hat{\mathbf{y}}) \log \left((L_\sigma \circ F_\theta)(y_i \mid P_\phi \circ V_\psi(\mathbf{x})) \right)$$

where N is the sequence length of $\mathbf{y}$.

5. Evaluation

5.1. Experiment design

We used a TinyLLaVA-1.5B model, with the AdamW optimizer. We fine-tuned a LoRA with r=32 and alpha=64. We experimented with training on more epochs, but found no noticeable improvements compared to 1 epoch. Experiments include changing the format of the data and how the image pairs are encoded.

Our research questions are:

- Which dataset format would be best, and whether it would affect model performance.
- Which image pair encoding method works best? (While encoding images separately would better retain the resolution of the original images, it's not what the model was pretrained on.)

5.2. Metrics

To evaluate our model performance, we need to calculate the semantic similarity between the ground truth annotations and the textual difference captured by the model. In [3], the authors have explored Bleu 1/2/3/4, Meteor, Cider, Rouge-L and Perplexity scores to evaluate model performance.

BLEU Scores are a popularly used metric for evaluating the quality of text that has been machine-translated from one natural language to another. It measures the correspondence between a machine's output and that of a human. BLEU-1, 2, 3, and 4 refer to the n-gram precision scores with unigram, bigram, trigram, and quad-gram, respectively. It also has a penalty associated with too-short translations. The score ranges from 0 to 1, where 1 is a perfect match with the reference translation.

METEOR is another metric suitable for evaluating machine translation tasks. Unlike BLEU, which primarily focuses on precision, METEOR accounts for both precision and recall, thereby balancing both aspects. It also considers synonyms and stemming, aligning more closely with human judgment by evaluating the adequacy and fluency.

CIDER was specifically designed for scoring image captioning systems. It evaluates the similarity of generated text to reference texts by considering the consensus between multiple references. The metric uses Term Frequency-Inverse Document Frequency (TF-IDF) weighting for each n-gram to capture the importance of n-grams according to their commonness across captions.

Perplexity is a measurement of how well a probability distribution or probability model predicts a sample. In the context of language models, it is used to quantify the uncertainty of the language model in predicting (or generating) the next token in a sequence. A lower perplexity score indicates that the model is more certain about its predictions, implying better performance.

ROUGE-L focuses on the longest common subsequence between the generated text and the reference texts, capturing both the sequence-based match and content overlap. This metric computes scores based on the longest matching sequence of words using recall, precision, and an F-measure. This emphasis on sequences ensures that the evaluation metric considers the order of words, which is crucial in understanding the coherence and fluency of the generated sentences. It rewards texts that cover the most important content of the reference texts. Additionally, ROUGE scores, including ROUGE-L, correlate well with human judgments of quality in generated texts. This correlation also allows it to produce human-like text outputs.

Each of these metrics provides a different lens through which to assess the performance of language generation models, reflecting various aspects of linguistic accuracy and fluency.

5.3. Baseline models

We compare the performance of our model with the baseline models published in [3].

- CAPT-MASK Model: Similar to the CAPT model, but includes a masking mechanism that utilizes the union of all cluster masks in the corresponding images. This approach helps focus the model's attention on specific areas identified as clusters, enhancing the relevance of the generated text to the highlighted changes.
- CAPT-MULTI Model: An extension of the CAPT model, designed to generate multi-sentence descriptions. It concatenates sentences in an arbitrary order, aiming to provide a comprehensive description covering multiple identified differences within the images.
- Nearest Neighbor (NN) Baseline: This method involves using the nearest neighbour approach to select the most similar annotation from the training set as the output. For single-sentence generation (NN), it randomly picks one sentence from the annotation of the closest training example, based on feature similarity computed via sklearn's Nearest-Neighbor module.
- DDLA-UNIFORM Model: A variant of the DDLA model that incorporates a fixed uniform prior. For single-sentence generation, this model samples a cluster randomly from a uniform distribution and then generates a description based on this selection. For multi-sentence outputs, it uses heuristics to prioritize clusters by the decreasing area of their bounding boxes, attempting to order the description according to the salience of the visual changes.
- Neural Image Captioning-Based Methods: This technique involves identifying and clustering differing pixels in the image pairs, which serves as a proxy for pinpointing object-level differences. The model leverages latent discrete variables to directly align these clusters of visual differences with generated text outputs. It also integrates a learned prior that effectively models the visual salience of these clusters, enhancing the model's ability to focus on the most relevant changes.

6. Results

6.1. Qualitative Analysis

We found the model was able to describe large differences between images, such as a car or people missing. The model was also able to list multiple differences between images. This shows significant improvements over unfinetuned model, which tends to repeat sentences with minor differences not relevant to the image.

6.2. Quantitative Analysis

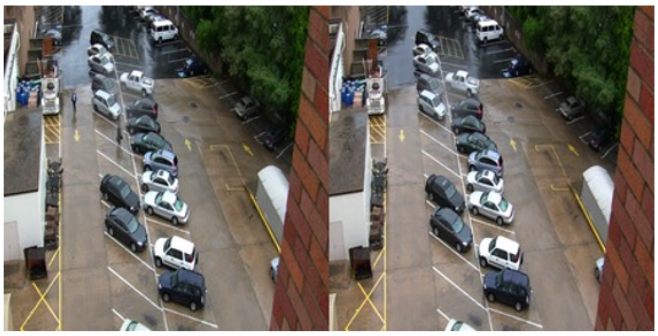
We report ROUGE-L, BLEU-4 and CIDEr scores as a metric for capturing the semantic similarity between the

```
{
    "id": "8145",
    "response": "The difference between the two images is that the parking lot is filled with cars, while the first image shows a parki
},
{
    "id": "8329",
    "response": "1. The cars are in a different position\n2. The cars are in a different position\n3. The cars are in a different posit
},
```


Figure 7. Outputs from encoding images individually. Sentences are meandering and frequently includes repeated lists.

annotated difference and the generated difference description. Our approach of using VLMs like TinyLLaVA to capture the difference between the pair of images textually proves to be better than the baseline methods as indicated in Table 1. Impressively, the model was able to outperform the original paper on CIDEr and ROUGE-L, which were also conditioned on pixel-level difference masks, while only using the images themselves.

Caption: There are no people in the after image



Caption: There is a person walking in the parking lot

Figure 8. Outputs generated by TinyLLaVA

6.3. Impact of dataset formatting

The dataset preparation method that performed the best was creating multi-step conversations from all the sentences. This is potentially because it's closest to what the instruction pre-training tasks were formatted as. We found that concatenating all sentences into 1 yielded worse results, possibly due to the wide differences in expected sequence length for the model (some annotations contained 1 short sentence, while others contain many sentences). Splitting sentences into multiple conversation yielded similar results, though slightly worse for Rouge-L.

Surprisingly, encoding the images separately did not work well, and instead, the model responds with repeated lists of responses and the responses are usually wrong. This is similar to a degraded version of the un-finetuned model, suggesting the model wasn't able to learn.

6.4. Impact of hyperparameters

We did not observe improvements from adding weight decay, or from training with more than 1 epochs. We found the optimal training weight to be around $2e-5$.

6.5. Limitations

We found that the model didn't infer the temporal order of the images. We speculate this has something to do with how the images are projected into the LLM space, as well as how the model reasons using the embeddings. We surprisingly found that encoding the images separately led to much worse performance. This could be because the model was pretrained only on single image conversations, so the projected embeddings didn't make sense. It could also be the case that there's a bug in how we embedded the image tokens into the conversation, as this required changes to the model architecture.

We also observe, perhaps due to the low capacity of the network combined with the quality of the annotations, the responses were short and usually only includes what object was different and whether the appeared or were gone.

The model attempts to count, such as "4 people appeared", but the counts are usually wrong.

6.6. Future Work

For future steps, it would be interesting to experiment with higher capacity models, such as LLaVA-7B, which uses LLaMA for its LLM component and thus promises better-quality of textual descriptions.

Employing full fine-tuning instead of Parameter-Efficient Fine-Tuning (PEFT) methods could enhance model capabilities significantly. This approach might necessitate the use of more diverse datasets that extend beyond the current limited scope, which primarily includes images

Model	Rouge-L	BLEU-4	CIDEr
NN	0.201	0.026	0.12
CAPT	0.256	0.073	0.263
CAPT-MASKED	0.271	0.078	0.285
DDLA-UNIFORM	0.247	0.064	0.250
DDLA	0.286	0.085	0.328
TinyLLaVA 1 image 1 conversation	0.304	0.0670	0.346
TinyLLaVA 1 image multi-step conversation	**0.3123**	0.0704	0.392
TinyLLaVA 1 image multiple conversation	0.3035	0.0700	**0.395**
TinyLLaVA concatenated encoding with multi-step conversation	0.1851	0.0213	0.021

Table 1. Evaluation scores for various models

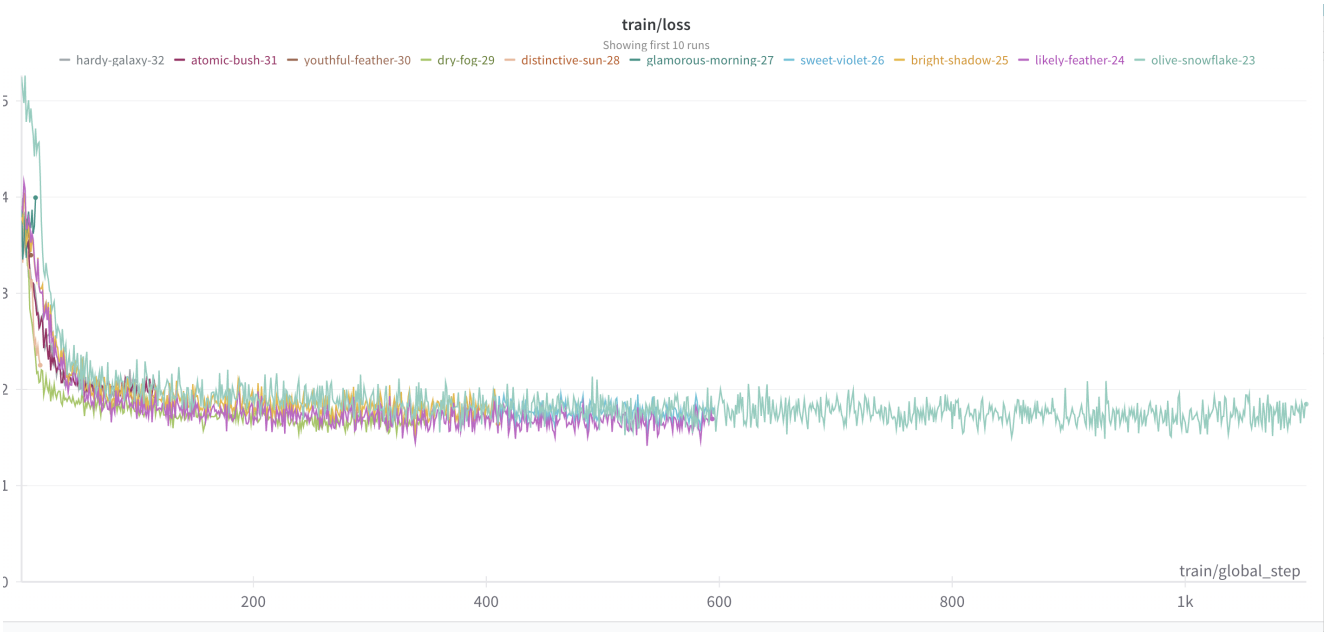


Figure 9. Loss graph. We did not observe improvements from additional epochs.

from a few parking spots with brief annotations. To further improve the model's applicability to security-related tasks, it would be beneficial to gather data from high-security zones where activities might pose potential security threats. Collecting such data would provide an extensive dataset, thus allowing a more in-depth exploration of security-based applications for our project, ensuring that the model is robust and effective in real-world scenarios.

The image encoding method could also be improved, while encoding images separately didn't work here, it's possible that it will work with larger models. Enhancing our encoding techniques could more accurately capture temporal discrepancies between images. As illustrated in Figure 1, the output incorrectly indicates that the blue truck is gone. However, if we recognize that the first image precedes the second in time, the correct caption should indicate that the blue truck has appeared. Enhancing our encoding strategies will ensure these temporal distinctions are more effectively represented.

To refine our model, we plan to employ Reinforcement Learning from Human Feedback (RLHF). This method is particularly valuable given our focus on security applications, where precise and contextually appropriate responses are crucial. RLHF not only promises more tailored outputs but also supports ethical surveillance practices by helping prevent misuse.

Furthermore, we propose the exploration and implementation of qLoRA to optimize our fine-tuning processes. This approach should streamline fine-tuning, making it more efficient and effective in adapting our model to specific tasks.

References

[1] Alexander Black, Jing Shi, Yifei Fai, Tu Bui, and John P. Collomosse. Vixen: Visual text comparison network for image

difference captioning. In *AAAI Conference on Artificial Intelligence*, 2024. 2

[2] Xinlei Chen and C. Lawrence Zitnick. Learning a recurrent visual representation for image caption generation, 2014. 2

[3] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images, 2018. 2, 6

[4] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Video storytelling: Textual summaries for events. *IEEE Transactions on Multimedia*, 22(2):554–565, 2020. 2

[5] Hao Tan, Franck Dernoncourt, Zhe Lin, Trung Bui, and Mohit Bansal. Expressing visual relationships via language, 2019. 2

[6] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models, 2024. 1