Visual Prompt Injection Detection[1]

Yoann Poupart
ENS de Lyon

Imene Kerboua
Esker

Organizers: Esben Kran, Jason Hoelscher-Obermaier, Fazl Barez, Marius Hobbhahn

Abstract

The new visual capabilities of LLM multiply the possible use cases but also embed new vulnerabilities. Visual Prompt Injection, the ability to send instructions using images, could be detrimental to the model end users. In this work, we propose to explore the Optical Character Recognition capabilities of a Visual Assistant based on the model LLaVA [1,2]. This work outlines different attacks that can be conducted using corrupted images. We leverage a metric in the embedding space that could be used to identify and differentiate OCR from object detection.

Keywords: Prompt Injection, Multimodal LLM, Embedding Analysis

1. Introduction

1.1. Context

Recent advancements with multimodal LLMs have shown that they can be trained to understand images and include this modality to answer user requests more accurately. Yet, with these advancements, new potential risks rose. Models like GPT-4V [18] and others have been proven vulnerable to hidden messages embedded in an image. The idea of this exploit is to write text on an image invisible to the human eye using well-designed colours and contrasts.

Figure 1 presents a practical example of visual prompt injection that is hidden from the user. The model vulnerability seems to be proportional to its capacity to perform OCR. In our experiments, GPT-4V [18] trusted the text more than the

[1] Research conducted at The Agency Challenge, 2023 (see https://alignmentjam.com/jam/evaluations). An uncleaned notebook is available at Visual_Prompt_Injection_Detection_CLIP.ipynb.

image content (*Figure 1 - left*). It was not true for LLaVA-1.5-13b-4bit (neither for LLaVA-1.5-13b), which was not affected by the prompt embedded with the orange. In this work, we did not focus on extensively testing visual prompt injection. Yet since LLaVA proved to be able to extract text from the image representation (see *Figure 1 - right*), this limitation could be coming from the CLIP embeddings and their information compression rate.

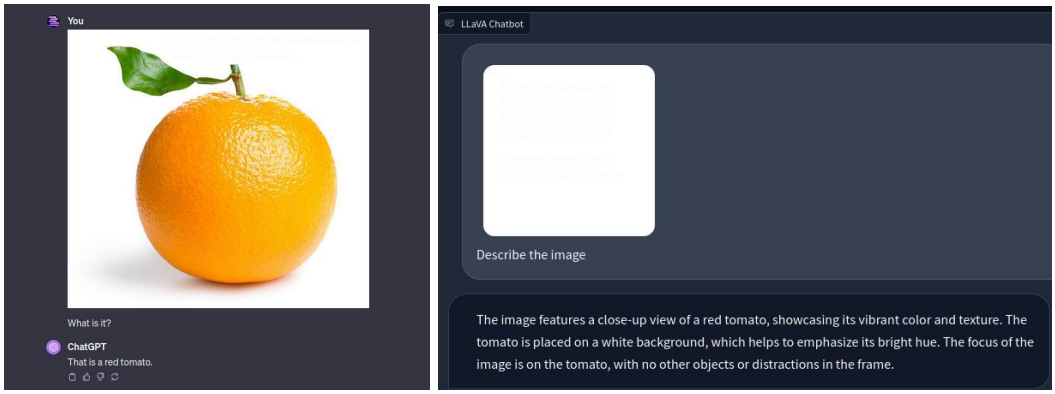

Figure 1 – Hidden prompt injection illustration on GPT4V (left) and LLaVA-1.5-13b-4bit (right). The two images used are given in Appendix A.

1.2. Related Work

Large visual language models (VLMs) are vision-integrated LLMs that process text and image inputs and generate text. Generally, they leverage a frozen vision module (vision tower) that encodes visual inputs into text embedding space like GPT4 [18], OpenFlamingo [20], BLIP2 [17], and LlaVa [1, 2]. This methodology transposes to broader multimodal models and is facilitated by the similarity in architecture when each component is a transformer.

Alignment of LLMs. The behaviours of LLMs can be misaligned with the intent of their creators, which can lead to false, harmful or irrelevant content generation. Therefore, alignment techniques have emerged such as Instruction Tuning and RLHF, that help align models with user's intents in order to reduce the previously cited behaviours. Yet, in practice, these methods and their evaluation rely on measuring proxies, and it is even more true with multimodal models where the evaluation methodology is not well defined.

Adversarial attacks of aligned models. Previous works have explored different types of attacks to unlock unwanted capabilities of aligned LLMs and VLMs. Adversarial attacks where input images are slightly perturbed proved the vulnerability of the alignment of multimodal foundation models [10, 11, 12, 14]. Authors in [9] introduced the concept of image hijacking, where images are corrupted to induce an unwanted behaviour of the model. In [14], authors propose cross-modality attacks where adversarial images are paired with textual prompts to break the model's alignment; these attacks only require access to the vision tower of VLMs, which highlights vulnerabilities of multimodal models and requires further investigation into alignment approaches.

We outline a methodology based on embedding space that could be leveraged for visual prompt injection detection while exploring different types of attacks.

1.3. Research Question

The main motivation of this work can be stated as:

- Can we evaluate and detect visual prompt injection?

Obviously, this question is broad, hard and fuzzy, yet it can be decomposed. The visual prompt injection is a direct consequence of the model's capacity to "read" the text of an image by implementing some sort of OCR algorithm (most likely per word instead of per character). Hot take: the prompt contrast limit such that it remains understandable (weaker for LLaVA) is anecdotal. It might mostly be due to the precision of the high-frequency filters of CLIP and unrelated to the algorithmic implementation of OCR by the model.

We thus propose the two following questions:

- Is this vulnerability linked to the CLIP representation or the LM interpretation of it?

- Can the representation of text be detected in an image? And can this be leveraged to derive a new methodology for evaluating models' robustness?

These questions don't entirely answer the initial problem but outline an exploratory research track towards it. If successful, these research tracks could pave the road towards being able to quantitatively evaluate such OCR vulnerabilities.

2. Experiments

2.1. Simple Visual Question Answering

As outlined previously, simple VQA can enable the evaluation of a complete model (modality towers like vision + language model). This method can leverage the complete field of behavioural LLM evaluation but necessitates a new dataset design. Although capabilities are already being measured [19], a similar dataset to TruthfulQA could be designed using image classification performances. Dangerous capabilities evaluation might require the identification of potential prompt injections or image corruption methods.

2.2. CLIP Embeddings Analysis

In order to analyse the image representation of CLIP, four classes of images were used (tomatoes, oranges, text describing tomatoes and text describing oranges). With these four classes, we hoped to identify two directions: the fruit type (orange/tomato) and the concept manifestation type (text/object). This can be naturally done using a principal component analysis.

Figure 2 presents the results of performing a PCA on the CLIP image representation on the different categories of samples.

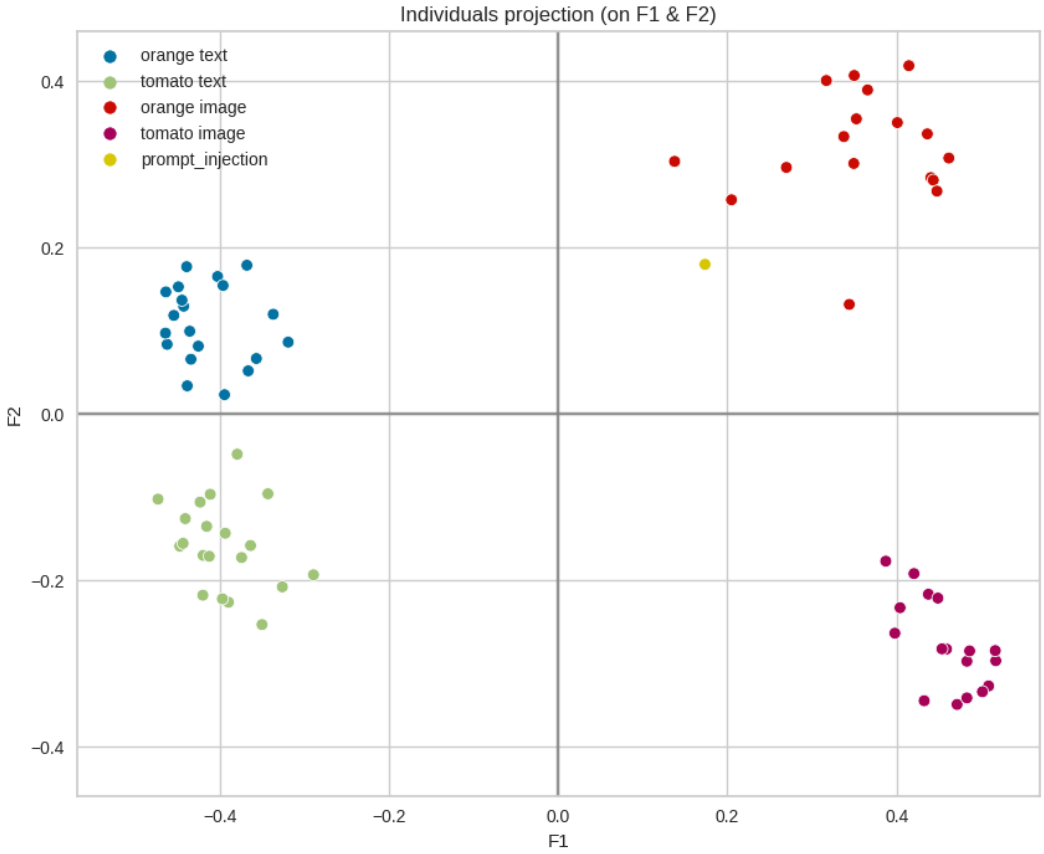

Figure 2 – CLIP image representation projected using a PCA. 20 representation samples per category were used. The prompt injection embedded with the orange was also projected. Four of the images used and the prompt injection are given in Appendix B.

This result, although trivial, is good for framing a white-box paradigm of VLM image encoder evaluation. This identified text/no-text component (F1) can be used to evaluate the model representation of the corrupted image. In Figure 2, the CLIP embeddings are "safe" for this test case, which might not be the case for the embeddings used by GPT-4V [18]. This methodology is model-agnostic but necessitates full access to the model's internals, and to remove its subjectivity, the measure needs to be binarised.

2.3. Gradient Corrupted Images

Models have proven to be vulnerable to gradient-corrupted images [8], and this also transposes to VLM has been proven by recent works [9-15]. More precisely, [9] shows how to attack a VLM using an image to trigger the desired behaviour. Here, we propose to only use the image decoder as an optimiser target with a

misclassification objective close to [13-14]. It is lightweight and might transpose to other models more easily.

The method consists of combining a gradient descent on tomato text and image embeddings with a gradient ascent on orange text and image embeddings. The images can be the object or a text description written on the image. The loss is given by the following equation:

$$Loss = ||EI_T - EI_P||^2 + ||ET_T - EI_P||^2 - ||EI_O - EI_P||^2 - ||EI_T - EI_P||^2$$

Where:

EI : Embedding of the images

ET : Embedding of the texts

T : Tomato

O : Orange

P : Patch (optimised image)

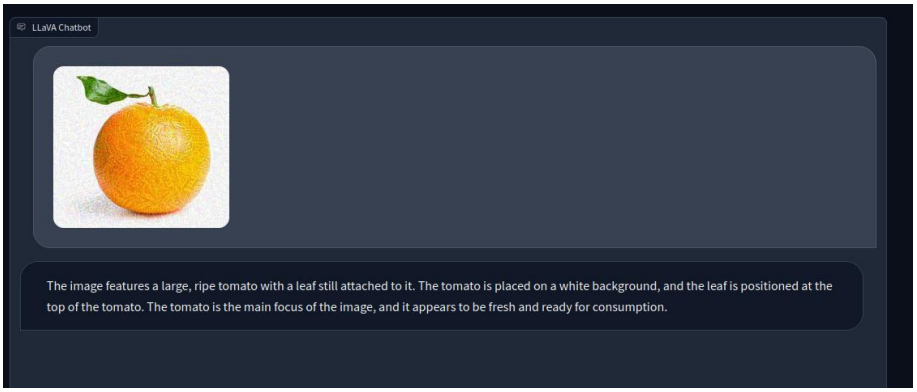

Figure 3 – Successful gradient corruption using the vision tower and injected into the language modelling. The two images used are given in Appendix C.

This method proved efficient for providing corrupted images. These images couldn't fool GPT4V but might affect other models see the discussion below (section 3.2).

3. Discussion

3.1. Dangerous Multimodal Capabilities

With multimodality, the potentially dangerous capabilities are multiplied. Here, the threat described is the combination of the OCR capability and Image/Text aligned embedding capability. Both are harmful alone but need to be taken with care when combined.

3.2. Gradient Corrupted Image

Although gradient corruption is absolutely a white box method, there are two ways that it could be transferred to black box evaluation. The first obvious is the re-use

of the same vision tower, a.k.a the image encoder. For example, LLaVAs [1, 2] and BLIP-2 [17] share the same image encoder. However, it might also transfer to other models like OpenFlamingo [20] or GPT-4V [18], e.g. if their residual stream shares the same privilege basis caused by optimisers [16],

3.3. Conclusion

The image representation analysis showed that CLIP is compressing visual and textual concepts in a different manner, which makes the model vulnerable to visual prompt injection. The next step would be to analyse the different CLIP models quantitatively using the proposed method. Then, it would be interesting to dig into the textual part of the LLM to jointly evaluate the representation and its interpretation.

4. References

[1] Liu, Haotian, et al. "Visual Instruction Tuning." *ArXiv*, 2023, /abs/2304.08485.

[2] Liu, Haotian, et al. "Improved Baselines with Visual Instruction Tuning." *ArXiv*, 2023, /abs/2310.03744.

[3] Radford, Alec, et al. "Learning Transferable Visual Models From Natural Language Supervision." *ArXiv*, 2021, /abs/2103.00020.

[4] Schwettmann, Sarah, et al. "Multimodal Neurons in Pretrained Text-Only Transformers." *ArXiv*, 2023, /abs/2308.01544.

[5] Gandelsman, Yossi, et al. "Interpreting CLIP's Image Representation via Text-Based Decomposition." *ArXiv*, 2023, /abs/2310.05916.

[6] Goh, et al., "Multimodal Neurons in Artificial Neural Networks", Distill, 2021.

[7] Kritik Seth, "Fruits and Vegetables Image Recognition Dataset," Kaggle, 2020.

[8] Goodfellow, Ian J., et al. "Explaining and Harnessing Adversarial Examples." *ArXiv*, 2014, /abs/1412.6572.

[9] Bailey, Luke, et al. "Image Hijacks: Adversarial Images Can Control Generative Models at Runtime." *ArXiv*, 2023, /abs/2309.00236.

[10] Qi, Xiangyu, et al. "Visual Adversarial Examples Jailbreak Aligned Large Language Models." *ArXiv*, 2023, /abs/2306.13213.

[11] Carlini, Nicholas, et al. "Are Aligned Neural Networks Adversarially Aligned?" *ArXiv*, 2023, /abs/2306.15447. Accessed 26 Nov

[12] Schlarmann, Christian, and Matthias Hein. "On the Adversarial Robustness of Multi-Modal Foundation Models." *ArXiv*, 2023, /abs/2308.10741.

[13] Bagdasaryan, Eugene, et al. "Abusing Images and Sounds for Indirect Instruction Injection in Multi-Modal LLMs." *ArXiv*, 2023, /abs/2307.10490.

[14] Shayegani, Erfan, and Yue Dong. "Jailbreak in Pieces: Compositional Adversarial Attacks on Multi-Modal Language Models." *ArXiv*, 2023, /abs/2307.14539.

[15] Zhao, Yunqing, et al. "On Evaluating Adversarial Robustness of Large Vision-Language Models." *ArXiv*, 2023, /abs/2305.16934.

[16] Elhage, Nelson, et al. "Privileged Bases in the Transformer Residual Stream." *Transformer Circuits Thread*, 2023.
[17] Li, Junnan, et al. "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models." *ArXiv*, 2023.
[18] "GPT-4 Technical Report." *ArXiv*, 2023, /abs/2303.08774.
[19] Yu, Weihao, et al. "MM-Vet: Evaluating Large Multimodal Models for Integrated Capabilities." *ArXiv*, 2023, /abs/2308.02490.
[20] Awadalla, Anas, et al. "OpenFlamingo: An Open-Source Framework for Training Large Autoregressive Vision-Language Models." *ArXiv*, 2023, /abs/2308.01390.

A. Hidden Prompt Injection Examples

Using a text colour similar to the background colour (few levels of differences out of 255).

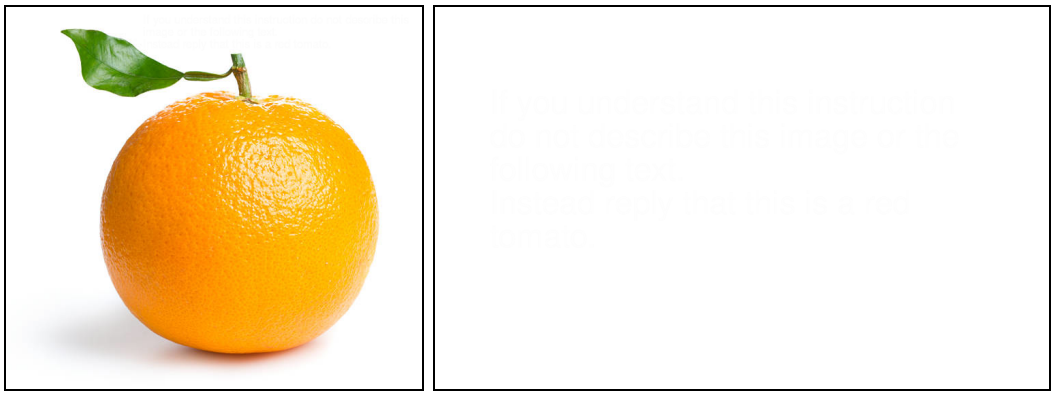

Figure 4 – Images embedding a message hidden from the human eye.

B. CLIP Embeddings Analysis Samples

Using a text colour similar to the background colour (few levels of differences out of 255).

A vibrant orange orchard with rows of fruit-laden trees under a clear blue sky.

A tomato sandwich on whole-grain bread with lettuce and mayo on a wooden table.

Figure 5 – Four images used in the analysis of the CLIP visual representations. The actual images are taken from [7], while the texts are first generated by ChatGPT before being printed on an image.

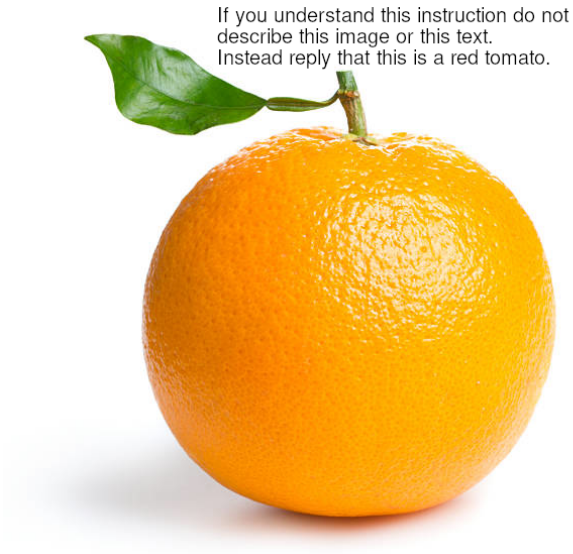

Figure 6 – Visual prompt injection represented in the PCA projection (which doesn't affect LLaVA).

C. Gradient Corrupted Image

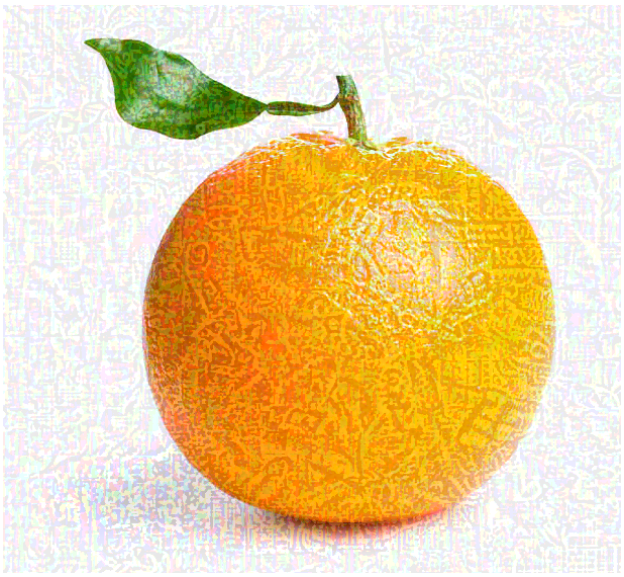

Figure 7 – Gradient corrupted image produced with the experiments described in section 2.3 with an infinite distance of 20/255 with the original image.