

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014

Abstract

Large multimodal models (LMMs) hold substantial promise across various domains, from personal assistance in daily tasks to sophisticated applications like medical diagnostics. However, their capabilities have limitations in the video game domain, such as challenges with scene understanding, hallucinations, and inaccurate descriptions of video game content, especially in open-source models. This paper describes the development of VIDEOGAMEBUNNY, a LLaVA-style model based on Bunny, specifically tailored for understanding images from video games. We release intermediate checkpoints, training logs, and an extensive dataset comprising 185,259 video game images from 413 titles, along with 389,565 image-instruction pairs that include image captions, question-answer pairs, and a JSON representation of 16 elements of 136,974 images. Our experiments show that our high quality game-related data has the potential to make a relatively small model outperform the much larger state-of-the-art model LLaVa-1.6-34b (which has more than 4x the number of parameters). Our study paves the way for future research in video game understanding on tasks such as playing, commentary, and debugging. Code and data are available at: <https://videogamebunny.github.io/>

1. Introduction

The video game industry is projected to be valued at \$321 billion by 2026 [12, 38] and continues to contribute more to the global economy. Generative Artificial Intelligence (GenAI) [41] is rapidly spreading across various sectors, disrupting the ways in which many traditional tasks are executed [6, 11]. In the realm of gaming, GenAI has the potential to enhance many aspects, such as providing a better in-game experience by generating more realistic conversations with non-playable characters (NPCs) [34] or better graphical assets [8, 17, 36, 55].

Large language models (LLMs) and large multimodal models (LMMs) represent advancements in GenAI with the

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

Anonymous WACV Applications Track submission
Paper ID 182

VIDEOGAMEBUNNY understands game context

Question: Are there any visible glitches or errors in the game environment?

VIDEOGAMEBUNNY: (D) No, there are no apparent glitches. ✓
Bunny: (B) Yes, the glowing orb is clipping through the counter. ✗
LLaVA-1.6-34b: (C) Yes, the 'Additional Download' progress bar seems stuck. ✗

Figure 1. VIDEOGAMEBUNNY is a model specifically fine-tuned on video game content, enabling it to understand game contexts and respond to related questions more accurately.

potential to function as vision assistants and solve complex problems across various domains [1, 5, 45]. In video games, LMMs can offer significant benefits for both in-game experiences and game development. In-game, LMMs can serve as vision assistants, enhancing players' experiences by guiding them through tasks like crafting new items [32]. They also have the potential to narrate the game, summarize events, and highlight critical gameplay moments [4]. For game development, LMMs have the potential to assist in detecting bugs [42], creating bug reports, and deploying automated in-game bots that interact with the environment [44]. These applications require robust models capable of understanding game content.

Despite advances and promises, existing LMMs, particularly open-source models, encounter challenges in accurately understanding game content, such as scenes and

108

world physics [42] (e.g., see Fig. 1).

109
110
111
112
113
114

In this study, we make the first important step towards addressing these challenges by releasing a suite of datasets specifically designed for video game content and introducing VIDEOGAMEBUNNY, a model trained for video game content understanding. Our study centers on the following research questions:

115
116
117
118
119

(RQ1) Which type of data has the potential to improve the model’s performance?

120
121

(RQ2) Which data type mixture strategy improves the model’s performance the most?

122
123
124

(RQ3) How does VIDEOGAMEBUNNY perform compared to state-of-the-art (SOTA) open-source models on game understanding tasks?

125
126
127
128
129
130
131

Our main contributions are as follows:

132
133
134

1. We release VIDEOGAMEBUNNY, a model specifically fine-tuned for video game question-answering tasks.
2. We release a suite of datasets containing 185,259 video game images from 413 games, featuring various game-play elements and graphical styles. Our datasets include 389,565 image-instruction pairs with captions, question-answering tasks, and JSON representations of images (see Sec. 4).
3. We conduct experiments to demonstrate the effectiveness of different instruction datasets and their impact on the model’s performance (see Sec. 6).
4. We release a replication package containing the training logs and intermediate checkpoints at <https://videogamebunny.github.io/>.

135
136
137

2. Background and Related Work

138
139
140
141

2.1. Large multimodal models

142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

Large multimodal models (LMMs) enhance large language models (LLMs) by incorporating additional modalities such as images or audio, enabling them to process multimodal inputs and generate textual outputs. The role of the language model is to comprehend user instructions and produce responses based on the additional modality inputs provided. Standard approaches to create LMMs involve combining pre-trained models with different modalities via projection layers. These layers can be implemented using simple mechanisms such as multilayer perceptrons (MLP) [24, 27] or transformer layers [25]. Alternatively, a resampler module like Perceiver [3, 20, 21] or Qformer [10, 60] selectively chooses features to reduce the number of visual tokens based on the context and instruction, enhancing efficiency and maintaining performance.

157
158
159
160
161

In this study, we focus on LMMs that accept input images and text to produce responses, particularly using the LLaVA-style architecture [29], which is one of the most popular methods [14, 26, 28, 33]. This architecture employs an MLP layer to integrate vision tokens with a language

162
163
164
165
166
167
168
169
170
171
172
173
174

2.2. Instruction following data

Large models trained on massive corpora of text, such as GPT-3 [7], T5 [37], and PaLM [9], are not inherently instruction-following, meaning they do not respond to user queries. To enable these models to follow user instructions and answer queries, they usually undergo a process called instruction tuning [35, 56]. This process involves fine-tuning the models to handle specific user instructions, such as questions or commands, allowing them to respond appropriately based on the given instructions.

In the multimodal context, particularly for models that accept visual inputs, there are various types of visual instruction-following data, such as detailed descriptions, conversational style question answering (Q&A), and complex reasoning. Researchers have explored diverse approaches to generate such data, including the use of academic text-oriented visual Q&A datasets [10]. The LLaVA model [24] demonstrated that leveraging a strong text-only LLM and an image dataset annotated with object names and bounding box information can be converted into effective visual instruction-following data.

2.3. LLMs and LMMs in video games

LLMs have shown strong promise for integration with games for a wide range of tasks, from content creation to game-playing agents [13, 19, 30, 39, 43, 47–50, 52, 57, 59]. Large multimodal models (LMMs) can further enhance this integration by providing richer context inputs such as images and videos to enable broader applications. Projects like Cradle [44], which focuses on playing Red Dead Redemption 2 with GPT-4V [1] showcase LMMs’ abilities to identify objects, characters, and environmental features and assist in controlling the game. Beyond gameplay, LMMs have found applications in game testing [42, 43], where they are leveraged for detecting and interpreting video game bugs.

Our study is the first to explore enhancing an LMM’s general game understanding, rather than focusing on a specific game or task. We use screenshots from 413 games, aiming to improve capabilities across various game-related tasks by developing broader game comprehension skills.

2.4. Empirical analysis of large multimodal models

Some previous studies have conducted experiments to see how different architectural components or data sources affect the general performance of large multimodal models [22, 23, 33, 46]. For example, McKinzie et al. [33] found that the input resolution of the input image plays a crucial role in improving performance, and Laurençon et al. [22] found that utilizing cross-attention between image and language is more effective than the adapter-based method.

216 We are the first to systematically investigate the impact
 217 of different instruction-following datasets and their combinations
 218 on the performance of LMMs in game understanding tasks.
 219

220 221 222 3. VIDEOGAMEBUNNY Model Architecture

223 In this section, we describe the architectural choices and
 224 configurations behind our model, VIDEOGAMEBUNNY.
 225 VIDEOGAMEBUNNY is based on Bunny [14], a family of
 226 efficient and high-performing LLMs known for their com-
 227 petitive or superior performance on various benchmarks
 228 compared to many open-source alternatives.

229 Bunny follows the same principle as LLaVA [24, 27] for
 230 the integration of image inputs. Using a shallow network of
 231 multilayer perceptrons (MLPs) as the projection layer, vi-
 232 sion embeddings extracted from a strong pre-trained vision
 233 model are processed and provided as image tokens for the
 234 language model. This technique effectively leverages pre-
 235 trained vision and language models, allowing them to work
 236 together efficiently.

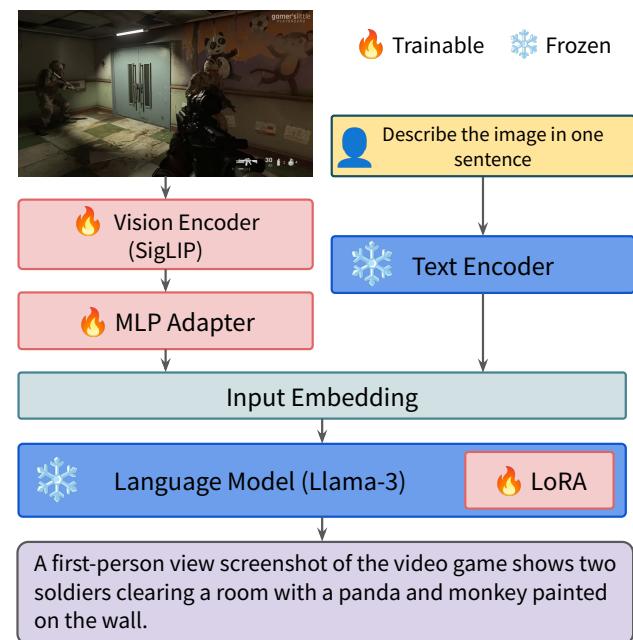
237 Bunny offers various combinations of vision and lan-
 238 guage models and supports images with resolutions up to
 239 1152×1152 pixels. For creating VIDEOGAMEBUNNY, we
 240 selected Bunny configurations that deliver the best perfor-
 241 mance [14] while being small enough to run on a consumer-
 242 grade graphics card. We use LLama-3-8B [2] as the lan-
 243 guage model and SigLIP [54] with the S² wrapper [40]
 244 for the vision encoder. The S² wrapper extracts features
 245 from an input image at various scales to form a multi-scale
 246 feature. This is potentially useful since video games of-
 247 ten contain visual elements at different scales, from tiny
 248 UI icons to large objects. A multi-scale feature could cap-
 249 ture these diverse elements. Fig. 2 shows the architecture of
 250 VIDEOGAMEBUNNY.

251 252 253 4. Instruction-following Data for Video Game Content

254 One of the main challenges limiting the ability of open-
 255 source models to generalize effectively to video game con-
 256 tent is the lack of instruction-following data specific to
 257 video games in public datasets. Our goal is to collect
 258 game-specific data to address this challenge. In this sec-
 259 tion, we explain the process of collecting and generating
 260 game-specific instruction-following data.

261 262 263 4.1. Video game images

264 We collect images from YouTube by searching for *game-*
 265 *play walkthroughs* with *Full-HD, 4K, and 8K* quality. These
 266 high-resolution videos ensure that downsampled frames re-
 267 tain more information and details compared to lower quality
 268 videos. We randomly sample frames from the downloaded
 269 videos and label them with the corresponding game name.



291 Figure 2. Architecture overview of VIDEOGAMEBUNNY. An im-
 292 age input and a textual instruction are fed into the language model
 293 to produce a response. The image is passed through a separate
 294 pre-trained vision encoder and a projection layer to align the em-
 295 bedding space between the two models. and icons show
 296 trainable and frozen layers respectively

297 In total, our dataset contains 185,259 images from 413 dif-
 298 ferent video games, encompassing various genres, graphic
 299 styles, and gameplay mechanics. Fig. 3 shows some sample
 300 images from our dataset, and Fig. A5 shows the distribution
 301 of images per game.

302 303 304 4.2. Generating instructions

305 Following previous studies [27, 29, 51], we employ an-
 306 other robust model to generate instructions in the form of
 307 user queries and responses for images in our dataset. We
 308 categorize the instructions into four types: *short captions*
 309 (70,673 samples), *long captions* (70,799 samples), *image-*
 310 *to-JSON* (136,974 samples) and *image-based question an-*
 311 *swering* (81,122 samples). In this section, we explain how
 312 we generate each type of instructions. Fig. 4 shows an
 313 overview of the data generation process.

314 315 316 4.2.1 Image captioning

317 Image captioning is a basic form of instruction-following
 318 that generates a description of the input image. An im-
 319 age caption can be short and concise, providing a high-level
 320 overview of the image, or very detailed, covering fine-grain
 321 details. Our dataset includes both forms of image caption-
 322 ing to meet user queries, whether they seek a detailed cap-



Figure 3. Our dataset includes sample video game images that showcase a wide range of characters, environments, mechanics, camera viewpoints, and artistic styles. These styles vary from western to contemporary and futuristic, and from realistic to fantasy settings.

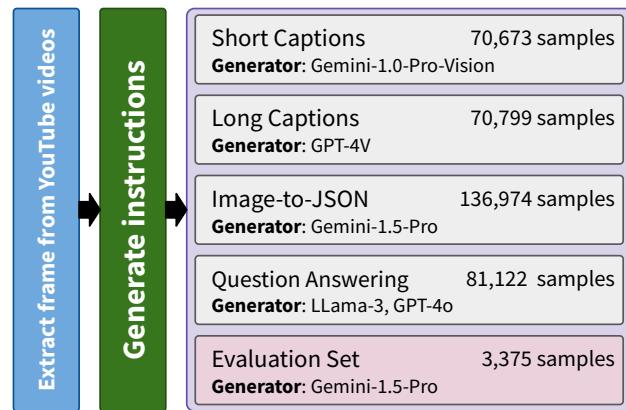


Figure 4. Overview of the dataset generation process.

tion or a short summary. In addition, it includes a structured version where the image is described in 16 predefined fields.

Short captions: We use Gemini-1.0-pro-vision to obtain short descriptions of a subset of images in our dataset, which includes 70,673 images. We use the “*Describe the image*” prompt, which generates captions typically consisting of one or a few sentences.

Long captions: While short captions provide a high-level overview of the image, it lacks many details in the image which might be useful for the user. To address this, we use GPT-4V to get detailed captions of all images in the previous section (see Fig. A1 for the used prompt).

Image-to-JSON: Another comprehensive method for describing images is converting them into a JSON format. This approach summarizes an input image into a JSON structure, with each key describing an element from the image, such as *characters in the image* or *description of Game UI*. Unlike typical captioning, this method provides a template that must be filled. If the image lacks a certain element, that part remains empty, indicating the absence of that element in the image. This ensures a more detailed and structured representation of the image content.

Another benefit of describing an image in JSON is that this structured representation facilitates integration with other systems. JSON is a widely adopted format for sharing information between different software systems. Summarizing the image as JSON can help integrate LMMs in other

systems, such as software testing pipelines, potentially verifying the game output and ensuring that the visual output has the desired properties and information.

To create the image-to-JSON dataset, we use Gemini-1.5-Pro with instructions (Fig. A2) to convert a given image into a JSON file with hierarchical levels of detail and information. The JSON file contains 16 elements that capture both high-level and fine-grained details of the image. These keys are chosen to capture game-specific elements from the image in isolation, which can be used for downstream applications, such as game testing. It starts with an overall summary of the image and then moves to specific aspects such as detailed character descriptions (including facial expressions and clothing), weather information, summaries of UI and player inventory, objects in the scene, and lighting and environmental effects. Tab. A1 shows the keys included in the JSON output. Our dataset contains 136,974 Image-JSON pairs. Fig. 5 shows a sample of information extracted from an image.

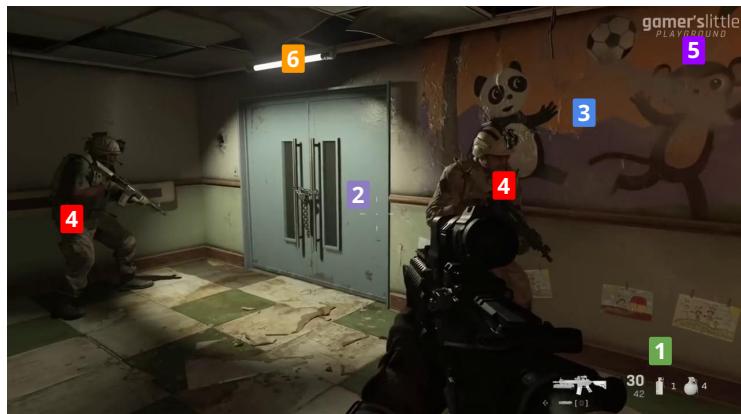
4.2.2 Question-Answering conversations

Moving beyond simple image descriptions, a general and capable model should be able to respond to user questions based on the content of the image. Below, we describe how we generate such data for each image (1) from its long caption and (2) directly from the image itself.

Llama-3-assisted visual instruction data generation: We use Llama-3 to convert long captions generated by GPT-4V into a series of question-answering conversations. This approach is similar to the original LLaVA [29] method, but instead of using an object’s name and its bounding box information, we directly utilize long captions.

While long captions provide a rich source of information, they lack the structure of question-answer formats. For example, if a caption describes a person in the image with specific details, such as clothing, an LLM can generate a question like, “*What is the color of the dress of the person in the image?*” By utilizing a strong text-only model, we can transform each caption into a multi-turn conversation between a user and an assistant.

We use Llama-3-70B to transform GPT-4V captions into question-answer conversations, with the prompt shown in Fig. A3. The prompt requires questions to directly relate to



1 The player has 30 rounds in the currently equipped magazine, 42 spare rounds, 1 grenade, and 4 of an unidentified item.

2 A set of double doors, secured by a chain and padlock

3 A whimsically painted mural on the right wall, depicting a panda bear and a monkey playfully engaged in a soccer match.

4 Two armed soldiers, clad in military fatigues, are positioned strategically within a room.

5 Watermark: Gamer's Little Playground

6 The room is dimly lit with the primary light source appearing to be a fluorescent light fixture on the ceiling.

Figure 5. Sample information extracted for the image-to-JSON dataset by Gemini-1.5-Pro. Each sample contains detailed information ranging from minor details to high-level descriptions, such as: **1** player inventory, **2** **3** details about the environment, **4** non-player characters, **5** the screenshot’s watermark, and **6** lighting.

the image description. We create 496,469 question-answer pairs for 70,232 images, grouping questions for each image into a multi-turn conversation.

Image-based question-answering: We use GPT-4o to generate questions and their answers based on an input image in a *single* prompt. In the prompt (Fig. A4), we first ask GPT-4o to examine the image and provide a detailed description of its content, then to generate relevant questions based on the content of the image and provide answers for each question. In the prompt, we emphasize that the questions should focus on understanding the image to avoid questions that might not be directly relevant to the image.

4.3. Evaluation dataset

To assess model performance on video game understanding tasks, we created a multiple-choice question evaluation set using Gemini-1.5-Pro [45]. This approach allows for an efficient comparison of various models. While Gemini-1.5-Pro offers significant advantages over open-source models for data generation, it does have limitations. We reduce noise in the generated questions as follows:

- Initial Generation:** We use Gemini-1.5-Pro to create 4,000 questions across 10 categories related to video game content understanding (see Tab. 1).

2. Quality Assessment:

- (a) Self-evaluation: We test Gemini-1.5-Pro on its own questions and found it achieves an accuracy of 84%.
- (b) Manual validation: A random sampling of questions and answers revealed a 14% error rate (incorrect or indeterminate answers).

3. Noise Reduction:

- We remove 625 samples that Gemini-1.5-Pro had answered incorrectly.
- We conduct a second manual analysis and found that the error rate dropped to 9%.

5. Experiments

In this section, we describe our experiments to explore how our collected instruction-following datasets can improve a model’s understanding of game context. We focus on three research questions:

(RQ1) Which type of data has the potential to improve the model’s performance? In addressing this question, we fine-tune Bunny using a single dataset at a time to observe overall performance trends. Since the primary goal of this experiment is to identify general trends, we fine-tune Bunny on different subset sizes for each dataset only *once*. We increase the subset size from 2K to 60K samples and stop the experiment if we observe a sharp decline in performance.

(RQ2) Which data type mixture strategy improves the model’s performance the most? We evaluate different data mixing strategies at various sizes to see how both mixture and subset size change the performance of the model. We use the following four strategies:

- Random:** We randomly sample without replacement from the combined dataset pool. This serves as a control group, using no specific selection strategy.
- Equal:** We select an equal number of samples from each dataset to ensure a balanced representation.
- Stratified:** Datasets are mixed based on *video games*, maintaining the game distribution in the final dataset. This balances game representation and ensures diverse image types. We focus on game variety rather than

Table 1. Categories of questions in our dataset, along with a sample question for each category.

Category	Description	Count
Action Understanding	Recognizing and describing the actions taking place within the image. <i>Sample: What action is the character in the foreground performing?</i>	356
Anomalies and Glitches	Identifying errors, bugs, glitches, or placeholder elements within the game environment. <i>Sample: Describe any anomalies or glitches present in the image.</i>	223
Character Analysis	Recognizing characters, understanding their roles, and interpreting their expressions and poses. <i>Sample: What is Aloy's emotional state based on her facial expression?</i>	312
Common Sense Reasoning	Understanding the image using general knowledge and everyday logic. <i>Sample: Based on the score and time remaining, which team is likely to win the match?</i>	430
Gameplay Mechanics	Understanding the rules and mechanics that govern the game. <i>Sample: What game mechanic is most likely being utilized by the player character?</i>	273
OCR and UI	Reading and interpreting on-screen text and user interface elements. <i>Sample: What is written in the caption box at the bottom of the image?</i>	334
Miscellaneous	Any other type of question that does not fit into the previous categories. <i>Sample: What material are the containers in the image primarily made of?</i>	239
Scene Understanding	Recognizing and interpreting the overall environment or setting in the image. <i>Sample: The racetrack depicted in the image is set in what type of environment?</i>	566
Small Details	Identifying and interpreting small but significant details within the image. <i>Sample: What color is the jacket worn by the character in the foreground?</i>	356
Spatial Reasoning	Testing the ability to understand spatial relationships of objects present in the image. <i>Sample: What is the spatial relationship between the two red markers visible in the image?</i>	286

instruction types. Games with insufficient samples are excluded.

4. **Weighted:** We use the three most effective datasets from RQ1: image-based question-answering (GPT-4o), long captions, and image-to-JSON. We assign weights: 30% each for GPT-4o and long captions, 40% for image-to-JSON. This prioritizes valuable datasets to assess their impact on model performance.

We fine-tune Bunny on the above dataset mixture strategies with sizes ranging from 2K to 30K. We repeat each experiment three times, using different samples for each strategy to report the mean performance and standard deviation. We stop at 30K since our smallest dataset (generated by GPT-4o) contains 10K samples, and at 30K, we will exhaust the *Equal* and *Weighted* strategies.

(RQ3) *How does VIDEOGAMEBUNNY perform compared to SOTA open-source models on game understanding tasks?* Building on insights from our experiments, we create VIDEOGAMEBUNNY, a model fine-tuned on a dataset of 50K image-instruction samples compiled from all previously introduced datasets. To assess the effectiveness of fine-tuning a smaller model on game-specific data, we evaluate VIDEOGAMEBUNNY against LLaVA-1.6-34b, a SOTA open-source model with $4.2 \times$ more parameters.

Experiment setup: We instruction tune Bunny with LoRA [18] using the PEFT [31] library. Given that Bunny has been trained on real images, we unfreeze the vision encoder (SigLIP [54]) to adapt to the diverse visual styles of different games. To prevent overfitting and memorization, we fine-tune for only one epoch in all experiments.

Given the importance of reproducibility and accessibility for all researchers, we perform all experiments on a



Figure 6. Relative performance improvement (pp) of Bunny fine-tuned on different subsets of each dataset. The image-to-JSON dataset shows a strong positive trend, while the short captions dataset degrades performance. The best performance achieved in the experiment is highlighted in bold.

single NVIDIA A100 GPU (80GB), ensuring a balance between computational power and accessibility. The total GPU hours needed to conduct all experiments, including some preliminary tests, is approximately 900 hours, which is roughly \$2,000 when using cloud providers.

6. Results

RQ1: Which type of data has the potential to improve the model's performance?

The image-to-JSON dataset has the greatest potential to improve the base model's performance. Fig. 6 shows the performance after fine-tuning Bunny using a single dataset at a time. Fine-tuning on a subset of the image-

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

Models comparison in video game understanding



Question: Based on the score and time remaining, which team is likely to win the match?

VIDEOGAMEBUNNY: (B): The blue team is likely to win ✓

Bunny: (C): The red team is likely to win. ✗

LLaVA-1.6-34b: (C): The red team is likely to win. ✗

Figure 7. VIDEOGAMEBUNNY correctly utilizes information on the HUD to answer the question, while the basemodel Bunny and the larger model LLaVA-1.6-34b fail to provide a correct answer.

7. Discussion

Potential negative impacts: Our study initiates the development of a model for understanding game content, with applications in game playing, testing, and commentary. The short-term implications for the gaming industry include enhancing the productivity of game testers and enhancing quality assurance processes. One possible negative impact of such advancement is the facilitation of in-game cheating. As AI models becomes more adept at understanding game contents, there is a risk that they could be used to create sophisticated cheating tools.

Biases and mistakes in dataset generation: In this study, we use existing models to annotate our dataset. While this approach aligns with previous studies [29, 61] and follows the same principles as the teacher-student learning paradigm [16], we acknowledge that our data may contain biases and mistakes introduced by these existing models.

Quality of images: Images in our dataset, sourced from YouTube videos, may include low-quality frames despite filtering for high-quality videos. Motion blur in video games and sampling during scene transitions can result in some blurry or less-than-ideal images.

Use of various models for annotation: In the annotation process, we utilize various models depending on their availability and cost. While this process brings diversity to the labels, it is not meant to compare the different models against each other in terms of performance.

Use of multiple-choice questions to evaluate game understanding: We use multiple-choice questions to evaluate

Models comparison in image captioning



Question: Write a short summary of the image

VIDEOGAMEBUNNY: The image is a screenshot from a video game, showing a character in mid-air, equipped with a wingsuit, flying over a mountainous landscape. The game's HUD elements are visible, including a distance marker and a weapon indicator. . . .

Bunny: A video game scene shows a man in a *green shirt* and blue pants jumping off a cliff. He is *holding a parachute* and is in mid-air. . . .

LLaVA-1.6-34b: The image appears to be a screenshot from a video game, showing a *first-person perspective* of a character in mid-air, likely after jumping or falling from a height. The character is *equipped with a parachute, which is open* and slowing their descent. . . .

Figure 8. VIDEOGAMEBUNNY provides a detailed description of video game content in the image, while other models show inaccuracies (highlighted in red). Responses are truncated to save space.

a model's game understanding, as it allows for a clear comparison. While this format has been extensively used for benchmarks [15, 53], it might not be the best proxy for game understanding. Future work needs to focus on human evaluation or the use of LLMs as judges [58].

8. Conclusion

We introduce a new instruction-following dataset, with 389,565 image-instruction pairs, specifically designed for video game understanding. We investigate the effectiveness of fine-tuning LMMs on different instruction-following dataset types and mixtures of them, and finally introduce VIDEOGAMEBUNNY, an 8B parameter model that outperforms a SOTA model, LLaVA-1.6-34b, on a game-related question answering benchmark.

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

864

References

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

- [1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 1, 2
- [2] AI@Meta: Llama 3 model card (2024), https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md 3
- [3] Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022) 2
- [4] Anonymous: While i wait for gpt-4o with updated voice capabilities, i decided to create a prototype using multiple open source models to simulate an ai commentator who can see your screen and listen to in-game dialogue. Reddit r/OpenAI (2024), https://www.reddit.com/r/OpenAI/comments/1dm6lg9/while_i_wait_for_gpt4o_with_updated_voice/, accessed: June 30, 2024 1
- [5] Anthropic: Introducing the next generation of claude (2024), <https://www.anthropic.com/news/clause-3-family-1>
- [6] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021) 1
- [7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020) 2
- [8] Chen, Y., He, T., Huang, D., Ye, W., Chen, S., Tang, J., Chen, X., Cai, Z., Yang, L., Yu, G., et al.: Meshanything: Artist-created mesh generation with autoregressive transformers. arXiv preprint arXiv:2406.10163 (2024) 1
- [9] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. Journal of Machine Learning Research **24**(240), 1–113 (2023) 2
- [10] Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P.N., Hoi, S.: Instructblip: Towards

- general-purpose vision-language models with instruction tuning. Advances in Neural Information Processing Systems **36** (2024) 2
- [11] Eloundou, T., Manning, S., Mishkin, P., Rock, D.: Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130 (2023) 1
- [12] Forum, W.E.: Gaming boomed in lockdown and market value will reach \$320bn. World Economic Forum (2023), <https://www.weforum.org/agenda/2023/01/gaming-market-value-320bn-2026/> 1
- [13] Gallotta, R., Todd, G., Zammit, M., Earle, S., Liapis, A., Togelius, J., Yannakakis, G.N.: Large language models and games: A survey and roadmap. arXiv preprint arXiv:2402.18659 (2024) 2
- [14] He, M., Liu, Y., Wu, B., Yuan, J., Wang, Y., Huang, T., Zhao, B.: Efficient multimodal learning from data-centric perspective. arXiv preprint arXiv:2402.11530 (2024) 2, 3
- [15] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300 (2020) 8
- [16] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 8
- [17] Holson, B.: Dimension hopper part 1 (Jun 2023), <https://generalrobots.substack.com/p/dimension-hopper-part-1> 1
- [18] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021) 6
- [19] Hu, S., Huang, T., Ilhan, F., Tekin, S., Liu, G., Kompell, R., Liu, L.: A survey on large language model-based game agents. arXiv preprint arXiv:2404.02039 (2024) 2
- [20] Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: International conference on machine learning. pp. 4651–4664. PMLR (2021) 2
- [21] Laurençon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S.,

- 972 Rush, A., Kiela, D., et al.: Obelics: An open web-
973 scale filtered dataset of interleaved image-text doc-
974 uments. Advances in Neural Information Processing
975 Systems **36** (2024) 2
- 976 [22] Laurençon, H., Tronchon, L., Cord, M., Sanh, V.: What matters when building vision-language models?
977 arXiv preprint arXiv:2405.02246 (2024) 2
- 978 [23] Li, B., Zhang, H., Zhang, K., Guo, D., Zhang,
979 Y., Zhang, R., Li, F., Liu, Z., Li, C.: Llava-
980 next: What else influences visual instruction tuning
981 beyond data? (May 2024), [https://llava-vl.github.io/blog/2024-05-25-llava-](https://llava-vl.github.io/blog/2024-05-25-llava-next-ablations/)
982 [next-ablations/](https://llava-vl.github.io/blog/2024-05-25-llava-next-ablations/) 2
- 983 [24] Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H.,
984 Yang, J., Naumann, T., Poon, H., Gao, J.: Llava-
985 med: Training a large language-and-vision assistant
986 for biomedicine in one day. Advances in Neural Infor-
987 mation Processing Systems **36** (2024) 2, 3
- 988 [25] Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping
989 language-image pre-training with frozen image
990 encoders and large language models. In: International
991 conference on machine learning. pp. 19730–19742.
992 PMLR (2023) 2
- 993 [26] Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan,
994 L.: Video-llava: Learning united visual representa-
995 tion by alignment before projection. arXiv preprint
996 arXiv:2311.10122 (2023) 2
- 997 [27] Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines
998 with visual instruction tuning. In: Proceedings of the
999 IEEE/CVF Conference on Computer Vision and Pat-
1000 tern Recognition. pp. 26296–26306 (2024) 2, 3
- 1001 [28] Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen,
1002 S., Lee, Y.J.: Llava-next: Improved reasoning, ocr,
1003 and world knowledge (January 2024), [https://llava-vl.github.io/blog/2024-01-30-](https://llava-vl.github.io/blog/2024-01-30-llava-next/)
1004 [llava-next/](https://llava-vl.github.io/blog/2024-01-30-llava-next/) 2
- 1005 [29] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruc-
1006 tion tuning. Advances in neural information process-
1007 ing systems **36** (2024) 2, 3, 4, 8
- 1008 [30] Ma, W., Mi, Q., Yan, X., Wu, Y., Lin, R., Zhang,
1009 H., Wang, J.: Large language models play starcraft ii:
1010 Benchmarks and a chain of summarization approach.
1011 arXiv preprint arXiv:2312.11865 (2023) 2
- 1012 [31] Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y.,
1013 Paul, S., Bossan, B.: Peft: State-of-the-art parameter-
1014 efficient fine-tuning methods. [https://github.](https://github.com/huggingface/peft)
1015 [com/huggingface/peft](https://github.com/huggingface/peft) (2022) 6
- 1016 [32] Mashable: You can use microsoft's copilot ai chat-
1017 bot to learn how to play 'minecraft'. Mashable
1018 (2024), <https://mashable.com/article/microsoft-build-2024-copilot-minecraft/>, accessed on June 20, 2024 1
- 1019 [33] McKinzie, B., Gan, Z., Fauconnier, J.P., Dodge,
1020 S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng,
1021 F., Weers, F., et al.: Mml: Methods, analysis
1022 & insights from multimodal llm pre-training. arXiv
1023 preprint arXiv:2403.09611 (2024) 2
- 1024 [34] NVIDIA: Nvidia ace for games sparks life into vir-
1025 tual characters with generative ai (2023), <https://nvidianews.nvidia.com/news/nvidia-ace-for-games-sparks-life-into-virtual-characters-with-generative-ai>, accessed: June 27, 2024 1
- 1026 [35] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wain-
1027 wright, C., Mishkin, P., Zhang, C., Agarwal, S.,
1028 Slama, K., Ray, A., et al.: Training language models
1029 to follow instructions with human feedback. Advances
1030 in neural information processing systems **35**, 27730–
1031 27744 (2022) 2
- 1032 [36] Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv
1033 preprint arXiv:2209.14988 (2022) 1
- 1034 [37] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang,
1035 S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring
1036 the limits of transfer learning with a unified text-to-
1037 text transformer. Journal of machine learning research
1038 **21**(140), 1–67 (2020) 2
- 1039 [38] Research, G.V.: Video game market size, share and
1040 growth report, 2030. Grand View Research (2023),
1041 <https://www.grandviewresearch.com/industry-analysis/video-game-market>
1
- 1042 [39] Shao, X., Jiang, W., Zuo, F., Liu, M.: Swarm-
1043 brain: Embodied agent for real-time strategy game
1044 starcraft ii via large language models. arXiv preprint
1045 arXiv:2401.17749 (2024) 2
- 1046 [40] Shi, B., Wu, Z., Mao, M., Wang, X., Darrell, T.: When
1047 do we not need larger vision models? arXiv preprint
1048 arXiv:2403.13043 (2024) 3
- 1049 [41] Stokel-Walker, C., Van Noorden, R.: The promise
1050 and peril of generative ai. Nature **614**(7947), 214–216
1051 (2023). <https://doi.org/10.1038/d41586-023-00340-6>
1

- 1080 [42] Taesiri, M.R., Feng, T., Bezemer, C.P., Nguyen,
1081 A.: Glitchbench: Can large multimodal models de-
1082 tect video game glitches? In: Proceedings of the
1083 IEEE/CVF Conference on Computer Vision and Pat-
1084 tern Recognition. pp. 22444–22455 (2024) 1, 2 1134
1085
1086 [43] Taesiri, M.R., Macklon, F., Wang, Y., Shen, H., Beze-
1087 mer, C.P.: Large language models are pretty good
1088 zero-shot video game bug detectors. arXiv preprint
1089 arXiv:2210.02506 (2022) 2 1135
1090
1091 [44] Tan, W., Ding, Z., Zhang, W., Li, B., Zhou, B., Yue,
1092 J., Xia, H., Jiang, J., Zheng, L., Xu, X., et al.: Towards
1093 general computer control: A multimodal agent for
1094 red dead redemption ii as a case study. arXiv preprint
1095 arXiv:2403.03186 (2024) 1, 2 1136
1096
1097 [45] Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac,
1098 J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M.,
1099 Hauth, A., et al.: Gemini: a family of highly capable
1100 multimodal models. arXiv preprint arXiv:2312.11805
1101 (2023) 1, 5 1137
1102
1103 [46] Tong, S., Brown, E., Wu, P., Woo, S., Middepogu, M.,
1104 Akula, S.C., Yang, J., Yang, S., Iyer, A., Pan, X., et al.:
1105 Cambrian-1: A fully open, vision-centric exploration
1106 of multimodal llms. arXiv preprint arXiv:2406.16860
1107 (2024) 2 1138
1108
1109 [47] Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao,
1110 C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An
1111 open-ended embodied agent with large language mod-
1112 els. arXiv preprint arXiv:2305.16291 (2023) 2 1139
1113
1114 [48] Wang, Z., Cai, S., Chen, G., Liu, A., Ma, X., Liang, Y.:
1115 Describe, explain, plan and select: Interactive plan-
1116 ning with large language models enables open-world
1117 multi-task agents. arXiv preprint arXiv:2302.01560
1118 (2023) 2 1140
1119
1120 [49] Wang, Z., Cai, S., Liu, A., Jin, Y., Hou, J.,
1121 Zhang, B., Lin, H., He, Z., Zheng, Z., Yang,
1122 Y., et al.: Jarvis-1: Open-world multi-task agents
1123 with memory-augmented multimodal language mod-
1124 els. arXiv preprint arXiv:2311.05997 (2023) 2 1141
1125
1126 [50] Wu, Y., Min, S.Y., Prabhumoye, S., Bisk, Y., Salakhut-
1127 dinov, R.R., Azaria, A., Mitchell, T.M., Li, Y.: Spring:
1128 Studying papers and reasoning to play games. Advances
1129 in Neural Information Processing Systems **36** (2024) 2 1142
1130
1131 [51] Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y.,
1132 Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a
1133 unified representation for a variety of vision tasks. In:
Proceedings of the IEEE/CVF Conference on Com-
puter Vision and Pattern Recognition. pp. 4818–4829
(2024) 3 1143
1144
1145 [52] Xu, X., Wang, Y., Xu, C., Ding, Z., Jiang, J., Ding, Z.,
Karlsson, B.F.: A survey on game playing agents and
large models: Methods, applications, and challenges.
arXiv preprint arXiv:2403.10249 (2024) 2 1144
1145
1146 [53] Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang,
G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.:
Mmmu: A massive multi-discipline multimodal un-
derstanding and reasoning benchmark for expert agi.
In: Proceedings of the IEEE/CVF Conference on
Computer Vision and Pattern Recognition. pp. 9556–
9567 (2024) 8 1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

1188	[61] Zhu, Y., Zhu, M., Liu, N., Ou, Z., Mou, X., Tang,	1242
1189	J.: Llava- ϕ : Efficient multi-modal assistant with small	1243
1190	language model. arXiv preprint arXiv:2401.02330	1244
1191	(2024) 8	1245
1192		1246
1193		1247
1194		1248
1195		1249
1196		1250
1197		1251
1198		1252
1199		1253
1200		1254
1201		1255
1202		1256
1203		1257
1204		1258
1205		1259
1206		1260
1207		1261
1208		1262
1209		1263
1210		1264
1211		1265
1212		1266
1213		1267
1214		1268
1215		1269
1216		1270
1217		1271
1218		1272
1219		1273
1220		1274
1221		1275
1222		1276
1223		1277
1224		1278
1225		1279
1226		1280
1227		1281
1228		1282
1229		1283
1230		1284
1231		1285
1232		1286
1233		1287
1234		1288
1235		1289
1236		1290
1237		1291
1238		1292
1239		1293
1240		1294
1241		1295

1296
1297
1298
1299
1300
1301
1302
13031350
1351
1352
1353
1354
1355
1356
1357

Supplementary Material

A1. Additional details

A1.1. Prompts used to generated datasets

Figure A1. Long caption generation with GPT-4V

Please provide a detailed description of the image, ensuring that no details are omitted. Describe every element you observe within the image to provide a comprehensive account of its contents. Don't be lazy and it is important to get everything well done.

1304
1305
1306
1307
1308
1309
1310
1311
13121358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

Figure A2. image-to-JSON data generation

First, provide a detailed description of the image, including every small detail possible. Next, create ten multiple-choice questions based on the content of the image. Each question should test the understanding of the image's content. Follow this JSON format:

```
{ "description": "Full Image Description", "short_description": "Short Image Description", "dialogue": ["Any visible dialogue text as a json list"], "on_screen_subtitle": "any subtitle on the image or n/a", "minimap_details": "Information from the minimap", "inventory_display": "Information about the player's inventory", "score_or_progress": "Details about scores or progress indicators", "NPC_status": "Information about NPCs", "event_indicators": "Indicators of any special events", "interaction_prompts": "Visible prompts for player interactions", "game_mode": "Current game mode or context", "HUD_description": "description of the game HUD or n/a if there is no HUD", "on_screen_watermark": "any watermark on the image or n/a", "summary_of_ui_values": "summary of the UI values as json or empty json if there is no UI", "scene_description": "A high-level overview of the entire scene", "character_list": [ { "name": "Character Name", "appearance": "Description of appearance", "clothing": "Description of clothing", "facial_expression": "Description of facial expression" } ], "object_list": [ "Object 1", "Object 2", ... ], "texture_details": "a json list of object name and texture patterns that they have", "lighting_details": "Specific information about the light sources and shadows in the scene", "color_palette": [ "hexadecimal color code", "hexadecimal color code", ... ], "weather_conditions": "Description of any weather effects present, or say cannot be determined", "environmental_effects": "Description of any environmental effects like fog, rain, fire, etc.", "animation_states": "Descriptions of any static poses or actions implied by character positions", "error_log": "Any noticeable glitches or anomalies in the image", "glitches": "any glitch or buggy aspect of the image or none if there is nothing", "player_status": { "health": "Player's health value", "equipment": "Player's equipment details", "other_status": "Other status indicators" } }
```

1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
13441387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

Figure A3. LLama-3-based data generation

Using the image description provided below, create 10 questions and their corresponding answers that pertain exclusively to the details given in the description. Format your response using JSON.

Image description: <image description here >

Ensure your questions are relevant and directly related to the image description. For example, do not ask about elements not explicitly mentioned in the description.

Figure A4. GPT-4o-based data generation

First, provide a detailed description of the image, including every small detail possible. Next, create 10 questions answers based on the content of the image. Each question should test the understanding of the image's content.

1345
1346
1347
1348
1349

1404
1405
1406
1407
1408
1409
1410

Table A1. Description of entries in the JSON structure

Key	Description
Description	Detailed description of the image
Short description	Concise description of the image
Dialogue	A (JSON) list containing any visible dialogue text
On-Screen subtitle	Subtitles displayed on the image
Inventory display	Details of the player's inventory visible on the image
HUD description	Description of the game's Head-Up Display (HUD)
Scene description	High-level overview of the entire scene
NPC status	High level information about non-playable characters (NPCs)
Character list	List of characters, including their appearances, clothing, and facial expressions
Animation states	Descriptions of static poses or actions suggested by character positions
Object list	A (JSON) list containing all the visible objects in the scene
Texture details	A (JSON) list detailing object names and their texture patterns
Lighting details	Specific information about the light sources and shadows in the scene
Weather conditions	Description of any weather effects present, or state if they cannot be determined
Environmental effects	Description of environmental effects such as fog, rain, or fire
Player status	Player's health, equipment details, and other status indicators

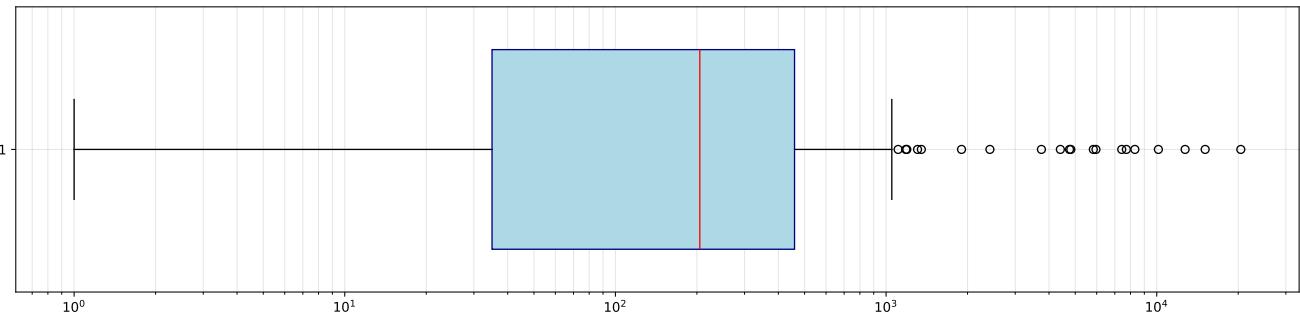
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
14571458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511

Figure A5. Image distribution across games, with a median of 205 unique images per game.

1512

A2. Additional results

1513

In this section, we provide complementary results for the experiments conducted in the main text.

1515

1516

1517

1518

1519

1520

1521

1522

1523

1524

1525

1526

1527

1528

1529

1530

1531

1532

1533

1534

1535

1536

1537

1538

1539

1540

1541

1542

1543

1544

1545

1546

1547

1548

1549

1550

1551

1552

1553

1554

1555

1556

1557

1558

1559

1560

1561

1562

1563

1564

1565

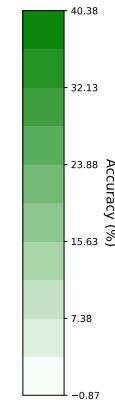


Figure A6. Breakdown of improvement (percentage points) compared to the basemodel Bunny, after fine-tuning it on different mixture strategies by different question categories. As we increase the size of the dataset, all categories show improvement, with *Anomalies and Glitches* showing the greatest gains.

Table A2. Comparison of VIDEOGAMEBUNNY and LLaVA-1.6-34b

Category	VIDEOGAMEBUNNY	LLaVA-1.6-34b
Action Understanding	84.6 ↑(+2.9)	81.7
Anomalies and Glitches	82.1 ↑(+16.6)	65.5
Character Analysis	84.9 ↑(+0.6)	84.3
Common Sense Reasoning	89.1 ↓(-2.8)	91.9
Gameplay Mechanics	80.6 ↓(-5.8)	86.4
HUD and UI	84.1 ↑(+3.0)	81.1
Miscellaneous	86.2 ↑(+1.7)	84.5
Scene Understanding	92.4 ↑(+0.2)	92.2
Small Details	80.1 ↓(-0.2)	80.3
Spatial Reasoning	78.3 ↑(+1.0)	77.3

1620

A3. Qualitative samples

1621

In this section, we provide qualitative results from VIDEOGAMEBUNNY.

1623

1624 Qualitative results where VIDEOGAMEBUNNY provides a detailed description of the image.

1625



1626

1627 **Question:** Briefly describe the image.

1628

1629 **VIDEOGAMEBUNNY:** The image is a screenshot from a video game, showing a character standing in front of a large, ornate
1630 painting. The painting depicts a man with long hair and a beard, wearing a period outfit with a high collar and a sash. The
1631 character is facing the painting, with their back to the viewer, and appears to be observing it. The room has a dimly lit,
1632 old-fashioned ambiance, with a sconce on the wall and a large, framed painting on the wall behind the character.

1633

1634

1635

1636

1637

1638

1639

1640

1641

1642

1643

1644

1645

1646

1647

1648 Figure A7. VIDEOGAMEBUNNY provides a detailed description of the image

1649

1650

1651

1652

1653

1654

1655

1656

1657

1658

1659

1660

1661

1662

1663

1664

1665

1666

1667

1668

1669

1670

1671

1672

1673

1674

1675

1676

1677

1678

1679

1680

1681

1682

1683

1684

1685

1686

1687

1688

1689

1690

1691

1692

1693

1694

1695

1696

1697

1698

1699

1700

1701

1702

1703

1704

1705

1706

1707

1708

1709

1710

1711

1712

1713

1714

1715

1716

1717

1718

1719

1720

1721

1722

1723

1724

1725

1726

1727

1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738

1739 Qualitative results where VIDEOGAMEBUNNY provides a detailed description of the image.
1740



1763 **Question:** Briefly describe the image.

1764 **VIDEOGAMEBUNNY:** The image is a still from a video game, depicting two characters in a dimly lit wooden structure. The
1765 character on the left is a man with a stern expression, wearing a dark robe with a patterned design. He is facing the other
1766 character, who is partially obscured and has their back to the viewer. The second character appears to be wearing a similar robe.
1767 The scene is set in a rustic environment with wooden walls and a thatched roof.

1769
1770 Figure A8. VIDEOGAMEBUNNY provides a detailed description of the image
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781

1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835

1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848

1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943

Qualitative results where VIDEOGAMEBUNNY provides a correct answer with minor inaccuracies



Question: Describe the characters in the image

VIDEOGAMEBUNNY: The image shows two characters on horseback. The character in the foreground is holding a lantern, and the character in the background *is following them*. Both characters are wearing hats.

Figure A9. VIDEOGAMEBUNNY provides a detailed description of the image with minor inaccuracies. Details that are incorrect are highlighted and emphasized in *red*.

1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889

1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954

1955 Qualitative results where VIDEOGAMEBUNNY provides a correct answer with some inaccuracies

1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979



Question: Describe the characters in the image

VIDEOGAMEBUNNY: The image shows Batman, who is in the center of the frame, flying through the air. He is wearing his iconic black suit *with a yellow utility belt*. *Below him, there is a character named Penguin, who is wearing a tuxedo and a bowler hat.*

1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997

Figure A10. VIDEOGAMEBUNNY hallucinates extra characters and a Batman utility belt in the image, which are not actually present. Details that are incorrect are highlighted and emphasized in *red*.

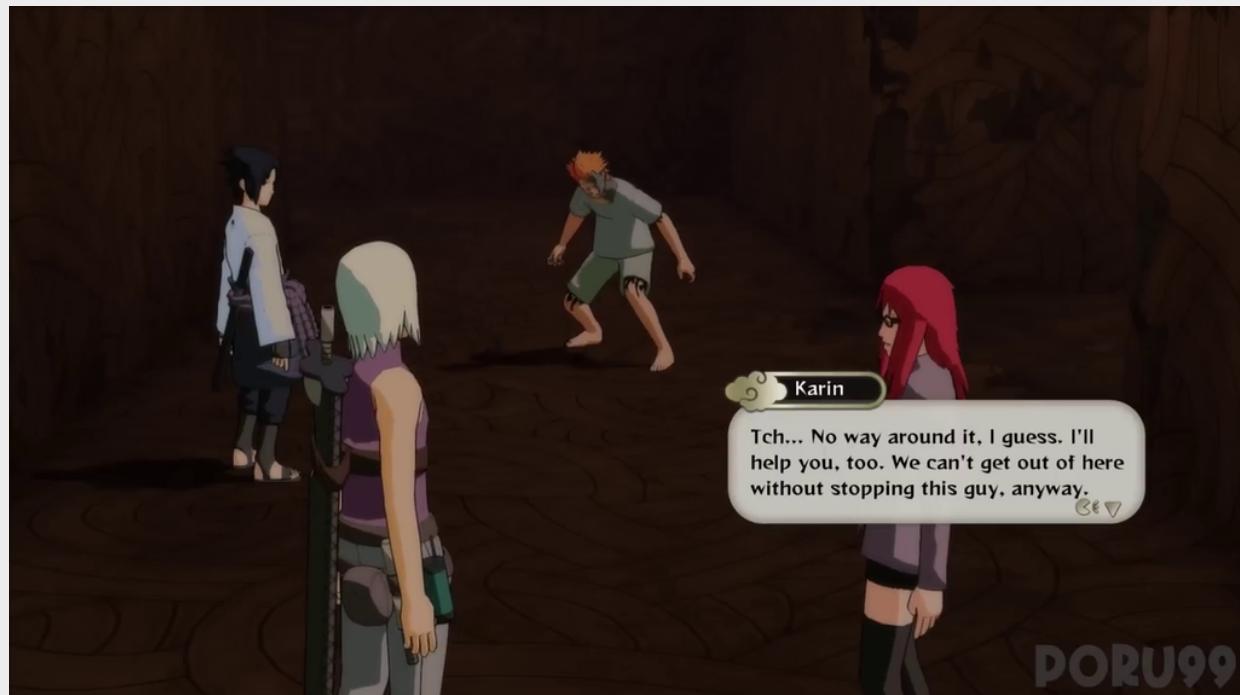
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051

2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063

2064 Qualitative results where VIDEOGAMEBUNNY provide a **wrong** answer for a query
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087

2088 **Question: What is written on the image?**

2089 VIDEOGAMEBUNNY: The text on the image reads: 'Tch... Tch... Tch... You way around it, I guess. Tch... Tch... Tch... I'll tell
2090 you, without stepping this guy away...' 2142
2143
2144
2145
2146



2091
2092
2093 Figure A11. A sample showing that VIDEOGAMEBUNNY struggles to convert the image to text.
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105

2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141

2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171

2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227

2172 Qualitative results where VIDEOGAMEBUNNY provide a **wrong** answer for a query
2173



2196 **Question: What is written on the left?**

2197 **VIDEOGAMEBUNNY:** The word 'RELOAD' is written on the left side of the gun.
2198

2200 Figure A12. A sample showing that VIDEOGAMEBUNNY fails to provide a correct answer. While the word *RELOAD* is written in the
2201 center of the image, the query asks, *What is written on the left?* which should be answered with *nothing*.
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213

2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267

2268	A4. Sample image-to-JSON conversion	2322
2269		2323
2270	Sample image-to-JSON conversion done by VIDEOGAMEBUNNY	2324
2271		2325
2272		2326
2273		2327
2274		2328
2275		2329
2276		2330
2277		2331
2278		2332
2279		2333
2280		2334
2281		2335
2282		2336
2283		2337
2284		2338
2285		2339
2286		2340
2287		2341
2288		2342
2289		2343
2290		2344
2291		2345
2292		2346
2293		2347
2294		2348
2295	{ "description": "The image shows a first-person perspective of a video game, likely a survival horror title. The player is holding a large, futuristic-looking rifle with a circular scope. The rifle has a white and black camouflage pattern. The player's left hand is visible, adorned with a tattoo of a skull and what appears to be a snake. The player is standing in a dark, dilapidated wooden structure, possibly a barn or shed. The walls are made of weathered wooden planks, and the floor is made of wooden planks as well. There is a doorway on the left side of the image, leading to a darker area. A small, white silhouette of a person holding a rifle is visible on the left side of the screen, indicating the player's position on a minimap. The minimap shows the player's location as a white dot within a circular area. The player's health bar is visible at the bottom of the screen, represented by a green bar with a white outline. The health bar is nearly full. The player's ammunition count is displayed in the bottom right corner of the screen, showing '0' bullets remaining in the current magazine and '41' bullets in reserve. A small icon of a skull with a knife is visible next to the ammunition count, possibly indicating a special ability or weapon. The overall atmosphere of the image is dark, gritty, and suspenseful, typical of the survival horror genre.", "dialogue": [], "on_screen_subtitle": "n/a", "minimap_details": "The minimap shows the player's location as a white dot within a circular area. The minimap is located in the bottom left corner of the screen.", "inventory_display": "The image does not show the player's full inventory.", "score_or_progress": "The image does not display any score or progress indicators.", "NPC_status": "The image does not provide any information about NPCs.", "event_indicators": "The image does not display any specific event indicators.", "interaction_prompts": "The image does not show any interaction prompts.", "game_mode": "The image suggests the game is in a single-player campaign mode, based on the presence of a minimap and health bar.", "HUD_description": "The HUD consists of a circular minimap in the bottom left corner, a circular health bar in the bottom left corner, and an ammunition counter in the bottom right corner. The ammunition counter displays the current magazine capacity, the remaining bullets in the magazine, and the total reserve ammunition.", "on_screen_watermark": "MKiceAndFire", "summary_of_ui_values": { "health": "Nearly full", "ammunition": "0/41" }, "scene_description": "The scene takes place inside a dilapidated wooden structure, likely a barn or shed. The structure is dark and appears to be abandoned. The walls and floor are made of weathered wooden planks. There is a doorway on the left side of the image, leading to a darker area.", "character_list": [{ "name": "Player", "appearance": "Cannot be fully determined, but the player's left hand is visible, showing a tattoo of a skull and a snake.", "clothing": "Cannot be fully determined", "facial_expression": "Cannot be determined" }], "object_list": ["Rifle", "Scope", "Wooden planks (walls and floor)", "Doorway"], "texture_details": [{ "object": "Rifle", "texture": "White and black camouflage pattern" }, { "object": "Wooden planks", "texture": "Weathered wood texture" }] }	2349
2296		2350
2297		2351
2298		2352
2299		2353
2300		2354
2301		2355
2302		2356
2303		2357
2304		2358
2305		2359
2306		2360
2307		2361
2308		2362
2309		2363
2310		2364
2311		2365
2312		2366
2313		2367
2314		2368
2315		2369
2316		2370
2317		2371
2318		2372
2319		2373
2320		2374
2321		2375

```

2376
2377     "texture": "Weathered, with visible grain and knots"
2378 ],
2379     "lighting_details": "The scene is dimly lit, with the primary light source appearing to be coming from the doorway on the left. This creates a sense of
2380     ↪ depth and shadows within the structure.",
2381     "color_palette": [
2382         "#181818",
2383         "#303030",
2384         "#505050",
2385         "#FFFFFF"
2386     ],
2387     "weather_conditions": "Cannot be determined",
2388     "environmental_effects": "None",
2389     "animation_states": "The player is holding the rifle in a ready position, suggesting a state of alertness and preparedness for combat.",
2390     "error_log": "None",
2391     "glitches": "None",
2392     "player_status": {
2393         "health": "Nearly full",
2394         "equipment": "The player is equipped with a large, futuristic-looking rifle with a circular scope.",
2395         "other_status": "The player has 0 bullets remaining in the current magazine and 41 bullets in reserve."
2396     }
2397 }
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429

```

2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483

Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483

```

1 {
2     "description": "The image shows a third-person perspective of a video game, likely Grand Theft Auto V, based on the HUD elements and graphics style. The
3     ↪ player is driving a white van on a multi-lane road. The van is positioned slightly to the right of the center of the road, heading towards the
4     ↪ viewer. The road is a typical city street with sidewalks, buildings, and other vehicles. The time of day appears to be midday with clear skies. The
5     ↪ health bar in the bottom right corner. The minimap shows the player's location on a simplified map of the surrounding area. The money counter
6     ↪ displays the amount of in-game currency the player has, which is 9969. The health bar is partially obscured, but it appears to be mostly full.",
7     "dialogue": [],
8     "on_screen_subtitle": "n/a",
9     "minimap_details": "The minimap shows the player's location on a simplified map of the surrounding area. The player is represented by a white arrow
10    ↪ pointing upwards, indicating their direction. The map shows roads, buildings, and a body of water, likely a river or lake.",
11     "inventory_display": "Not visible in the image.",
12     "score_or_progress": "The player's current in-game money is displayed as 9969 in the top right corner.",
13     "NPC_status": "Not applicable, as there are no visible NPCs in the image.",
14     "event_indicators": "None visible in the image.",
15     "interaction_prompts": "None visible in the image.",
16     "game_mode": "Cannot be determined from the image.",
17     "HUD_description": "The HUD consists of a minimap in the bottom left corner, a money counter in the top right corner, and a small portion of the
18     ↪ player's health bar in the bottom right corner.",
19     "on_screen_watermark": "The watermark \"Elgin Ave\" is visible in the bottom right corner.",
20     "summary_of_ui_values": {
21         "money": "9969"
22     },
23     "scene_description": "The scene depicts a typical urban environment with buildings, roads, and vehicles. The player is driving a van on a multi-lane
24     ↪ road, heading towards a large, modern-looking building with a blue light display.",
25     "character_list": [],
26     "object_list": [
27         "Van",
28     ]
29

```

```

2484
2485     "Buildings",
2486     "Road",
2487     "Sidewalk",
2488     "Trees",
2489     "Vehicles",
2490     "Streetlights",
2491     "Blue Light Display"
2492 ],
2493 "texture_details": [
2494   {
2495     "object": "Buildings",
2496     "texture": "Brick, concrete, glass"
2497   },
2498   {
2499     "object": "Road",
2500     "texture": "Asphalt"
2501   },
2502   {
2503     "object": "Sidewalk",
2504     "texture": "Concrete"
2505   }
2506 ],
2507 "lighting_details": "The scene is brightly lit, suggesting midday sun. The lighting casts soft shadows from the buildings and trees onto the road.",
2508 "color_palette": [
2509   "#87CEEB",
2510   "#FFFFFF",
2511   "#A9A9A9",
2512   "#696969",
2513   "#B0B0B0"
2514 ],
2515 "weather_conditions": "Clear and sunny.",
2516 "environmental_effects": "None visible in the image.",
2517 "animation_states": "The van's wheels suggest it is in motion, and the other vehicles on the road imply traffic.",
2518 "error_log": "None visible in the image.",
2519 "glitches": "None visible in the image.",
2520 "player_status": {
2521   "health": "Partially obscured, but appears to be mostly full.",
2522   "equipment": "Not visible in the image.",
2523   "other_status": "The player has 9969 in-game currency."
2524 }
2525 }
```

2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591

2507 Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```

2526
2527
2528
2529
2530
2531
2532
2533   {
2534     "description": "The image captures a scene from the video game Red Dead Redemption 2, showcasing the protagonist, Arthur Morgan, riding his horse, Black
2535     → Jack, along a dirt path. The path leads towards a small, industrial settlement built on a cliffside. The settlement features wooden buildings with
2536     → corrugated metal roofs, characteristic of the late 19th-century American frontier. A plume of black smoke rises from a chimney in the background,
2537     → suggesting industrial activity. The surrounding environment is a mix of rocky terrain and lush vegetation, with a dense forest visible in the
2538     → distance. The sky is overcast, with a hazy, diffused light suggesting either early morning or late afternoon. The overall atmosphere is one of quiet
2539     → anticipation, as if Arthur is about to enter a new chapter in his journey.",
2540   "dialogue": [],
2541   "on_screen_subtitle": "n/a",
2542 }
```

2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591

```
2592
2593     "minimap_details": "n/a",
2594     "inventory_display": "n/a",
2595     "score_or_progress": "n/a",
2596     "NPC_status": "n/a",
2597     "event_indicators": "n/a",
2598     "interaction_prompts": "n/a",
2599     "game_mode": "n/a",
2600     "HUD_description": "n/a",
2601     "on_screen_watermark": "n/a",
2602     "summary_of_ui_values": {},
2603     "scene_description": "Arthur Morgan riding his horse Black Jack towards a small industrial settlement on a cliffside.",
2604     "character_list": [
2605         {
2606             "name": "Arthur Morgan",
2607             "appearance": "A rugged man with a beard and mustache, wearing a dark hat and a dark coat.",
2608             "clothing": "Dark hat, dark coat, dark pants, and boots.",
2609             "facial_expression": "Cannot be determined as his back is to the camera."
2610         }
2611     ],
2612     "object_list": [
2613         "Horse",
2614         "Dirt path",
2615         "Wooden buildings",
2616         "Corrugated metal roofs",
2617         "Smoke plume",
2618         "Chimney",
2619         "Forest",
2620         "Rocks",
2621         "Vegetation",
2622         "Trees"
2623     ],
2624     "texture_details": [
2625         {
2626             "object": "Horse",
2627             "texture": "Short, dark brown fur"
2628         },
2629         {
2630             "object": "Dirt path",
2631             "texture": "Uneven, dusty, with patches of grass"
2632         },
2633         {
2634             "object": "Wooden buildings",
2635             "texture": "Weathered wood planks"
2636         },
2637         {
2638             "object": "Corrugated metal roofs",
2639             "texture": "Rusted and dented metal sheets"
2640         },
2641         {
2642             "object": "Rocks",
2643             "texture": "Rough, jagged, with moss and lichen"
2644         },
2645         {
2646             "object": "Vegetation",
2647             "texture": "A mix of grass, shrubs, and trees"
2648         }
2649     ],
2650     "lighting_details": "The scene is lit by diffused sunlight, suggesting an overcast sky. The light is soft and even, with no strong shadows.",
2651     "color_palette": [
2652         "#A0A0A0",
2653         "#808080",
2654         "#606060",
2655         "#404040",
2656         "#202020",
2657         "#008000",
2658         "#006400",
2659         "#004000"
2660     ],
2661     "weather_conditions": "Overcast",
2662     "environmental_effects": "Smoke plume rising from a chimney",
2663     "animation_states": "Arthur Morgan is sitting upright on his horse, suggesting a calm and controlled ride.",
2664     "error_log": "None",
2665     "glitches": "None",
2666     "player_status": [
2667         {
2668             "health": "n/a",
2669             "equipment": "n/a",
2670             "other_status": "n/a"
2671         }
2672     ]
2673 }
```

2700 Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



```
1 {
2   "description": "The image showcases a close-up of a man in a World War II-era military uniform. He is positioned slightly to the right of the center, with his gaze directed slightly upwards and to the left. His expression is serious, bordering on somber. He has short, dark hair and a prominent nose. The man's uniform is a muted green, with a noticeable dirt smudge on his left cheek. A dark green helmet with a netting pattern covers his head, partially obscuring his hair. The helmet's chin strap is visible, fastened with a metal buckle. A thick, dark green strap crosses his chest, likely part of a backpack or equipment harness. The background is a blurred depiction of a battlefield, with other soldiers in similar uniforms visible in the distance. The overall color palette is muted and desaturated, reflecting the grim atmosphere of war.",
3   "dialogue": [],
4   "on_screen_subtitle": "n/a",
5   "minimap_details": "n/a",
6   "inventory_display": "n/a",
7   "score_or_progress": "n/a",
8   "NPC_status": "n/a",
9   "event_indicators": "n/a",
10  "interaction_prompts": "n/a",
11  "game_mode": "n/a",
12  "HUD_description": "n/a",
13  "on_screen_watermark": "n/a",
14  "summary_of_ui_values": {},
15  "scene_description": "The scene depicts a soldier in the midst of a battlefield, likely during World War II, given the uniforms and equipment.",
16  "character_list": [
17    {
18      "name": "Unknown",
19      "appearance": "The man appears to be in his late twenties to early thirties. He has a rugged appearance with short, dark hair and a prominent nose.",
20      "clothing": "He is wearing a World War II-era military uniform in a muted green color. A dark green helmet with a netting pattern covers his head, and a thick, dark green strap crosses his chest, likely part of a backpack or equipment harness.",
21      "facial_expression": "His expression is serious and slightly somber, suggesting he is focused on the situation at hand."
22    }
23  ],
24  "object_list": [
25    "Military uniform",
26    "Helmet",
27    "Backpack strap",
28    "Backpack (partially visible)",
29    "Soldiers (in the background)"
30  ],
31  "texture_details": [
32    {
33      "object": "Helmet",
34      "texture": "Netting pattern"
35    },
36    {
37      "object": "Uniform",
38      "texture": "Camoouflage pattern"
39    }
40  ],
41  "lighting_details": "The lighting is soft and diffused, likely due to overcast skies or fog. It casts subtle shadows on the man's face, accentuating his features.",
42  "color_palette": [
43    "#5C6B5A",
44    "#3D4A3B",
45    "#7A8A76",
46    "#B2B4A9"
47  ],
48  "weather_conditions": "The weather appears to be overcast or foggy, as evidenced by the soft lighting and lack of harsh shadows."

```

2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807

2808
2809
2810
2811
2812
2813
2814
2815

```

49 "environmental_effects": "The background suggests a battlefield, with smoke or dust particles visible in the air.",
50 "animation_states": "The man appears to be in a still pose, likely captured in a moment of pause or reflection during the chaos of battle.",
51 "error_log": "None",
52 "glitches": "None",
53 "player_status": {
54   "health": "n/a",
55   "equipment": "n/a",
56   "other_status": "n/a"
57 }
58 }
```

2862
2863
2864
2865
2866
2867
2868
28692816
2817

Sample image-to-JSON conversion done by VIDEOGAMEBUNNY



2828

```

1 {
2   "description": "The image shows a male character in a video game, seemingly in the middle of a climbing or jumping action. He is positioned on a stone
3   ↪ platform within a dimly lit, ancient-looking structure. The platform is part of a larger architectural complex, with visible stone walls and a
4   ↪ barred gate in the background. The character is dressed in a sleeveless top and pants, suggesting a warrior or adventurer archetype. He is equipped
5   ↪ with a sword sheathed on his back and appears to be in motion, with his body angled towards the platform's edge. The overall atmosphere is one of
6   ↪ adventure and exploration, with the character's pose hinting at a moment of action or transition within the game.",
7   "dialogue": [],
8   "on_screen_subtitle": "n/a",
9   "minimap_details": "Cannot be determined",
10  "inventory_display": "Not visible",
11  "score_or_progress": "Not visible",
12  "NPC_status": "No NPCs present",
13  "event_indicators": "None visible",
14  "interaction_prompts": "None visible",
15  "game_mode": "Cannot be determined",
16  "HUD_description": "n/a",
17  "on_screen_watermark": "SHIRRAKO in the bottom right corner",
18  "summary_of_ui_values": {},
19  "scene_description": "The scene is set within a large, ancient-looking structure, possibly a temple or palace. The architecture suggests a Middle
20  ↪ Eastern or Mediterranean influence. The environment is dimly lit, with the primary light source appearing to be natural light filtering in from an
21  ↪ unseen opening above.",
22  "character_list": [
23    {
24      "name": "Unknown",
25      "appearance": "Male, athletic build",
26      "clothing": "Sleeveless top, pants, possibly leather or fabric",
27      "facial_expression": "Cannot be determined"
28    }
29  ],
30  "object_list": [
31    "Stone platform",
32    "Stone walls",
33    "Barred gate",
34    "Sword",
35    "Pottery jug"
36  ],
37  "texture_details": [
38    {
39      "object": "Stone walls",
40      "texture": "Rough, uneven stone with visible cracks and weathering"
41    }
42  ]
43 }
```

2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915

```
2916
2917     },
2918     {
2919         "object": "Stone platform",
2920         "texture": "Similar to the walls, but with a slightly smoother appearance"
2921     }
2922 ],
2923 "lighting_details": "The scene is dimly lit, with the primary light source appearing to be natural light coming from an unseen opening above. This
2924     ↪ creates a sense of depth and highlights the textures of the stone surfaces.",
2925     "color_palette": [
2926         "#5C4A3A",
2927         "#3B322A",
2928         "#7A6D5B",
2929         "#A1887F"
2930     ],
2931 "weather_conditions": "Cannot be determined",
2932 "environmental_effects": "None visible",
2933 "animation_states": "The character's pose suggests a dynamic action, either in the middle of a jump or a climbing maneuver.",
2934     "error_log": "None visible",
2935     "glitches": "None",
2936     "player_status": {
2937         "health": "Not visible",
2938         "equipment": "Sword visible on back",
2939         "other_status": "Not visible"
2940     }
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
```