Internship Project Report

Master of Science in Big Data and Internet of Things

MBDIoT

3D CAPTIONING WITH PRETRAINED MODELS

Submitted by:

Mr. FAZZA Abdellah

ENSAM Casablanca
2023-2024

Abstract

This report presents a detailed overview of a project focused on generating descriptive captions for 3D objects as part of an internship at 3D Smart Factory. The project leverages pre-trained models to create captions from multi-view renders of 3D objects, reducing the need for costly and time-consuming manual annotation. Inspired by the methodology in *Cap3D* from the research paper *Scalable 3D Captioning with Pretrained Models*[1], the process includes rendering 3D objects from eight specific viewpoints using *Blender*, applying image captioning using the *Paligemma*[2] model, and consolidating the generated captions through the *Gemma 2 LLM* [3, p. 2] (Large Language Model). The methodology focuses on computational efficiency by utilizing less resource-intensive models while maintaining the quality of the generated descriptions. Results demonstrate the effectiveness of multi-view rendering for 3D captioning, though challenges such as viewpoint coverage and model limitations remain. The report concludes with a discussion of the project's outcomes and potential future improvements.

Figure Table

Table of Content

I. Introduction

In recent years, the demand for generating high-quality descriptions of 3D objects has risen significantly, driven by advancements in fields such as virtual reality, 3D modeling, and autonomous systems. Automatic 3D object captioning, which aims to generate textual descriptions of 3D assets, offers a promising solution to reduce manual annotation, enabling scalable applications in various industries.

The purpose of this report is to present a comprehensive overview of a 3D object captioning project undertaken as part of an internship at 3D Smart Factory. The project aims to generate descriptive captions for 3D objects using pre-trained models and a multi-view image captioning approach. The work builds on the methodology outlined in *Cap3D*, utilizing a pipeline that includes rendering 3D objects from different viewpoints, applying image captioning models, and fusing captions from various perspectives to create a coherent and accurate description.

This report will first present the general context of the project and the hosting organization, followed by a review of related literature on 3D object recognition and captioning techniques. The methodology, implementation, results, and limitations of the project will then be detailed. Finally, the report concludes with an analysis of the findings and future directions for improving the system.

II. General Context

1. Introduction

This chapter provides the general context of the project, focusing on its objectives and the specific challenges it addresses. It outlines the background and purpose of the project, emphasizing the need for automated 3D object captioning. The chapter also introduces the structure of the project and its goals as well as the host organization.

2. Host Organization

a. 3D Smart Factory

3D SMART FACTORY is an innovative Moroccan company, established in Mohammedia since 2018 and led by Mr. Khalil Lbbarki as its director. Its core business focuses on the research and development of 3D technologies.

The primary objective of 3D SMART FACTORY is to support young entrepreneurs in realizing their projects, providing them with the necessary tools to turn their ideas into reality.

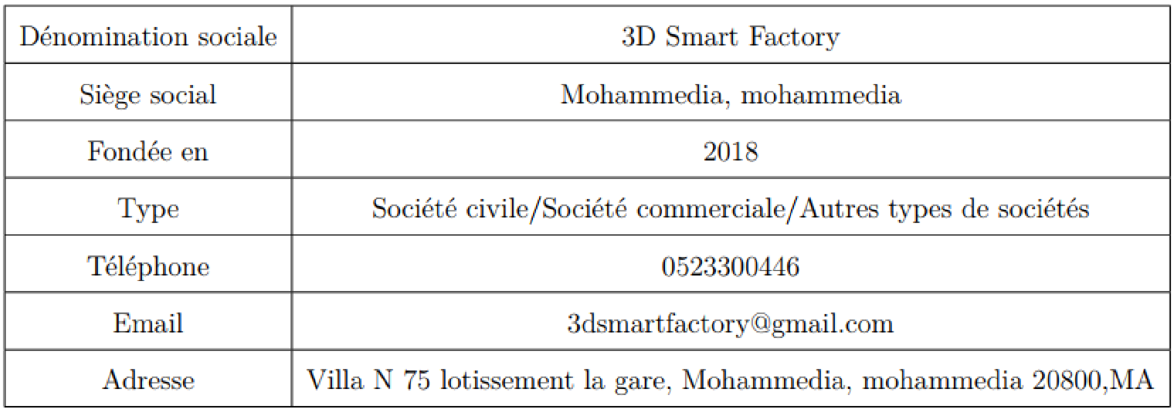
As a research and development-oriented company, 3D SMART FACTORY is committed to supporting entrepreneurs at every stage of their journey, from the research phase to production, with a vision of having a positive impact on economic and social development.

Beyond its research and development activities, the company also specializes in designing multi-technology devices for the medical sector, demonstrating its ability to innovate and diversify its areas of expertise.

Thanks to its commitment to innovation and supporting young entrepreneurs, 3D SMART FACTORY has contributed significantly to the growth and technological progress in Morocco.

Figure 1: 3D Smart Factory Logo

b. Technical Sheet



Dénomination sociale	3D Smart Factory
Siège social	Mohammedia, mohammedia
Fondée en	2018
Type	Société civile/Société commerciale/Autres types de sociétés
Téléphone	0523300446
Email	3dsmartfactory@gmail.com
Adresse	Villa N 75 lotissement la gare, Mohammedia, mohammedia 20800,MA

Figure 2: 3D Smart Factory Technical Sheet

3. Project Presentation

a. Project Context

The project, conducted during an internship at 3D Smart Factory, focuses on developing an automated pipeline for generating captions for 3D objects. By leveraging pre-trained image captioning models and multi-view rendering techniques, the project aims to solve the challenges associated with manual annotation and create a scalable captioning system for 3D datasets. The process is inspired by the Cap3D framework and seeks to enhance computational efficiency while maintaining accuracy in the descriptions generated.

b. Objectives

The main objectives of the project include:

- Automated Caption Generation: To build a system capable of generating captions for 3D objects based on multi-view rendered images.

- Computational Optimization: To use less resource-demanding models that minimize the need for fine-tuning, making the system more efficient.

- Coherent Caption Fusion: To consolidate captions from various viewpoints into a single accurate description of each 3D object.

- Scalability: To ensure the system can handle large datasets of 3D objects with minimal manual intervention.

4. Conclusion

This chapter introduced the background and purpose of the 3D object captioning project and outlined the main objectives. The next chapter will cover a review of relevant literature in 3D object recognition, image captioning models, and challenges related to 3D-text data.

III. Literature Review

1. Introduction

This chapter reviews the literature on 3D object recognition and captioning, exploring the advancements in multimodal learning and the use of pre-trained models for generating descriptive captions for 3D objects. With the growing demand for automated systems capable of understanding and describing 3D environments, researchers have developed a variety of techniques that combine 3D data, images, and language models. This review will cover major approaches to 3D captioning, focusing on methods that leverage large datasets and multi-modal models to bridge the gap between 2D and 3D understanding.

2. 3D Object Recognition and Captioning

Recent advancements in 3D dense captioning have expanded the ability to localize and describe objects in 3D scenes, offering more detailed and accurate representations compared to 2D visual captioning. A comprehensive survey titled *A Comprehensive Survey of 3D Dense Captioning*[4] highlights the potential of 3D dense captioning in capturing the complexity of real-world scenes. The survey provides an in-depth review of methods, datasets, and evaluation metrics, proposing a standard pipeline for existing approaches. It also identifies key challenges in data collection and processing from 3D point clouds, which serve as the primary data source for 3D scene descriptions.

One emerging framework in this field is GPT4Point, introduced in the paper *GPT4Point: A Unified Framework for Point-Language Understanding and Generation*[5]. This model integrates 3D object understanding and generation within a multimodal large language model (MLLM) framework. GPT4Point is capable of handling tasks such as point-cloud captioning and 3D generation, demonstrating the potential of combining language models with 3D data for improved performance. The model is supported by Pyramid-XL, a large-scale point-language dataset that enhances its 3D-text comprehension.

Another prominent contribution to the field is the Cap3D methodology, as discussed in the paper *Scalable 3D Captioning with Pretrained Models*. Cap3D utilizes pre-trained models for image captioning and image-text alignment to generate captions for 3D objects rendered from multiple views. This approach eliminates the need for manual annotations by automating the caption generation process and has been applied to the *Objaverse*[6] dataset, resulting in the creation of over 660k 3D-text pairs. Through its scalability and efficiency, Cap3D surpasses traditional human-authored descriptions in both quality and speed, proving its effectiveness in large-scale 3D datasets.

Lastly, the introduction of ULIP (Unified Language, Image, and Point Cloud) in the paper *ULIP: Learning a Unified Representation of Language, Images, and Point Clouds for 3D Understanding*[7] explores how multimodal learning can be applied to improve 3D understanding. By aligning 3D object representation with image-text spaces through pre-trained vision-language models, ULIP addresses the limitations posed by small datasets and predefined categories in traditional 3D models. This model demonstrates a significant performance boost in both 3D classification and zero-shot learning, showing the promise of using unified representations across different modalities.

3. Models in Image Captioning and Text Alignment

Advancements in image captioning and text alignment have transformed how machines understand and describe visual data. Over recent years, several models have been developed to enhance the efficiency and accuracy of this task, incorporating multimodal techniques that bridge the gap between vision and language.

One significant model in this field is BLIP (Bootstrapping Language-Image Pre-training), introduced in the paper *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*[8]. BLIP is designed for vision-language pre-training (VLP), excelling in both vision-language understanding and generation tasks. Unlike traditional models that often perform well in one area, BLIP transfers flexibly to both types of tasks. It addresses the issue of noisy web data by using a captioner to generate synthetic captions and a filter to remove inaccurate ones. This technique has enabled BLIP to achieve state-of-the-art results in image captioning (+2.8% *CIDEr*[9]) and image-text retrieval, offering strong generalization capabilities when transferred to video-language tasks.

Another approach is ClipCap, which leverages the CLIP (Contrastive Language-Image Pre-training) model's rich semantic features for vision-language perception. The paper *ClipCap: CLIP Prefix for Image Captioning*[10] introduces this method, which uses a CLIP encoding as a prefix to the captioning process, combined with a pre-trained language model such as *GPT-2*[11]. ClipCap's key advantage is its lightweight nature, requiring only a small mapping network for training while keeping both the CLIP and language models frozen. This enables efficient caption generation with a simpler architecture, providing competitive results on datasets like *Conceptual Captions*[12] and *nocaps*[13], all while remaining faster and lighter compared to other models.

A more recent contribution is GIT (Generative Image-to-Text Transformer), outlined in the paper *GIT: A Generative Image-to-Text Transformer for Vision and Language*[14]. GIT simplifies the architecture for image-to-text generation by using one image encoder and one text decoder under a single task of language modeling. Unlike previous work that relies on external modules like object detectors or OCR (Optical Character Recognition), GIT focuses on a unified model that scales effectively with pre-training data. GIT has set new benchmarks across 12 vision-language tasks, surpassing human performance on *TextCaps*[15], and achieving new state-of-the-art results in image captioning.

In response to the growing cost of training and fine-tuning large models, SmallCap, introduced in the paper *SmallCap: Lightweight Image Captioning Prompted with Retrieval Augmentation*[16], proposes a lightweight alternative. SmallCap combines a CLIP encoder with a GPT-2 decoder, using cross-attention layers for training. It retrieves relevant captions from a datastore, allowing for prompt-based caption generation without the need for large-scale pre-training. SmallCap demonstrates competitive performance on benchmarks like *COCO*[17] and generalizes to other domains without additional fine-tuning by simply adjusting the contents of the datastore.

In the area of text alignment, the CLIP model, introduced in *Learning Transferable Visual Models from Natural Language Supervision*[18], has revolutionized the ability to learn image representations directly from text descriptions. CLIP is pre-trained on a large dataset of 400 million image-text pairs, enabling it to perform zero-shot transfer across a wide range of computer vision tasks, including image classification, action recognition, and fine-grained object classification. The power of CLIP lies in its ability to align visual and textual concepts effectively, allowing it to describe unseen objects and transfer knowledge across domains without the need for additional labeled data. This makes it a versatile model for image captioning tasks where text alignment is critical.

4. Challenges in 3D-Text Data

Despite the significant advancements in 3D object recognition and captioning, numerous challenges persist, particularly in the creation and utilization of 3D-text paired data. One of the primary challenges is the scarcity of large-scale annotated datasets for 3D-text tasks. Unlike 2D image datasets, which have been extensively labeled for various vision tasks, 3D datasets are often smaller and limited to specific domains, making it difficult to generalize models across diverse environments.

Another challenge is the complexity of 3D data representation. 3D objects are typically represented through point clouds, meshes, or voxel grids, each of which introduces unique difficulties in terms of data processing and alignment with textual descriptions. For instance, point clouds—the primary data type used for 3D scene representation—are sparse and unordered, requiring models to accurately capture spatial relationships to generate meaningful captions. Additionally, translating the geometric information in 3D models into natural language descriptions poses a significant hurdle, as it involves capturing both the visual and contextual aspects of the objects in detail.

Moreover, the process of aligning textual data with 3D representations is more challenging than with 2D images. This is due to the inherent complexity of 3D objects, which often require multiple views or in-depth exploration to be fully understood. For example, Cap3D utilizes multi-view rendering to generate captions, which is necessary to cover all aspects of a 3D object. However, this multi-view approach introduces additional computational costs and increases the difficulty of generating coherent and concise textual descriptions from various perspectives.

Another issue is the difficulty of fine-tuning models for 3D-text tasks. While pre-trained models like CLIP and GPT have been widely used for 2D image-text tasks, transferring their success to 3D environments remains an open challenge. Fine-tuning these models on 3D-text paired data is often resource-intensive, requiring large-scale datasets such as *Objaverse* and specialized architectures to handle the complex nature of 3D representations.

Lastly, the evaluation of 3D-text captioning systems is still underdeveloped compared to 2D image captioning. Metrics like *CIDEr*, commonly used for image captioning, are not always adequate for evaluating 3D captions, as they fail to capture the nuanced geometric and spatial information necessary for accurately describing 3D objects. Developing more specialized evaluation metrics that consider both the quality of the description and the accuracy of the geometric interpretation remains a pressing challenge in the field.

5. Conclusion

The literature on 3D object recognition and captioning has evolved significantly with the development of pre-trained models, multi-modal techniques, and large-scale datasets. However, there remain numerous challenges, particularly in the alignment of 3D data with textual descriptions and the scarcity of annotated 3D-text paired data. Models such as BLIP, CLIP, and Cap3D have demonstrated promising advancements, especially in terms of scalability and caption quality. However, addressing the existing challenges, such as computational limitations and evaluation techniques, will be crucial for future developments in this domain. This chapter has provided a broad overview of the advancements in 3D object recognition and captioning and has outlined the significant challenges that must be overcome to continue progressing in this field.

IV. Methodology

1. Introduction

This chapter outlines the methodology employed in the project to generate descriptive captions for 3D objects. The approach is built upon a multi-view rendering technique and leverages pre-trained models for caption generation and fusion. The methodology focuses on efficient data preparation, image captioning, and post-processing to ensure that the generated captions are coherent, accurate, and computationally efficient. Each step in the process is designed to address the specific challenges of 3D-text captioning, such as viewpoint coverage and the complexity of 3D object representation.

2. Data Preparation

a) 3D Models

The 3D objects used in this project are provided in the '.glb' format, a widely accepted format for 3D assets. The models used in this project are sourced from the Objaverse dataset and are downloaded using a custom Python script. Objaverse provides a wide variety of 3D objects with detailed geometric and texture information, making it an ideal dataset for the project. While Objaverse is the primary source, any 3D model in an appropriate format (such as '.glb') from other sources can also be used, provided they meet the necessary standards for rendering and captioning.

b) Multi-View Rendering

To capture the full structure of each 3D object, the project employs multi-view rendering, generating 2D images from eight specific viewpoints: front, back, above, below, left, right, and two diagonal perspectives. This multi-view approach is inspired by methods like Cap3D, where multiple viewpoints are necessary to accurately describe complex 3D shapes. Each view provides a unique perspective of the object, ensuring that no critical visual details are missed in the final caption.

The rendering process is conducted using Blender, a robust open-source 3D creation suite that allows for precise control over camera angles and object lighting. Blender's scripting capabilities are utilized to automate the rendering process, ensuring consistency across the dataset.

c) Image Preprocessing

Once the 2D images are generated, they undergo preprocessing to standardize their size and quality. Each image is resized to a fixed resolution and normalized for consistent input into the captioning models. The preprocessing step also involves converting the images into the appropriate format required by the captioning models used in subsequent steps. This ensures that the images are fed into the model with minimal noise and optimal clarity, facilitating better caption generation.

d) Data Organization

The preprocessed images are stored in a structured format, where each 3D object's rendered views are saved in separate folders, categorized by their viewpoint. This organization helps during the caption generation and fusion stages by associating each image with its corresponding viewpoint, making it easier to reference and combine captions later.

3. Image Captioning

Once the 3D objects are rendered into 2D images from multiple viewpoints, the next step is to generate descriptive captions for each image using pre-trained models. The captioning process leverages existing image captioning models that have been fine-tuned for vision-language tasks, ensuring that the generated descriptions are both accurate and contextually relevant.

a) Model Selection

For image captioning, the project employs *Paligemma 3B-224*, a pre-trained model capable of handling various tasks, including image captioning and visual question answering (VQA). To generate captions, both an image and a corresponding text prompt are passed to the Paligemma model. The text prompt provides context for the model, allowing it to focus on generating a relevant caption. Paligemma has been chosen for its efficiency and ability to generate high-quality captions without the need for fine-tuning, making it suitable for the computational constraints of this project.

The decision to use less resource-intensive models was made to ensure the system can scale efficiently without requiring large computational resources. Paligemma’s pre-trained architecture allows it to generate captions from the rendered views with minimal overhead.

b) Caption Generation Process

Each rendered image from the eight viewpoints is passed into the captioning model along with a corresponding text prompt. The model analyzes the visual input and uses the prompt to generate a caption that describes the visible features of the 3D object. This process is repeated for every image, ensuring that each viewpoint has a corresponding caption that captures the unique aspects of the object’s geometry and appearance from that angle.

The output is a set of eight captions, each representing the object from a different perspective. These captions serve as the foundation for the next stage of the pipeline, where they are combined into a unified description.

c) Handling Multiple Viewpoints

Since the model generates a caption for each individual viewpoint, it is essential to ensure that the captions provide a comprehensive and non-redundant description of the 3D object. Each caption is checked for overlap and redundancy with the others, ensuring that the final set of captions covers the full spectrum of visual details without unnecessary repetition.

By generating a unique caption for each viewpoint, the model can highlight specific features that might not be visible from other angles. This multi-view captioning approach is crucial for accurately describing complex 3D objects, where a single viewpoint may not provide enough information to generate a complete description.

4. Caption Fusion

After generating individual captions for each of the eight viewpoints, the next step is to combine these captions into a single, coherent description of the 3D object. This process, known as caption fusion, ensures that the final output captures the object comprehensively while avoiding redundancy.

a) Approach

For the caption fusion process, the project utilizes the *Gemma2_instruct_2b_en* model, which was downloaded separately and is specifically designed for consolidating multi-view captions. The fusion process involves analyzing the individual captions generated for each viewpoint and merging them into a unified text that incorporates the most relevant and non-redundant details from all views. This ensures that the final description retains key features from each perspective without repetitive information.

b) Caption Fusion

During the fusion process, the model evaluates each caption to determine which details are essential to the object's overall description. The generated captions are evaluated and fused based on their relevance and quality. The quality of the captions is sufficient to avoid the need for generating multiple captions for each image and selecting the best one through a caption ranking phase. This decision streamlines the process and allows for an efficient workflow.

c) Final Output

The result is a single, consolidated caption that provides a comprehensive description of the object based on the combined information from all eight views. This caption serves as the final textual output for the 3D object, ensuring that no critical details are omitted and that the object is described in a way that is both accurate and coherent.

5. Post-Processing

After caption fusion, a post-processing step is employed to clean and refine the generated captions. This step is necessary to remove any unwanted elements or formatting issues that may have been introduced during the captioning and fusion phases.

a) Caption Cleaning

To ensure that the final caption is clear and concise, all unnecessary information, such as the original prompts and any tags or irrelevant formatting, is removed. This step ensures that the caption is not only accurate but also readable and properly structured for interpretation. The cleaning process is automated, following predefined rules to eliminate extraneous data while preserving the integrity of the core description.

b) String Formatting

The cleaned captions are then stored in string format for easy integration into downstream applications. The decision to use string format instead of more complex data structures (e.g., .pkl files) simplifies the handling and retrieval of captions, making them easier to manage, especially when dealing with large datasets.

6. Conclusion

This chapter has outlined the methodology employed for generating captions for 3D objects, from the initial data preparation and multi-view rendering to image captioning, caption fusion, and post-processing. By leveraging pre-trained models and implementing efficient multi-view captioning techniques, the project aims to generate accurate and comprehensive descriptions of 3D objects with minimal computational overhead. The final caption, which is the result of a carefully designed pipeline, provides a robust and scalable solution for 3D object captioning, enabling future enhancements and scalability in both data and model improvements.

V. Implementation

1. Introduction

This chapter details the practical aspects of the project’s implementation, focusing on the tools and frameworks used to build the 3D object captioning pipeline. The implementation phase involves integrating the multi-view rendering, caption generation, and caption fusion components into a streamlined workflow that efficiently handles large-scale 3D models. This chapter will also discuss the selection of tools and how they interact within the project to meet the specific objectives of scalability and efficiency.

2. Tools and Frameworks

a) Python

The entire pipeline is implemented in Python, given its versatility and the wide range of libraries available for machine learning, rendering, and data processing. Python's flexibility allows for seamless integration of various components such as rendering scripts, image preprocessing, and caption generation models.

Figure 3: Python Logo

b) Blender

Blender, an open-source 3D creation suite, is used for rendering 3D objects from multiple viewpoints. It allows for precise control over camera angles, lighting, and object orientation, making it ideal for producing high-quality 2D images from 3D models. Blender's Python scripting capabilities are leveraged to automate the rendering process, ensuring consistency and scalability when dealing with large datasets like Objaverse.

Figure 4: Blender Logo

c) JAX

The project uses JAX, a powerful library for array-oriented numerical computation like NumPy but with enhanced capabilities for automatic differentiation and Just-In-Time (JIT) compilation. JAX enables high-performance computation on CPU, GPU, or TPU, making it an ideal choice for the large-scale image captioning and fusion tasks involved in this project.

JAX's JIT compilation, enabled through the Open XLA ecosystem, allows for significant performance boosts by optimizing computations at runtime. Additionally, JAX functions support automatic differentiation, which is essential for efficiently evaluating gradients during model training. Another key feature of JAX is its ability to automatically vectorize functions, making it highly efficient when working with arrays representing batches of input data. These capabilities make JAX a robust and scalable tool for implementing the Paligemma and Gemma 2 models, ensuring that the computational processes can scale smoothly.

d) Objaverse Dataset

The Objaverse dataset serves as the primary source of 3D models for the project. Objaverse 1.0 is a large-scale dataset containing over 800K 3D models, each accompanied by descriptive captions, tags, and animations. It addresses the gap in the availability of high-fidelity 3D model datasets by improving the scale, diversity of categories, and visual diversity within categories compared to existing 3D repositories. The richness of Objaverse makes it an ideal dataset for this project's caption generation task, as it provides a vast collection of diverse objects, enabling effective multi-view rendering and captioning.

e) Jupyter Notebook

The entire implementation is developed and run within Jupyter Notebooks, an interactive environment that facilitates the execution of code, visualizations, and documentation in one place. Jupyter Notebook is essential for developing, testing, and debugging the pipeline, as it allows for the easy integration of Python libraries, visualization of rendered images, and real-time updates on model outputs.

Figure 5: Jupyter Logo

3. Computational Considerations

The computational requirements for this project were carefully managed to ensure that the pipeline could handle large-scale 3D objects efficiently without extensive resource demands. Several strategies were employed to optimize performance and manage the computational load.

a) Use of Pre-Trained Models

The project relies on pre-trained models such as Paligemma and Gemma 2, which eliminates the need for extensive training from scratch. This decision reduces both the computational resources and time required for model deployment. The pre-trained models, combined with JAX's efficient computation and JIT compilation, ensure that the models can perform real-time caption generation and fusion without overloading the system.

b) Multi-View Rendering Efficiency

To render 3D objects from multiple viewpoints, the use of Blender and its Python scripting capabilities allows for automation and consistency. By optimizing the rendering process to output eight views per object, the system achieves a balance between visual coverage and computational efficiency. The rendered images are then processed in batches, taking advantage of JAX’s automatic vectorization to handle arrays representing multiple images at once.

c) Hardware Utilization

The project was designed to run on GPU hardware to leverage the parallel processing capabilities required for tasks such as image captioning and fusion. JAX's ability to seamlessly switch between CPU and GPU execution allowed for flexible resource allocation. In a distributed setting, JAX can also run computations on TPUs, offering further scalability when dealing with larger datasets or more complex models.

d) Scalability

The system was built with scalability in mind. The ability to upgrade to more efficient pre-trained models, such as newer versions of Gemma or more advanced captioning models, allows the project to maintain computational efficiency while improving performance. Moreover, the modular nature of the pipeline ensures that each component can be independently upgraded or replaced, providing long-term flexibility and adaptability.

4. Conclusion

The implementation of this project successfully integrates multiple tools and frameworks, from Blender for rendering to JAX for machine learning. By leveraging pre-trained models and optimizing computational processes through JIT compilation and batch processing, the pipeline is capable of handling large-scale 3D object captioning tasks with minimal resource demands. The careful balance between computational efficiency and scalability ensures that the system is not only robust but also flexible enough to accommodate future advancements in models and hardware.

VI. Results and Limitations

1. Introduction

This chapter presents the outcomes of the 3D object captioning project, evaluating the performance of the implemented pipeline, and highlighting the key results. The results are analyzed in terms of caption quality and computational efficiency. In addition, this chapter discusses the limitations encountered during the project, such as computational constraints, challenges in multi-view captioning, and potential areas for improvement. The goal is to provide a clear understanding of both the successes and the limitations of the system, as well as insights into how future work can address these challenges.

2. Example Captions

The following are example captions generated by the implemented pipeline using different prompts passed to the Paligemma model. The captions reflect various levels of detail and context based on the nature of the prompt provided.

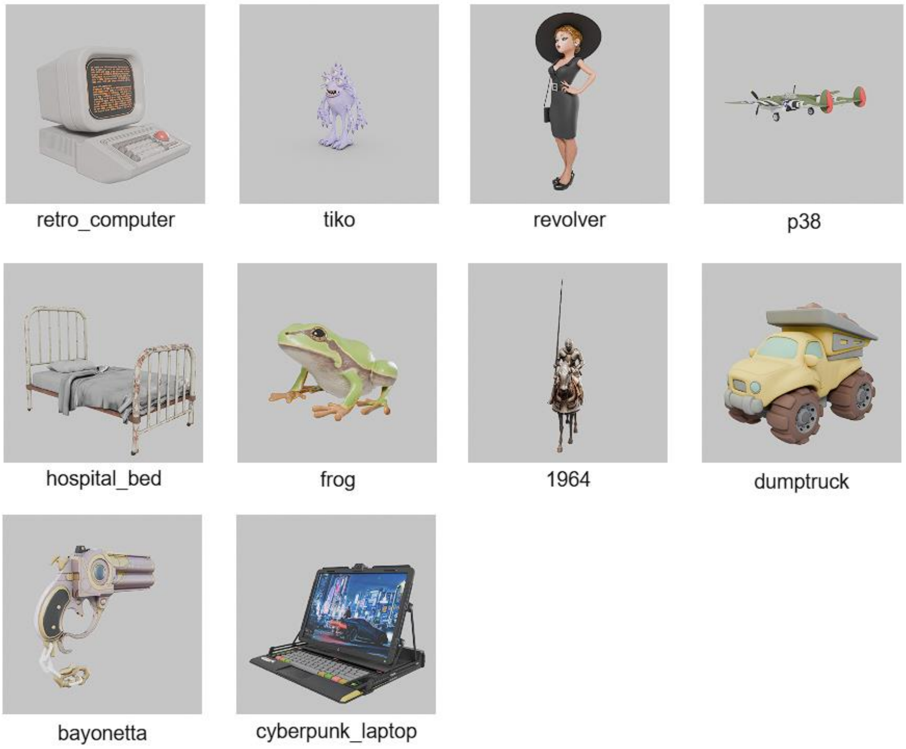


Figure 6: Some of the Tested 3D Models

a. Prompt: caption en

N.B. ‘en’ stands for English. The language the captions generated by Paligemma.

- **retro_computer:** "A computer with a green light on top, a yellow light on the bottom, a keyboard, a monitor with the words 'I'm the boss' on it, and a red mouse on a gray mouse pad."
- **p38:** "A small airplane with a propeller is flying in the sky. The plane has a white tail, black and white stripes on the wing, and a black and white star on the wing."
- **bayonetta:** "A 3D model of a gun with two barrels, a trigger, and a white background with gold accents."
- **hospital_bed:** "A bed with a white sheet and a pillow. The bed frame is white, and the bed is made of metal. The sheet is wrinkled and the pillow is white."
- **dumptruck:** "This is a 3D model of a toy dump truck with a yellow cab, brown wheels, and a gray bed. It has rocks in the back and features a large wheel on the back and a small wheel on the front, along with a taillight and side mirror."
- **cyberpunk_laptop:** "A laptop computer with a screen displaying a city scene. The laptop is black with a silver frame and has a silver hinge on the side."
- **1964:** "A statue of a person in armor, sitting on a horse and holding a sword, with a helmet."
- **revolver:** "A 3D model of a woman in a black dress and hat, holding a gun and a purse, with her head tilted back and eyes closed."
- **frog:** "A green toy frog sitting on a gray background."
- **tiko:** "This is a 3D model of a cartoon toy."

b. Prompt: describe en

- **retro_computer:** "A computer with a screen."
- **p38:** "This is a toy airplane."

- **bayonetta:** "A 3D model of a gun."
- **hospital_bed:** "A bed with a white blanket and a metal frame."
- **dumptruck:** "A toy truck."
- **cyberpunk_laptop:** "A black laptop with a triangle logo, open and plugged in."
- **1964:** "A statue of a person wearing armor and holding a sword, sitting on a horse."
- **revolver:** "3D model of a woman standing and holding a bag."
- **frog:** "A toy frog."
- **tiko:** "This is a 3D model of a toy monster."

c. Prompt: caption this 3d object

- **retro_computer:** "A , ."
- **p38:** "An air."
- **bayonetta:** "Unanswerable."
- **hospital_bed:** "An old, white metal with a chipped white frame, a gray blanket, and a white pillow is in a corner of a white-walled room."
- **dumptruck:** "A ."
- **cyberpunk_laptop:** "A ."
- **1964:** "A statue of a sitting on a horse, wearing armor and holding a sword."
- **revolver:** "A 3D model of a ."
- **frog:** "A ."
- **tiko:** "A toy."

d. Prompt: caption en this 3d object

- **retro_computer:** "A , ."
- **p38:** "An air."
- **bayonetta:** "A firearm."
- **hospital_bed:** "A with a white frame and a gray blanket. The has a white pillow and a gray sheet."
- **dumptruck:** "A ."
- **cyberpunk_laptop:** "A with a triangle logo."
- **1964:** "A statue of a person wearing armor and sitting on a horse, holding a sword."
- **revolver:** "A 3D model of a ."
- **frog:** "A ."
- **tiko:** "A 3D , , toy."

These examples highlight the variation in caption quality depending on the prompt. While the detailed captions ("caption en") offer comprehensive descriptions, the minimalistic captions ("caption this 3d object") tend to be incomplete or vague. This illustrates the importance of providing appropriate prompts to guide the captioning model toward generating useful and detailed descriptions.

3. Analysis

The generated captions across different prompts provide valuable insights into the performance and limitations of the 3D object captioning pipeline. This section analyzes the quality of the captions, the impact of different prompts, and the overall effectiveness of the system in generating coherent and accurate descriptions of 3D objects.

a. Quality of Captions

The quality of the captions varies significantly based on the prompt provided to the Paligemma model. For example, with the prompt "caption en," the captions are highly detailed and accurate, describing multiple attributes of each object such as color, structure, and specific features. These captions often captured key aspects of the object, such as the color and text on the retro_computer or the detailed description of the dumptruck and p38 airplane. This level of detail makes the "caption en" prompt ideal for generating comprehensive descriptions, especially for more complex objects.

However, with simpler prompts like "describe en" the captions become much shorter and less descriptive. While these captions still convey the essence of the object (e.g., "A computer with a screen" for the retro_computer), they lack the richness of detail present in the more complex prompts. This simplification may be useful for scenarios where brevity is preferred, but it may not be sufficient for applications requiring detailed object descriptions.

The "caption this 3d object" prompt led to incomplete captions, often producing fragments like "A , ." for the retro_computer or "An air." for the p38 airplane. This shows that the model struggles to generate meaningful captions when the prompt is too vague or lacks context. Similarly, the prompt "caption en this 3d object" resulted in captions with missing or repeated words, further emphasizing the need for well-structured prompts to guide the model effectively.

b. Impact of Prompt Engineering

The variations in the generated captions highlight the importance of prompt engineering in obtaining high-quality descriptions. As seen in the results, more detailed prompts like "caption en" resulted in rich, descriptive captions, while vague or minimalistic prompts led to incomplete outputs. This suggests that the performance of the Paligemma model is highly dependent on the clarity and specificity of the input prompt.

This dependency on prompts also indicates that further optimization of prompt engineering could enhance the model's performance, especially for more complex 3D objects. By refining the text prompts passed to the captioning model, it may be possible to extract more detailed and coherent descriptions, even in cases where the object is intricate or has multiple features.

c. Multi-View Captioning Effectiveness

The multi-view rendering approach, which captures eight different perspectives of each object, plays a key role in providing diverse visual information for the captioning model. In cases where the object has complex geometry or multiple distinct features, the multi-view approach ensures that the generated captions cover different aspects of the object. For instance, the dumptruck and cyberpunk_laptop benefit from multiple views that allow the model to describe different parts, such as the laptop's screen and the truck's wheels.

However, in some cases, the multi-view approach did not entirely resolve ambiguities. For example, the caption for p38 airplane ("An air.") using the "caption this 3d object" prompt failed to capture essential details despite having multiple views available. This suggests that while the multi-view approach is beneficial, the model’s ability to synthesize information from multiple perspectives into a coherent description can still be improved, particularly when the prompt is unclear or lacks context.

d. Model Limitations

While the Paligemma model performs well with detailed prompts, certain limitations were observed, particularly in generating captions for simpler or vague prompts. The model occasionally produced incomplete captions or repeated phrases, as seen with the hospital_bed and tiko objects when vague prompts were used. Additionally, the model’s ability to generate meaningful captions without a well-structured prompt was limited, resulting in less coherent or incomplete descriptions.

This indicates that while the model excels in specific scenarios, there is room for improvement in its robustness, particularly in handling a wider range of prompt styles and complexity levels. Fine-tuning or additional training could potentially address these issues, ensuring that the model performs consistently across various types of inputs.

e. Computational Considerations

The use of less resource-demanding models and pre-trained architectures, such as Paligemma and Gemma 2, enabled the system to generate captions without requiring extensive fine-tuning. However, some computational limitations remain, particularly when generating captions for large-scale datasets or highly detailed 3D objects. The need for prompt refinement and the model’s dependency on clear inputs add additional computational overhead, as the system may need to generate multiple captions before finding the most accurate one.

4. Model Constraints

Despite the promising results achieved with the Paligemma and Gemma 2 models, certain constraints were identified during the project. These limitations affect the model’s overall performance, scalability, and adaptability, especially when generating captions for complex 3D objects or handling less structured inputs.

a. Dependency on Prompts

The Paligemma model relies heavily on well-structured and prompts it’s been trained on to generate accurate and meaningful captions. As seen in the results, prompts that lacked specificity or context resulted in incomplete or incoherent captions. For example, the prompt "caption this 3d object" led to vague captions like "A , ." for the retro_computer or "An air." for the p38 airplane. This highlights the model’s limitation in handling ambiguous or under-defined prompts.

To mitigate this, substantial effort must be placed on prompt engineering, which adds complexity to the system and places an additional burden on the user. This constraint reduces the model's flexibility, as it struggles to generate accurate captions without explicit guidance.

b. Incomplete Captions

In some cases, the model generated captions with missing or incomplete information, even when provided with a more structured prompt. For instance, when using the prompt "caption en this 3d object," captions for objects like hospital_bed and tiko were fragmented or repetitive, failing to provide a coherent description. This behavior suggests that the model has difficulty synthesizing all the information from the multi-view images into a single, cohesive caption.

This constraint limits the model’s usability in real-world applications where generating fully descriptive captions for complex objects is necessary. Further refinement of the caption generation process, possibly through additional fine-tuning or post-processing, would be needed to address this issue.

c. Limited Robustness Across Object Types

While the model performed well for certain object types, such as the retro_computer and dumptruck, its performance dropped when faced with objects that had more abstract or minimalistic features, like tiko or frog. The simplified nature of these objects seems to confuse the model, leading to overly general or incomplete descriptions. This suggests that the model's internal representations may not fully capture the subtleties of certain 3D object types, particularly when the object lacks distinctive, easily describable features.

This constraint presents a challenge for generalizing the model to work effectively across a wide range of 3D object categories, especially when dealing with non-standard or abstract objects. Improving the model’s ability to handle more diverse and less conventional object types would enhance its overall robustness.

d. Difficulty with Caption Fusion

Although the Gemma 2 model is used to fuse captions from multiple viewpoints into a single description, the fusion process is not always seamless. In some cases, the resulting captions did not fully integrate the unique details from each viewpoint, leading to partial or incomplete descriptions. For example, the caption generated for the cyberpunk_laptop using the "caption this 3d object" prompt was fragmented and missed key details about the laptop's design and features.

This limitation in the fusion process affects the overall quality of the final caption, especially for objects with intricate or multi-faceted details. Enhancing the caption fusion mechanism to better consolidate information from multiple views would improve the completeness and accuracy of the generated descriptions.

e. Scalability and Computational Efficiency

While the project successfully employed less resource-intensive models, the scalability of the system remains a challenge, especially when handling larger datasets or highly complex 3D models. The reliance on pre-trained models such as Paligemma reduces the computational burden, but additional constraints arise when the model needs to process large volumes of data or generate captions in real-time.

Moreover, the need for careful prompt engineering introduces additional computational overhead, as multiple prompts or iterations may be required to generate high-quality captions. Although JAX and GPU/TPU support help alleviate some of the computational strain, further optimizations may be needed to ensure that the system can scale efficiently while maintaining performance across diverse object categories.

f. Use of External Rendering Software

The project currently relies on Blender for rendering 3D objects, which introduces an external dependency. Integrating a dedicated 3D rendering library would eliminate the need for external software, streamline the pipeline, and simplify implementation. Adopting such a library would make the system more self-contained and efficient.

5. Viewpoint Coverage

The project's use of multi-view rendering aimed to capture each 3D object from multiple angles to ensure comprehensive descriptions. By rendering images from eight specific viewpoints (e.g., front, back, top, bottom, and diagonals), the system provides diverse visual data for the Paligemma model to generate captions that cover different aspects of each object. However, while this multi-view approach improved coverage, certain limitations were observed.

a. Redundancy in Captions

Although the multi-view rendering captures different angles, there is often overlap in the generated captions, with repetitive descriptions across multiple views. For example, objects with relatively simple geometry, such as frog or tiko, resulted in captions with similar phrasing across views, providing little added value from each additional perspective. This redundancy affects the efficiency of the caption fusion process, as the model may not need all viewpoints to generate a cohesive final description for simpler objects.

To address this, a future enhancement could involve selectively reducing the number of viewpoints for simpler objects or implementing a redundancy filter to minimize overlapping information before fusion. This would help optimize the process and reduce computational overhead.

b. Insufficient Detail for Complex Objects

For complex objects with intricate features, such as cyberpunk_laptop and retro_computer, eight viewpoints may still not fully capture all relevant details. In some cases, critical details were either partially covered or omitted entirely in the final caption due to limitations in viewpoint selection. For example, certain angles might obscure features on the far side of the object, which impacts the completeness of the resulting description.

One potential solution to enhance viewpoint coverage would be to implement adaptive viewpoint selection based on object complexity. This approach would involve dynamically increasing the number of viewpoints for objects with more complex structures to capture additional visual information, ensuring that important details are not missed.

c. Difficulty in Synthesizing Multi-View Information

While the Gemma 2 model consolidates captions from various viewpoints, synthesizing these multi-view descriptions into a single coherent caption remains challenging. In some instances, the fusion process led to captions that emphasized certain views while neglecting details from others, resulting in incomplete or unbalanced descriptions. This challenge is particularly evident for objects like p38 airplane and hospital_bed, where details captured from some angles were omitted or downplayed in the final caption.

Improving the model’s ability to weigh and prioritize details from each viewpoint could enhance the final caption’s completeness and accuracy. Future work could explore refining the caption fusion mechanism to give equal consideration to critical details from all views, especially for objects with complex or asymmetrical structures.

d. Computational Constraints and Efficiency

The use of eight viewpoints, while beneficial for coverage, does increase computational requirements. Rendering and captioning multiple views for each object adds processing time, which could impact scalability in applications involving large datasets. Additionally, the need to process and fuse captions from multiple views introduces further computational demands, making the pipeline less efficient when dealing with high object volumes.

To balance viewpoint coverage with computational efficiency, future work could explore viewpoint reduction strategies that adaptively select essential views based on object characteristics, thereby optimizing resource usage without sacrificing caption quality.

6. Conclusion

The results of this project demonstrate the potential of a multi-view, prompt-based captioning system for generating descriptions of 3D objects. By leveraging pre-trained models like Paligemma and Gemma 2, the system achieved detailed and contextually relevant captions, particularly when using well-structured prompts and multi-view inputs. The use of eight specific viewpoints enabled greater coverage of each object, allowing the captions to capture a range of visual details essential for coherent descriptions.

However, several limitations were identified. The model’s dependency on prompt specificity, challenges with caption fusion, and redundancy in multi-view coverage highlight areas for improvement. Additionally, computational constraints related to viewpoint rendering and caption fusion limit the scalability of the pipeline, particularly in scenarios where high efficiency is required. Moreover, the project currently relies on Blender for rendering 3D objects, which introduces an external dependency. Despite these challenges, the generated captions effectively capture the primary features of most objects, confirming the value of a structured approach to 3D object captioning.

Future work should explore strategies to address these limitations, such as refining the prompt engineering process, improving the caption fusion mechanism, and optimizing viewpoint selection based on object complexity. These enhancements would contribute to a more robust, flexible, and scalable system for 3D object captioning, advancing the potential applications of automated captioning in various fields.

Future Work

This project demonstrated the potential of a multi-view, prompt-based captioning system for 3D objects, but several areas of enhancement remain to further improve its accuracy, robustness, and scalability. Future work could focus on refining the current methodology, testing quality with additional models, and enhancing the pipeline to better meet the demands of 3D object captioning tasks.

a. Evaluation of Caption Quality with Text-3D Models

To assess the accuracy and relevance of the generated captions, a text-3D model could be tested on the outputs. This approach would allow for a structured evaluation by measuring how well the generated captions align with 3D representations of the objects. Integrating a text-3D evaluation model could also offer insights into the model's ability to capture the most critical features, particularly for complex or multi-faceted objects.

b. Enhanced Prompt Engineering and Adaptive Prompts

The project results highlighted the dependency on structured prompts for high-quality outputs. Future work could explore advanced prompt engineering strategies, such as adaptive prompts that adjust based on object characteristics or viewpoint. Additionally, training or fine-tuning the Paligemma model on more diverse prompt structures might help reduce the system’s dependency on explicit prompt instructions, making it more flexible and scalable.

c. Caption Ranking and Multiple Caption Generation

For each viewpoint, generating multiple captions and implementing a caption ranking phase to select the best-fitting caption could improve the overall quality and coherence of the final output. This process would involve generating multiple captions per image, then using a ranking mechanism to choose the most accurate description based on specific criteria, such as relevance to the object's key features. This approach would help ensure the final caption incorporates the best aspects of each individual view.

d. Adaptive Viewpoint Selection

The current use of eight fixed viewpoints works well for diverse coverage, but an adaptive approach could optimize resource use. By dynamically selecting the number and type of viewpoints based on object complexity, the system could reduce redundancy for simpler objects while enhancing coverage for complex models. This adjustment would improve both the computational efficiency and quality of the final captions.

e. Integration of a Dedicated Rendering Library

The reliance on Blender as an external rendering software introduces dependencies that could be streamlined by integrating a dedicated 3D rendering library. This enhancement would eliminate the need for external software, making the pipeline more self-contained and improving its efficiency. Adopting an integrated rendering library would also simplify implementation, enabling more seamless control over rendering parameters.

f. Scalability and Computational Efficiency

Further optimization of the system's computational demands remains an important goal, especially for large-scale datasets. Implementing techniques such as dynamic viewpoint reduction and automatic prompt refinement could help reduce processing time and resources without sacrificing output quality. Additionally, the ability to upgrade to more efficient pre-trained models would ensure that the system can adapt to future developments in machine learning and 3D processing.

Conclusion

This project set out to develop a scalable and efficient pipeline for generating descriptive captions for 3D objects using a multi-view approach and pre-trained models. By integrating Blender for rendering, Paligemma for caption generation, and Gemma 2 for caption fusion, the system effectively produces coherent captions that capture diverse object features from multiple viewpoints. The Objaverse dataset provided a rich source of 3D models, enabling the project to test the pipeline on a variety of objects with differing complexities and visual details.

The results demonstrated that the quality of the captions relies heavily on prompt specificity, viewpoint selection, and caption fusion. While structured and recommended prompts yielded high-quality captions, vague prompts often resulted in incomplete or repetitive descriptions, emphasizing the need for prompt engineering. Similarly, the use of eight viewpoints improved object coverage but introduced redundancy, particularly for simpler objects. Despite these limitations, the system effectively captured key visual details for most objects, confirming the feasibility of multi-view 3D object captioning in an automated context.

Several limitations were identified, including the dependency on structured prompts, computational constraints, challenges in synthesizing information from multiple viewpoints, and reliance on Blender as an external tool for rendering. The Future Work chapter outlines potential improvements in these areas, such as integrating a dedicated rendering library, implementing caption ranking, and refining the prompt generation process to enhance the system's performance and flexibility.

Overall, this project demonstrates the potential of multi-view 3D object captioning for generating descriptive and accurate text, paving the way for applications in virtual reality, 3D modeling, and autonomous systems. With continued advancements, this system could serve as a foundation for further innovation in automated 3D content description and analysis.

References

[1] T. Luo, C. Rockwell, H. Lee, and J. Johnson, “Scalable 3D Captioning with Pretrained Models,” Jun. 15, 2023, *arXiv*: arXiv:2306.07279. Accessed: Jul. 25, 2024. [Online]. Available: http://arxiv.org/abs/2306.07279

[2] L. Beyer *et al.*, “PaliGemma: A versatile 3B VLM for transfer,” Jul. 10, 2024, *arXiv*: arXiv:2407.07726. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2407.07726

[3] G. Team *et al.*, “Gemma 2: Improving Open Language Models at a Practical Size,” Oct. 02, 2024, *arXiv*: arXiv:2408.00118. Accessed: Nov. 02, 2024. [Online]. Available: http://arxiv.org/abs/2408.00118

[4] T. Yu, X. Lin, S. Wang, W. Sheng, Q. Huang, and J. Yu, “A Comprehensive Survey of 3D Dense Captioning: Localizing and Describing Objects in 3D Scenes,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1322–1338, Mar. 2024, doi: 10.1109/TCSVT.2023.3296889.

[5] Z. Qi *et al.*, “GPT4Point: A Unified Framework for Point-Language Understanding and Generation,” Dec. 05, 2023, *arXiv*: arXiv:2312.02980. Accessed: Jul. 25, 2024. [Online]. Available: http://arxiv.org/abs/2312.02980

[6] M. Deitke *et al.*, “Objaverse: A Universe of Annotated 3D Objects,” Dec. 15, 2022, *arXiv*: arXiv:2212.08051. Accessed: Jul. 25, 2024. [Online]. Available: http://arxiv.org/abs/2212.08051

[7] L. Xue *et al.*, “ULIP: Learning a Unified Representation of Language, Images, and Point Clouds for 3D Understanding,” Jun. 12, 2023, *arXiv*: arXiv:2212.05171. Accessed: Jul. 25, 2024. [Online]. Available: http://arxiv.org/abs/2212.05171

[8] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation,” Feb. 15, 2022, *arXiv*: arXiv:2201.12086. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2201.12086

[9] R. Vedantam, C. L. Zitnick, and D. Parikh, “CIDEr: Consensus-based Image Description Evaluation,” Jun. 03, 2015, *arXiv*: arXiv:1411.5726. Accessed: Nov. 02, 2024. [Online]. Available: http://arxiv.org/abs/1411.5726

[10] R. Mokady, A. Hertz, and A. H. Bermano, “ClipCap: CLIP Prefix for Image Captioning,” Nov. 18, 2021, *arXiv*: arXiv:2111.09734. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2111.09734

[11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners”.

[12] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2556–2565. doi: 10.18653/v1/P18-1238.

[13] H. Agrawal *et al.*, “nocaps: novel object captioning at scale,” Sep. 30, 2019, *arXiv*: arXiv:1812.08658. Accessed: Nov. 02, 2024. [Online]. Available: http://arxiv.org/abs/1812.08658

[14] J. Wang *et al.*, “GIT: A Generative Image-to-text Transformer for Vision and Language,” Dec. 15, 2022, *arXiv*: arXiv:2205.14100. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2205.14100

[15] O. Sidorov, R. Hu, M. Rohrbach, and A. Singh, “TextCaps: a Dataset for Image Captioning with Reading Comprehension,” Aug. 04, 2020, *arXiv*: arXiv:2003.12462. Accessed: Nov. 02, 2024. [Online]. Available: http://arxiv.org/abs/2003.12462

[16] R. Ramos, B. Martins, D. Elliott, and Y. Kementchedjhieva, “SmallCap: Lightweight Image Captioning Prompted with Retrieval Augmentation,” Mar. 28, 2023, *arXiv*: arXiv:2209.15323. Accessed: Aug. 21, 2024. [Online]. Available: http://arxiv.org/abs/2209.15323

[17] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context,” Feb. 21, 2015, *arXiv*: arXiv:1405.0312. Accessed: Nov. 02, 2024. [Online]. Available: http://arxiv.org/abs/1405.0312

[18] A. Radford *et al.*, “Learning Transferable Visual Models From Natural Language Supervision,” Feb. 26, 2021, *arXiv*: arXiv:2103.00020. Accessed: Sep. 27, 2024. [Online]. Available: http://arxiv.org/abs/2103.00020