

SnapScore: A Novel NLP Technique to Evaluate Text-to-Image Models

Jash Mitesh Dalal
jdalal@umass.edu

Rahul Hemal Shah
rhshah@umass.edu

Kartikay Gupta
kartikaygupta@umass.edu

Chaitali Agarwal
cragarwal@umass.edu

1 Problem statement

1.1 Introduction

Stable diffusion has emerged as a state-of-the-art text-to-image generative model, capable of producing visually stunning images from textual prompts. However, existing evaluation metrics like Inception Score and Fréchet Inception Distance (FID) have limitations in accurately capturing the semantic relevance and quality of the generated images with respect to the input text prompts.

Our work aims to address this gap by introducing SnapScore, a novel evaluation metric that leverages image captioning and text similarity techniques to quantify the alignment between the generated images and the corresponding text prompts.

1.2 Problem Formulation

The key problem we aim to solve is the lack of effective evaluation metrics to assess the quality and relevance of images generated by text-to-image models like Stable Diffusion. Existing metrics such as Inception Score and FID have several drawbacks, including:

1. Failing to capture gradual improvements in iterative models
2. Not reflecting distortion levels in generated images
3. Producing inconsistent results when varying sample sizes

Additionally, metrics like LPIPS measure perceptual similarity between images but require ground truth images and do not consider variations in image generation based on textual input.

Our SnapScore metric aims to overcome these limitations by incorporating an image captioning

model and finding the semantic similarity between the generated caption and the input text prompt, providing a more holistic evaluation of the generated image's quality and relevance

1.3 Impact and Applications

The implementation of SnapScore aims to provide a robust tool for developers and researchers to better understand and refine the capabilities of diffusion (text-to-image) models. By quantitatively measuring how well the images reflect the content and intent of their textual prompts, SnapScore can help enhance the model's training process, leading to improvements in both precision and contextual accuracy. Furthermore, this metric is crucial for various applications across industries such as digital marketing, creative design, and automated content generation, where the quality of visual representations directly impacts user engagement and satisfaction. Through this project, we seek to bridge the gap between human visual perception and algorithmic image generation, paving the way for more intuitive and reliable generation on images encapsulating the users prompt.

2 What you proposed vs. what you accomplished

In our project, we set out to create an evaluation metric based on the following main objectives:

1. **Use Stable Diffusion Model:** We aimed to use the Stable Diffusion model to generate images from given prompts. This involved understanding and using the Stable Diffusion model for image generation.

Achieved: 100%, we incorporated user prompt using diffusion models to generate images which were further used for analysis in part 2 and 3.

2. **Captioning Generated Images:** After generating images using the Stable Diffusion model, we planned to use an LSTM model to generate captions for these images.

Achieved: 100%, We achieved this step by training a LSTM model on the coco dataset and then performed the captioning tasks by integrating it with the image generation process.

3. **Calculate Text Similarity:** Once we generated captions, we intended to calculate the text similarity between these captions and the input prompts.

Achieved: 100%, We achieved this step and evaluated the SnapScore using both our LSTM model and a pre-trained open source model - Lavis.

3 Related work

Image generation models play a crucial role in creating new visual content from existing datasets, with both unconditional and conditional generation tasks being prominent in the field. Evaluating the performance of these models is essential to understand their capabilities and limitations. Here, we delve into recent research and developments in image generation models and their evaluation.

We propose to build an evaluation metric to compare the images generated from a diffusion model. Currently, the most sought metrics for evaluating generated images are *Inception Score* (Salimans et al., 2016) and *Fid Score*. The Inception Score is a metric for automatically evaluating the quality of image-generative models. This metric was shown to correlate well with human scoring of the realism of generated images from the CIFAR-10 dataset. However, the Inception Score suffers from suboptimality of the Inception Score itself and problems with the popular usage of the Inception Score as shown by the work of (Barratt and Sharma, 2018).

Another popular metric for evaluating image generation is *Fréchet Inception Distance (FID)*. FID is a mathematical model that measures the similarity between generated images and real images by comparing their feature representations extracted from a pre-trained classifier, such as the Inception network. Inception’s poor representation of the rich and varied content generated by modern text-to-image models, incorrect normality assumptions, and poor sample complexity leads

to a major drawback of FID score. The work of (Jayasumana et al., 2024) demonstrate that FID contradicts human raters, it does not reflect the gradual improvement of iterative text-to-image models, it does not capture distortion levels, and it produces inconsistent results when varying the sample size.

Other works on image evaluation include LPIPS (Zhang et al., 2018). It measures the perceptual similarity between images by comparing the structure of image patches. It assesses whether the structure of a patch has changed and by how much, which can be particularly useful for tasks like image editing or super-resolution where structural fidelity is important. However, this needs to have an image ground truth and doesn’t take into consideration the possible variations of the image. In short, this too doesn’t provide a holistic view of the quality and relevance of the generated image in the context of the input given. In our approach, we provide a novel method to evaluate the relevance and similarity of the generated image from textual input by incorporating an image captioning model and finding the semantic similarity between the generated caption and the input prompt.

One of the important parts of our proposal is building an LSTM model to generate image captioning. Notable related work in this domain includes that of (Wang et al., 2016) and (Singh et al., 2023) which show that LSTM accomplishes significant results in caption generation. They demonstrate that bidirectional LSTM models achieve highly competitive performance to the state-of-the-art results on caption generation even without integrating additional mechanisms (e.g. object detection, attention model, etc.) and significantly outperform recent methods on retrieval tasks.

Finally, our metric relies on text comparison metrics including METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004) and, BLEU: (Papineni et al., 2002) and Cosine similarity (Lahitani et al., 2016).

4 Dataset

Data Overview

For the SnapScore project, we use the Microsoft COCO data to evaluate the performance of stable diffusion model and to train our LSTM model to generate captions.

Basic Statistics:

1. **Size:** Over 330,000 images.
2. **Annotations:** Each image is paired with five captions, providing a diverse range of textual descriptions essential for training the captioning model.
3. **Categories:** The dataset encompasses 80 object categories, ensuring a wide variety of scenes and objects.

Task Description and Challenges

The primary tasks involve:

1. **Complexity in Image Representation:** Capturing the depth and nuances of textual descriptions in image form is complex due to the variability and subtlety in natural language.
2. **Quality of Reverse-Captioning:** Generating coherent and contextually relevant captions from images, ensuring they accurately reflect the depicted scenes.
3. **Model Alignment:** Ensuring the image generation model (Stable Diffusion) and the captioning model are well-aligned in terms of the semantic content they process and produce.
4. **Metric generation:** One of the key challenges is to calculate a metric that encompasses all the features of the sentence and also take into account the syntactic variances in generated captions.

Example Input/Output Pairs

1. **Image Example** See Figure
2. **Ground Truth:**
 - (a) A woman leaning on a pole holding two traffic signs.
 - (b) a person leaning on a stop sign with a skate board
 - (c) A young girl with a skateboard is leaning against a stop sign.
 - (d) A girl leans on the stop sign with her skateboard.
 - (e) A girl is leaning against the stop sign with her skateboard.



Figure 1: Generated image for a person leaning on a stop sign with a skate board

Data Preprocessing

This section focuses on data preprocessing for training the LSTM model.

Preprocessing Steps

To prepare the dataset for both the text-to-image and image-to-text components:

1. **Image Normalization:** Standardizing the size and scale of images to ensure uniform input to the model.
2. **Text Tokenization and Normalization:** Tokenizing and normalizing text to streamline processing and improve model performance.
3. **Vocabulary Standardization:** Establishing a consistent vocabulary for both encoding text prompts and decoding generated captions.

Data Processing

Caption Generation: Post image generation, captions are generated using the trained LSTM and pre-trained Lavis models to describe the content, focusing on accuracy and relevance to the initial text prompt.

5 Baselines

Baseline Models

The project does not directly utilize traditional baseline models such as Inception Score (IS),

Frechet Inception Distance (FID), or Learned Perceptual Image Patch Similarity (LPIPS) commonly found in image generation tasks. Instead, it focuses on text-to-image generation and the subsequent evaluation of generated captions using text similarity metrics. The models employed in this project include:

1. **Stable Diffusion:** Used for generating high-quality images from textual descriptions.
2. **BLIP (Berkeley Language-Image Pre-training model):** Used for generating captions from images, which are then evaluated against the input prompts to assess the text-to-image generation quality. Lavis is based on this model.

How They Work and Results

1. Stable Diffusion:

- (a) **How it works:** This model is a text-to-image diffusion model that conditions on text inputs via a CLIP encoder. It progressively refines the image through a series of latent space adjustments, guided by the textual input.
- (b) **Results:** Produces detailed and contextually relevant images based on textual prompts. The quality of the images is contingent on the model's ability to interpret the text semantically.

2. BLIP:

- (a) **How it works:** BLIP employs a vision Transformer coupled with a language model to generate captions that describe the images. It is pre-trained on a large dataset, enabling it to generate accurate and relevant captions for a wide range of images.
- (b) **Results:** Offers captions that are assessed for semantic similarity with the original text prompts using metrics such as BLEU, ROUGE, and METEOR.

Why These Baselines Over Others

These models were chosen based on their capability to generate and evaluate high-quality, contextually accurate images and captions from text. This is essential for the objectives of this project. Unlike traditional GAN-based image generation

models, Stable Diffusion and BLIP provide advanced capabilities in handling detailed and varied textual descriptions for image synthesis and captioning, critical for assessing model performance in text-to-image tasks.

Hyperparameters and Training Setup

1. Stable Diffusion:

- (a) **Hyperparameters:** Uses defaults provided by the diffusers library, with a `guidance_scale` set to 8.5 to adjust the fidelity and relevance of the generated images.
- (b) **Training:** No training involved; directly used the pre-trained models.

2. BLIP:

- (a) **Hyperparameters:** Default settings for the pre-trained model, optimized for generating captions on the COCO dataset images.
- (b) **Training:** No additional training involved; directly uses pre-trained models for evaluation.

Train/Validation/Test Split

Data Handling: The evaluation of generated captions does not involve a traditional training process but utilizes pre-existing models applied to test datasets. The Microsoft COCO dataset is split into training, validation, and test sets, with these models applied to generate or evaluate on unseen test data. The division is already given by MSCOCO with 164K images split into training (83K), validation (41K) and test (41K) sets

Integrity in Experiments

1. No hyperparameter tuning on the test set to ensure that the evaluations remain unbiased and reflect true model performance under general conditions.
2. Evaluation metrics are calculated post-hoc, ensuring that no information leakage occurs from test data during any form of training or tuning.

6 Our approach

Our project involves developing an evaluation metric which revolves around three main objectives: (1) Training an LSTM model to generate

captions. (2) captioning images generated by the Stable Diffusion model (3) text similarity to create a SnapScore.

Step 1: Training a ResNet-LSTM Model

Our objective is to develop an evaluation metric to assess the efficiency of a diffusion model. The evaluation consists of first generating the caption of the image produced by the diffusion model. To produce the caption we trained an image captioning model using a ResNet encoder and an LSTM (Long Short Term Memory) decoder on MSCOCO dataset.

6.0.1 Encoder (ResNet)

The encoder part of the model is a pre-trained ResNet model, which is used to extract visual features from the input images. Specifically:

1. A pre-trained ResNet model- ResNet-50 is loaded with weights pre-trained on the ImageNet dataset.
2. The input image is passed through the ResNet model to obtain a feature map tensor.
3. The feature map tensor is processed (e.g., spatially flattened, average pooled) to obtain a fixed-size visual feature vector representing the image.

6.1 Decoder (LSTM)

The decoder part of the model is an LSTM network, which takes the visual features from the encoder and generates the caption word-by-word:

1. The visual feature vector from the encoder is used to initialize the hidden state and cell state of the LSTM.
2. At each time step, the LSTM takes the previous hidden state, previous word (starting with a special start token), and outputs a new hidden state and a probability distribution over the vocabulary.
3. The word with the highest probability is selected as the next word in the generated caption.
4. This process continues until the LSTM generates a special end token or reaches a maximum caption length.

The model is trained in an end-to-end fashion on MSCOCO, which contains images and their corresponding captions. The training objective is to maximize the likelihood of generating the correct captions given the input images. Techniques like teacher forcing, beam search, and optimization algorithms like Adam or RMSProp are used to update the model weights and improve caption generation performance

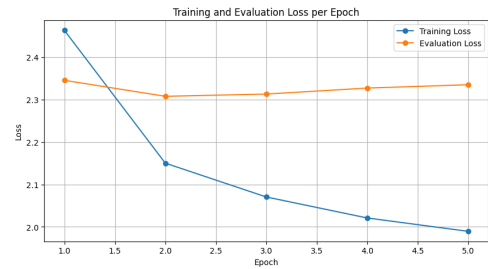


Figure 2: Graph for loss in training and evaluation dataset

6.2 Step 2: Generating of captions of images using Diffusion Model

We generated images using the descriptions from the validation set of MSCOCO dataset using the stable-diffusion-v1-4 (Rombach et al., 2022), which is an open-source model accessible through HuggingFace. We then used these images to generate the captions using our trained ResNet-LSTM model and also on our baseline Lavis Model.

6.3 Step 3: SnapScore Calculation: Evaluating the Generated Images

We have used five prominent text comparison metrics to calculate our SnapScore. Specifically, we have used

1. **METEOR:** Measures the alignment of words and phrases between the generated text and reference text, considering synonyms and stemming for better correlation with human judgment. (Banerjee and Lavie, 2005)
2. **ROUGE:** Evaluates the quality of the summary by counting overlapping units such as n-grams, word sequences, and word pairs between the generated text and reference text (Lin, 2004)
3. **BLEU:** Calculates the precision of n-grams in the generated text against reference texts,

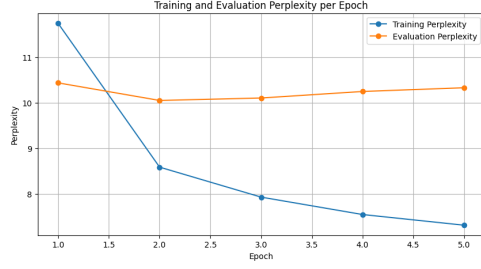


Figure 3: Graph for perplexity in training and evaluation dataset

penalizing for short lengths to avoid overly brief outputs. (Papineni et al., 2002)

4. **Cosine similarity:** Measures the cosine of the angle between two vectors of an inner product space, which in this case represent the generated and reference text embeddings, to determine their similarity (Lahitani et al., 2016)

We take a weighted average of these scores to calculate the SnapScore. Our approach differs from previous work as we seek to establish a self-created evaluating metric tool. We tried to bring an innovative solution to address the common challenge of the scarcity of readily available image-generation tools bundled with their performance evaluation based on textual prompts.

$$\begin{aligned}
 \text{snap_score} = & 0.20 \times \text{score}_{\text{BLEU}} \\
 & + 0.20 \times \text{score}_{\text{ROUGE}} \\
 & + 0.25 \times \text{score}_{\text{METEOR}} \\
 & + 0.35 \times \text{score}_{\text{cosine}}
 \end{aligned} \quad (1)$$

We give more weight to text similarity like cosine similarity (30%) over pure n-gram matching metrics like BLEU (15%).

FID score has several limitations including being dataset, dependent on the model, sensitive, high computational cost, image characteristic bias and preprocessing sensitivity.

SnapScore on the other end is not dependant on the distribution of the dataset. It isn't much dependant on the image characteristics and preprocessing sensitivity. However, SnapScore is dependant on the image-to-text model.

We have completed the entire proposed implementation. We have used the StableDiffusion-Pipeline for using the stable diffusion model, lavis models for baseline evaluation of caption generation, nltk for tokenization, numpy, skimage,

torchvision and torch.nn for building the LSTM model. We used a few references online to build our model and get inspiration. This includes (Nayak, 2020), (Soliao, 2022), (Li et al., 2023), (Sairam et al., 2021). However, we completed the implementation ourselves with these references. Our files "nlp project" and "LSTM Image Cap model" are our work. We used Colab Pro to run our files.

6.4 Outputs

Here we have generated the results for our concept by showcasing a snippet of generated images, we also have additional details appended along with it such as User Prompts, Human Annotation, generated using LSTM model, Captions generated using Lavis model, SnapScore generated using LSTM, SnapScore generated using Lavis

In our results we have observed that the LSTM approach give us a good SnapScore which is even better as compared to the Lavis scores, indicating that it could better caption

Comparative Performance Evaluation

Image Name	Model	BLEU	ROUGE	METEOR	Cosine	SNAP Score
Animal Eating Grass	LSTM	0.2985	0.4973	0.4409	0.3408	0.3886
	Lavis	0.2627	0.4672	0.4361	0.3112	0.3639
Doctor in a Suit	LSTM	0.3439	0.6114	0.6954	0.3486	0.4869
	Lavis	0.2938	0.5750	0.6169	0.3680	0.45680
Apple and Banana	LSTM	0.4594	0.6969	0.6988	0.5056	0.5829
	Lavis	0.3328	0.6969	0.6861	0.5727	0.5779

Table 1: Model Performance Comparison on Various Prompts

Model	FID Score	Snap Score using LSTM	Snap Score using Lavis
Stable Diffusion	12.63 (Yasunaga et al., 2022)	0.4861	0.3948

Table 2: Comparison of FID Score and Snap Scores using LSTM and Lavis

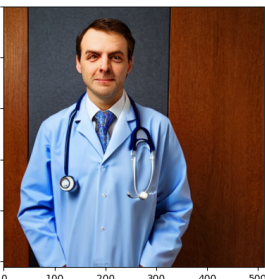
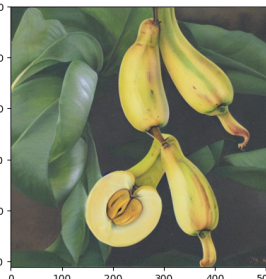

	Image - 1	Image - 2	Image - 3
Images			
User Prompt	Doctor in suit	Apple and banana	Animal eating grass, detailed, 8k
Human Annotation	a man wearing a suit and tie standing with a stethoscope	a bunch of bananas and fruits hanging on a tree	animal standing in the grass with its mouth open
LSTM - Captions generated	a man in a suit and tie standing in front of a television	a bunch of bananas and oranges on a table	a large bear is standing in the grass
Lavis - Captions generated	a man wearing a blue lab coat and a stethoscope	a bunch of bananas hanging from a tree	a close up of a dog with its mouth open
SnapScore using LSTM model	0.4869	0.5829	0.3886
SnapScore using Lavis model	0.4568	0.5779	0.3639

Table 3: Table with Images, Captions and SnapScore Generated

7 Error analysis

This section discusses the types of inputs that challenge the baseline models and the proposed SnapScore approach, along with a manual error analysis to identify common error patterns.

Baseline Model Failures

1. Stable Diffusion:

- (a) **Failures:** Struggles with abstract concepts and ambiguous prompts that do not specify clear visual elements. Examples include "dreams of a robot" or "silence in the mountains," which lead to inconsistent and varied image outputs.
- (b) **Commonalities:** These failures often exhibit a lack of semantic clarity in the text prompts, causing the model to generate irrelevant or overly generalized images.

2. LSTM:

- (a) **Failures:** Inaccurate captions when images contain complex scenes with multiple objects or actions, or subtle details that are significant to the overall context of the image.
- (b) **Commonalities:** The errors typically occur in images with high semantic density, where the model overlooks critical details or misinterprets the visual context.

SnapScore Approach Failures

- (a) **Difficult Examples:** The SnapScore sometimes fails to accurately measure the semantic alignment between generated images and the text prompts for abstract concepts or when the text involves cultural or context-specific nuances.
- (b) **Commonalities:** Errors often arise in scenarios where the textual prompt requires specific cultural or contextual knowledge that the image generation and captioning models fail to encapsulate.

Manual Error Analysis

We conducted an error analysis focusing on the automated evaluation of generated images and

captions. This analysis aimed to identify systematic issues in how the models handle specific types of prompts.

1. **Methodology:** Using the developed SnapScore metric, we automatically evaluated a large set of images and captions generated by Stable Diffusion and BLIP. We then selected examples with the lowest scores for detailed review to understand the types of errors occurring.
2. **Observational Findings:** Our review focused on cases where there was a significant divergence between the generated captions and the original text prompts. These examples often involved abstract concepts, cultural specificities, or complex scenes that were not described accurately in the captions.

3. Observational Findings from Models:

- (a) **LSTM:** The LSTM model demonstrated a strong ability to generate text that closely aligns with the input prompts, capturing both the semantic content and the intended meaning effectively. This capability is reflected in the higher SnapScore values, indicating that the LSTM-generated captions are more accurate and contextually relevant.
- (b) **Lavis:** While Lavis also produces accurate text descriptions of the images, it tends to use a more varied vocabulary, resulting in captions that are less consistent with the input prompts. This greater diversity in language can sometimes lead to lower SnapScore values due to slight deviations in the exact wording and phrasing compared to the LSTM model.

4. Common Challenges:

- (a) **Stable Diffusion Failures:** The model occasionally generated images that were visually appealing but did not accurately represent the details or themes specified in the text prompts, particularly for prompts requiring detailed or specialized knowledge.

(b) **LSTM Caption Failures:** The model sometimes produced generic or partially relevant captions that failed to capture the nuances or specific elements of the images, suggesting a limitation in the model's understanding of complex visual content.

5. **Patterns in Errors:** Errors frequently occurred with prompts that were abstract or metaphorical, where the literal interpretation led to mismatches between the text and the generated content. Additionally, prompts with culturally or contextually dense themes often resulted in inaccuracies.

6. **Implications for Model Training:** The errors highlight potential deficiencies in the training datasets used for both the image generation and captioning models. The lack of diverse examples, especially in abstract and culturally rich contexts, appears to limit the models' effectiveness.

Recommendations for Improvement

To address these shortcomings, we suggest the following strategies:

1. **Enhanced Training Data:** Enrich the training datasets with a broader range of themes, including more abstract, non-literal, and culturally diverse content.
2. **Model Refinement:** We could explore advanced model architectures or training techniques that improve the semantic understanding of complex prompts and their accurate visual representation.

8 Contributions of group members

List what each member of the group contributed to this project here. For example:

1. **Kartikay Gupta:** Did data collection and processing, LSTM model execution with scores for all examples generated by the text-to-image model
2. **Chaitali Agarwal:** Literature review, Lavis model, metric formulation and documentation.
3. **Jash Dalal:** Model training, Saving LSTM model, Loading pre-trained model to caption images, fine-tuning

4. **Rahul Shah:** Created the stable diffusion models to generate images, Prompt Engineering, generating and evaluating SnapScore and Error Analysis

9 Conclusion

This project provided valuable insights into the challenges and intricacies of developing and evaluating text-to-image models using advanced NLP tools. We successfully implemented the SnapScore metric, leveraging it to assess the semantic fidelity of images generated from textual descriptions by models like Stable Diffusion, coupled with LSTM for comparison of caption generation.

Project Takeaways

1. The integration of text-to-image generation with NLP-based evaluation proved to be an effective approach to measure the qualitative aspects of generated images beyond traditional metrics.
2. The project underscored the complexity of accurately translating textual prompts into coherent images, particularly for abstract concepts and detailed requests, highlighting the nuanced understanding required by generative models.
3. The project then dives deeper into NLP applications of using the score studies in the Class like BLEU, Rouge and some new like METEOR, Cosine for calculating snap score on both the models and then present a comparative study of these models on same image and caption generated by different models and measuring their SnapScore.

Challenges Encountered

1. One significant challenge was ensuring that the generated images maintained relevance to the text prompts, especially when the prompts were abstract or culturally specific, which often led to unexpected or irrelevant visual outputs.
2. Developing a robust evaluation metric that could universally and reliably measure semantic accuracy across diverse prompt types was more complex than initially anticipated.

Surprising Findings

1. The accuracy of the LSTM model in generating relevant captions for complex images was surprisingly high, though not without occasional errors that provided insights into limitations of current models.
2. The variability in SnapScore outcomes across different types of prompts provided unexpected insights into the strengths and weaknesses of the underlying model architectures.

Future Directions

Looking ahead, several enhancements could be pursued:

1. **Model Iteration:** Further development of the SnapScore metric to include more nuanced linguistic and visual elements could refine evaluation capabilities.
2. **Semantic Consistency Score:** Introduce a metric assessing the contextual consistency within images. Evaluate whether elements in the image contextually align with each other based on the textual description, such as seasonal appropriateness or interactions among depicted subjects. This would involve advanced image recognition and semantic analysis technologies to detect and evaluate contextual discrepancies.
3. **Visual-Semantic Alignment:** Employ scene graph matching techniques to compare the relationships and interactions between objects in the image against those specified in the text. This approach requires constructing and comparing scene graphs from both the generated images and the textual prompts, ensuring that not only are objects correctly depicted, but they also interact as described.
4. **Cross-Model Evaluation:** Applying SnapScore across different generative models could help benchmark capabilities and inspire competitive enhancements in the generative model landscape.
5. **Real-World Applications:** Exploring practical applications of this technology in fields like digital marketing, creative media, and educational content creation, where custom

image generation from text can provide substantial value.

6. **Generating multiple captions:** To enhance the robustness and accuracy of SnapScore, it should be designed to generate multiple captions for a given image. These multiple captions created for every image, would reduce the dependency on metrics that rely heavily on exact word matching, allowing for a more nuanced and comprehensive evaluation of the semantic alignment between the generated images and the input prompts.

Overall, this project not only advanced our understanding of the capabilities and limitations of current text-to-image technology but also set the groundwork for significant future advancements in automated content generation.

AI Disclosure

1. Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.

(a) Yes, ChatGPT 4 was used

If you answered yes to the above question, please complete the following as well:

1. If you used a large language model to assist you, please paste **all** of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
 - (a) How to calculate BLEU Score in python
 - (b) How to calculate ROUGE Score in python
 - (c) How to calculate METEOR Score in python
 - (d) How to calculate Cosine Score in python
 - (e) Provide a detailed guide on how to set up a comparative study to benchmark different AI models on the same tasks using metrics like BLEU, ROUGE, METEOR, and Cosine similarity
 - (f) My training takes lot of time to run and I want to speed up the process. What could be the possible ways to do so?

2. **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?

- (a) Yes I used ChatGPT 4 and 3.5 and tried to get some relevant answers to various questions.

References

- Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Goldstein, J., Lavie, A., Lin, C.-Y., and Voss, C., editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Barratt, S. and Sharma, R. (2018). A note on the inception score.
- Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., and Kumar, S. (2024). Rethinking fid: Towards a better evaluation metric for image generation.
- Lahitani, A. R., Permanasari, A. E., and Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6.
- Li, D., Li, J., Le, H., Wang, G., Savarese, S., and Hoi, S. C. (2023). LAVIS: A one-stop library for language-vision intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 31–41, Toronto, Canada. Association for Computational Linguistics.
- Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Nayak, J. (2020). Cnn-lstm architecture and image captioning. Accessed: 2024-05-16.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, page 311–318, USA. Association for Computational Linguistics.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.
- Sairam, G., Mandha, M., Prashanth, P., and Swetha, P. (2021). Image captioning using cnn and lstm. In *4th Smart Cities Symposium (SCS 2021)*, volume 2021, pages 274–277.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242.
- Singh, P., Kumar, C., and Kumar, A. (2023). Next-lstm: a novel lstm-based image captioning technique. *International Journal of Systems Assurance Engineering and Management*, 14:1492–1503.
- Soliao (2022). Image-captioning. USA. GitHub.
- Wang, C., Yang, H., Bartz, C., and Meinel, C. (2016). Image captioning with deep bidirectional lstms.
- Yasunaga, M., Aghajanyan, A., Shi, W., James, R., Leskovec, J., Liang, P., Lewis, M., Zettlemoyer, L., and Yih, W.-t. (2022). Retrieval-augmented multimodal language modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.