Image to Prompts - Final Report

Abhiroop Tejomay Kommalapati[1], Hajar Sadeghi[2], Shriya Mandarapu[3], and Ajay Digumarthy[4]

[1]akommala@iu.edu
[2]hasadeg@iu.edu
[3]shmanda@iu.edu
[4]ajdigu@iu.edu

1 Introduction

The emergence of text-to-image models like Stable Diffusion [1] has led to the development of a new area of prompt engineering. This field involves both artistic and scientific aspects, and machine learning professionals and researchers are actively working to comprehend the connections between prompts and the images they produce. They are exploring questions such as whether adding ”4k” to a prompt is the most effective way to make it more photographic, whether minor alterations to prompts can result in significantly different images, and how the order of prompt keywords influences the generated scene. Our goal in this work is to produce models that can effectively reverse the diffusion process that generates a given image from a prompt and to gain an understanding of how reversible the latent relationship is between images and prompts.

Our work has been primarily inspired by the ”Stable Diffusion: Image to Prompts” competition[2], which was released by Kaggle in February 2023 (ongoing). We intend to utilize pairs of `(image, prompt)` where the images are generated by Stable Diffusion 2.0 given a prompt, to develop models that can predict the prompts that generated said images. Concretely, our model will take an image that has been produced by the diffusion process as input and generate an embedding vector for the prompt that was used to generate the image. To address this challenge, there are two primary approaches that are pertinent:

1. Predict the prompts using text captioning, and then extract embeddings - Using an image captioning model like BLIP [3] we can create a text caption for the images.
2. Predict the embeddings directly - Train a model that predicts the embedding directly. Perhaps, encode the image via CLIP[4] embeddings and predict the prompt embeddings.

2 Background and Related Work

Diffusion models have been a concept for many years, but it wasn’t until a 2015 paper by Jascha Sohl-Dickstein et al [5]. that they were formalized in their modern form. The authors used thermodynamics to model a diffusion process, much like a drop of milk diffusing in a cup of tea. The central idea is to train a model to learn how to reverse the process by starting from a fully mixed state and gradually separating the milk from the tea. In 2020, Jonathan Ho et al. [6] created a diffusion model that could generate realistic images, which they called a denoising diffusion probabilistic model (DDPM).

Later, in 2021, Robin Rombach, Andreas Blattmann, et al.[1] introduced latent diffusion models where the diffusion process occurs in latent space rather than pixel space. The researchers also applied various conditioning techniques to guide the diffusion process using text prompts, images, or any other inputs, allowing for the quick creation of high-quality images of anything from an insect robot making a delicious meal to anything else you desire.

Finally, a powerful pre-trained latent diffusion model named Stable Diffusion was open-sourced in August 2022 through a collaboration between LMU Munich and several companies, including StabilityAI, Runway, EleutherAI, and LAION. We are using the outputs of the Stable Diffusion 2.0 as the dataset for our work to predict the prompts and determine the reversibility of the latent relationships.

Fundamentally, the problem at hand is an image captioning problem. CLIP (Contrastive Language-Image Pre-Training)[3] and BLIP (Bootstrapping Language-Image Pre-training)[4] are two recent models for multi-modal understanding and generation of text and images, which have been shown to achieve state-of-the-art performance on a variety of vision and language tasks, including image captioning.

CLIP is a model that uses a contrastive learning objective to learn joint representations of text and images. The model is trained on a large-scale dataset of text and images, such as the COCO dataset, where each image is paired with a caption. The goal of the model is to learn a joint embedding space where the representations of semantically similar text and images are close to each other, while the representations of dissimilar text and images are far apart.

BLIP, on the other hand, is a model that extends the CLIP architecture to include a language generation component. The model is trained using a bootstrapping approach, where it is first pre-trained on a large-scale dataset of text and images, such as Conceptual Captions, using a self-supervised learning objective. The pre-trained model is then fine-tuned on a smaller dataset of captioned images, where the goal is to generate natural language descriptions of visual content.

3 Methods

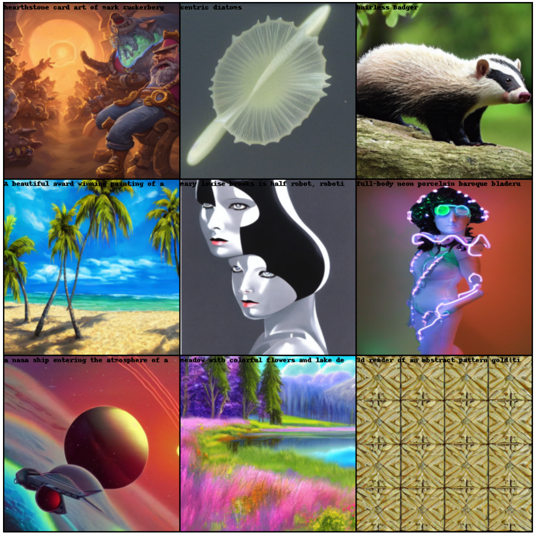
The following sections explain our dataset and methods.

3.1 Dataset and Exploratory Data Analysis

We noticed that the competition-provided dataset only contained 7 pairs of images and prompts. This encouraged participants to either create their own data or use open-source datasets. To save time, we opted to use the DiffusionDB[7] dataset for our project. This dataset contains 14 million images that were generated using Stable Diffusion with prompts and hyperparameters specified by actual users. Given its extensive size and diverse characteristics, this dataset presents several opportunities for research, including exploring the connection between prompts and generative models, identifying deepfakes, and creating human-AI interaction tools that help users effectively utilize these models. The DiffusionDB dataset comprises two datasets, the complete dataset, and a smaller 2 million dataset. We began by training our model on a subset of 200,000 images from the 2 million images dataset.

Exploratory data analysis was conducted on the DiffusionDB dataset, where image and prompt pairs were analyzed and presented visually. Figure 1 displays a few instances of such image-prompt pairs from the dataset.

Figure 1: Images and prompts from the DiffusionDB dataset


hearthstone card art of mark zuckerberg
centric diatoms
hairless Badger
A beautiful award winning painting of a
mary louise brooks is half robot, roboti
full-body neon porcelain baroque bladeru
a nasa ship entering the atmosphere of a
meadow with colorful flowers and lake de
3d render of an abstract pattern gold ti

We train an ensemble of 4 models to convert images to prompt embeddings. The four models we used are:

1. Vision Transformer
2. BLIP
3. CLIP Interrogator (pre-trained)
4. OFA (pre-trained)

The CLIP interrogator, BLIP, and OFA largely play the same role in the pipeline. That is, the images are fed to these models and their corresponding prompts are returned. Then, the prompts are passed through the pre-trained Sentence Transformer to get their corresponding embeddings. The Vision Transformer [8] however, predicts the embeddings directly. These prompt embeddings are produced by the same Sentence Transformer [9] model, which acts as the ground truth for the Vision Transformer. Figure 2 shows the pipeline for the CLIP Interrogator/BLIP and the OFA, and figure 3 shows the pipeline for the Vision Transformer.

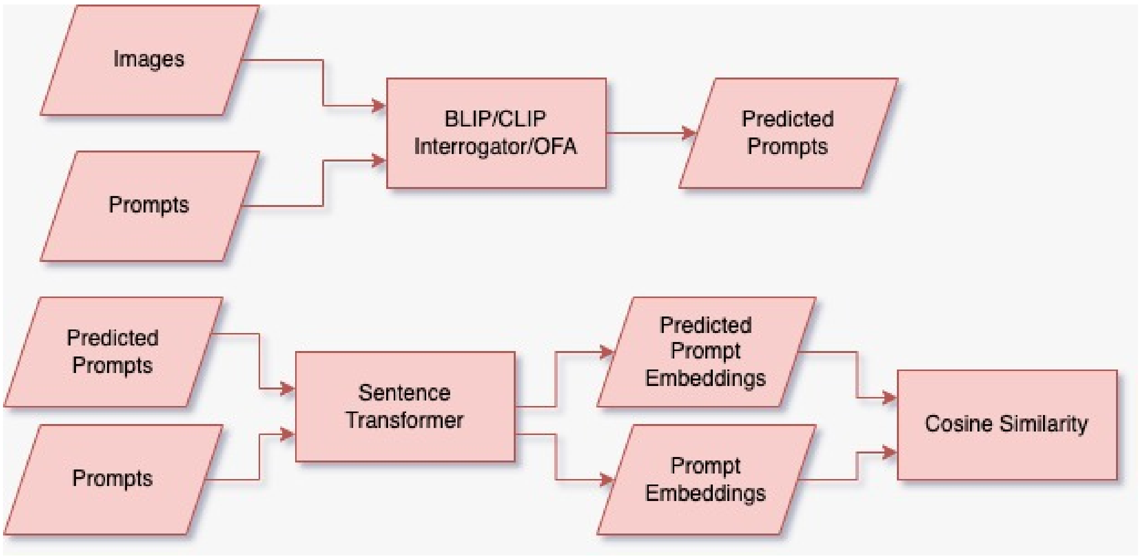


Figure 2: BLIP/CLIP Interrogator and OFA Pipeline

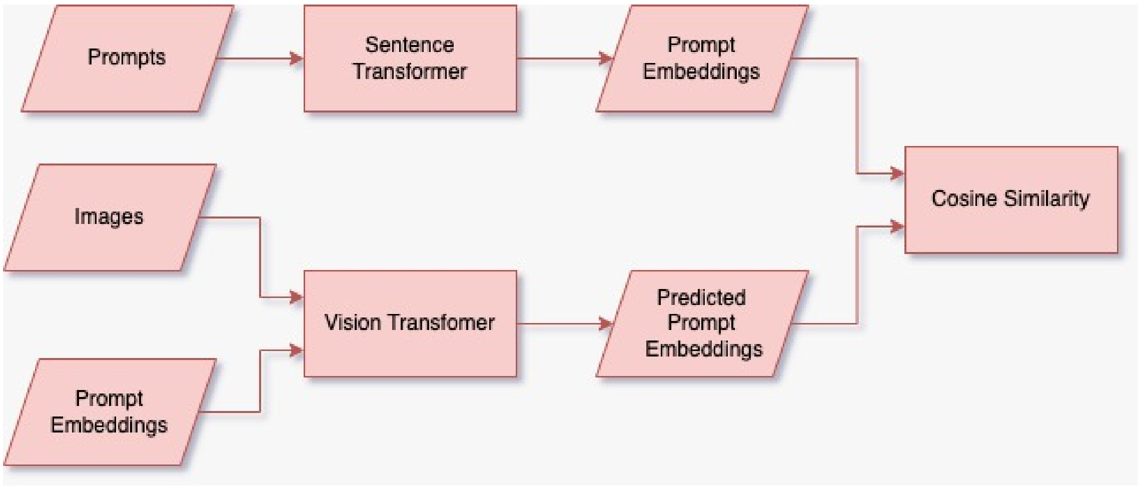


Figure 3: Vision Transformer Pipeline

3.2 Vision Transformer Baseline

We opted to develop a Vision Transformer model in tandem with the CLIP Interrogator. To initiate this process, we created dataset and dataloader pipelines that employed the DiffusionDB dataset and a pretrained sentence transformer model to produce pairs of prompt embeddings and images. The pretrained sentence transformer checkpoint utilized was the all-MiniLM-L6-v2. The images were resized to 224x224 pixels and normalized using the ImageNet protocol. The batch size we utilized was 64 and the size of the embedding vector was 384.

The Vision Transformer model training pipeline was developed utilizing the Vision Transformer implementation from the `timm` library. The Vision Transformer (ViT) was introduced as a new approach to image recognition, demonstrating competitive results on several benchmark datasets. ViT achieves this by dividing an image into patches, which are then processed by a transformer-based model, enabling end-to-end training and better generalization to previously unseen data. In order to facilitate result reproducibility, the entire pipeline was implemented in PyTorch Lightning, a library that eliminates PyTorch's boilerplate code while preserving maximum flexibility and scalability performance. The pre-trained checkpoint for the Vision Transformer was `vit_base_patch16_224`, with a patch embedding size of 16. The Vision Transformer predicts the prompt embeddings of size 384.

Finally, we also implemented code to log the model's loss to Weights and Biases, making experiment tracking easier. The loss we used was the cosine embedding loss defined by:

$$\operatorname{loss}(x, y) = \begin{cases} 1 - \cos\left(x_1, x_2\right), & \text{if } y = 1 \\ \max\left(0, \cos\left(x_1, x_2\right) - \text{margin}\right), & \text{if } y = -1 \end{cases}$$

where x_1 and x_2 are predicted and ground truth prompt embeddings respectively, and y is a tensor of ones equal to the batch size.

3.3 BLIP

We decided to utilize other pre-trained models to improve our results using an ensemble approach. One of the models we employed is the BLIP (Backbone-Label-Interpolation-Prediction). BLIP utilizes a bootstrapping approach to pre-train a unified vision-language model for understanding and generating text from images. The bootstrapping process begins by pre-training a vision model on a large image dataset to obtain visual features. These visual features are then used to pre-train a language model that predicts textual descriptions or prompts. Subsequently, the pre-trained language model is utilized to generate text for a large number of images, and these generated texts are then used to fine-tune the vision model.

The fine-tuned vision model is then utilized to generate visual features for a larger and more diverse image dataset. These visual features, along with textual embeddings obtained from a pre-trained language model, are used to create a joint visual-language embedding. The joint embedding is then passed through a series of backbone models, which are pre-trained classifiers on various datasets, to predict candidate labels.

These candidate labels are then interpolated with the textual embedding to generate a set of label embeddings. Finally, a separate pre-trained language model is utilized to generate a textual description or prompt based on the label embeddings.

We used the `blip-image-captioning-base` BLIP checkpoint on Hugging Face, and finetuned it on our dataset for 10 epochs, given the extreme training time on our GPU (10 hr/epoch on an NVIDIA Quadro RTX 5000). We used a batch size of 4, a learning rate of 1e-4 with Cosine Annealing learning rate scheduler with minimum learning rate and weight decay of 1e-6. The predicted prompts from the BLIP are passed through the Sentence Transformer to get a 384-dimensional prompt embedding, and subsequently the cosine similarity of the original and predicted prompt embeddings is computed.

3.4 CLIP Interrogator

The CLIP Interrogator is a tool for optimizing text prompts to match a given image, which combines OpenAI's CLIP and Salesforce's BLIP. It utilizes OpenCLIP, which supports several pre-trained CLIP models, with `ViT-H-14/laion2b_s32b_b79k` being the best one for Stable Diffusion 2.0. The CLIP Interrogator pipeline [10] has several steps, starting with passing an image to BLIP to get its main description, followed by passing it to CLIP to obtain its embedding. The embeddings are then compared to those obtained from labels in lists, and the top four with the most similarity are chosen. The outgoing prompt for the CLIP portion is formed using four main lists, namely artists.txt, flavors.txt, mediums.txt, and movements.txt, along with popular artwork sites. Finally, the resulting texts are combined and returned as an image description or prompt for generating an image. Figure 4[11] shows how CLIP interrogator can be used to generate prompts for stable diffusion.

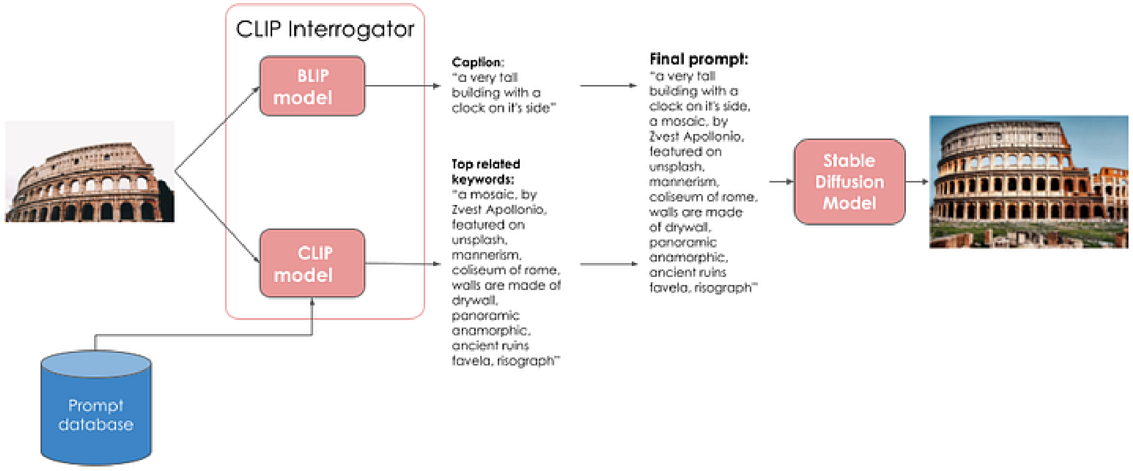
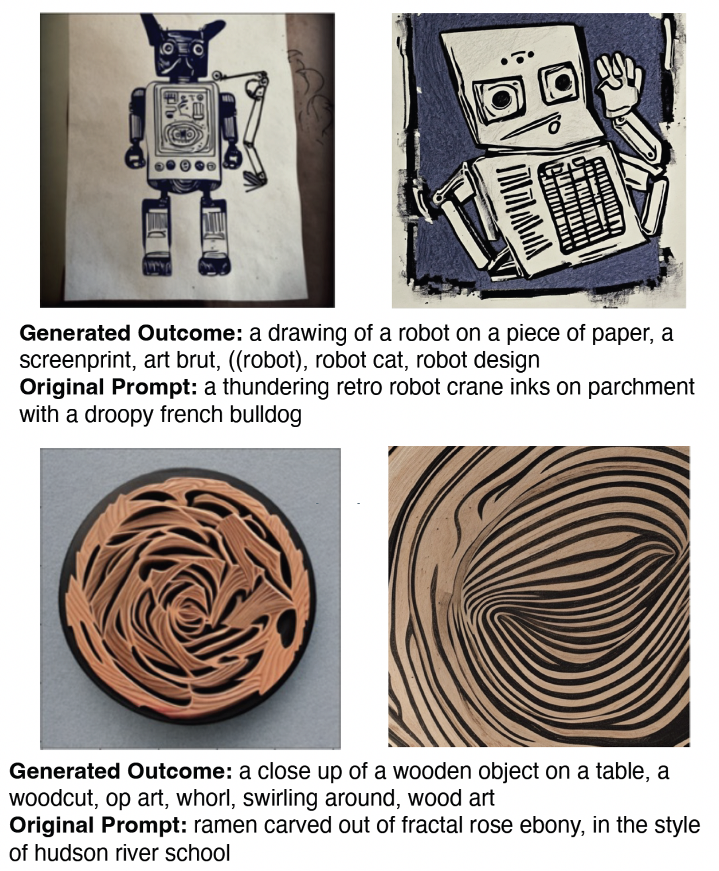

CLIP Interrogator
BLIP model
CLIP model
Prompt database
Caption: "a very tall building with a clock on it's side"
Top related keywords: "a mosaic, by Zvest Apollonio, featured on unsplash, mannerism, coliseum of rome, walls are made of drywall, panoramic anamorphic, ancient ruins favela, risograph"
Final prompt: "a very tall building with a clock on it's side, a mosaic, by Zvest Apollonio, featured on unsplash, mannerism, coliseum of rome, walls are made of drywall, panoramic anamorphic, ancient ruins favela, risograph"
Stable Diffusion Model

Figure 4: Pipeline of the CLIP Interrogator

Some outputs of the CLIP Interrogator are shown in Figure 5:



Generated Outcome: a drawing of a robot on a piece of paper, a screenprint, art brut, ((robot), robot cat, robot design
Original Prompt: a thundering retro robot crane inks on parchment with a droopy french bulldog

Generated Outcome: a close up of a wooden object on a table, a woodcut, op art, whorl, swirling around, wood art
Original Prompt: ramen carved out of fractal rose ebony, in the style of hudson river school

Figure 5: Outputs from the CLIP Interrogator

We use the `CLIP-ViT-H-14-laion2B-s32B-b79K` checkpoint and did not fine-tune it given the training time constraints. The predicted prompts from the CLIP Interrogator are passed through the Sentence Transformer to get a 384-dimensional prompt embedding, and subsequently the cosine similarity of the original and predicted prompt embeddings is computed.

3.5 OFA

We also employed the OFA (Once For All) vision-language model, which is a pre-trained architecture that aims to achieve a balance between model efficiency and performance. The OFA model is based on the idea of designing a single neural network that can adapt to different resource constraints (e.g., varying computational power or memory) by selecting a subset of the network's layers at runtime. For our work, we used the `OFA-large-caption` checkpoint and did not fine-tune it any further due to training time constraints.

Similar to the BLIP and CLIP Interrogator models, the predicted prompts are passed through the Sentence Transformer to get a 384-dimensional prompt embedding and the cosine similarity is computed.

3.6 Results

We tested each of the 4 models in the ensemble as well the final ensemble on the test set consisting of 40000 images. The ensemble embedding was generated by performing a weighted sum of the embeddings of the individual models. the weights used for the models are 0.2, 0.3, 0.4, and 0.1 for the Vision Transformer, BLIP, CLIP Interrogator, and OFA respectively, chosen based on each individual model's performance on the test set. The final cosine similarities of the 4 models individually and the ensemble if given in Table 1.

Model	**Cosine Similarity**
Vision Iransformer	0.63
BLIP	0.65
CLIP Interrogator	0.66
OFA	0.62
Ensemble	0.68

Table 1: Cosine Similarities on the test set.

4 Discussion

In this study, we successfully generated meaningful prompts for the Stable Diffusion images, and our output showed that the images generated using our predicted prompts were highly similar to the original images. However, due to the significant training time required by the models, we employed pre-trained vision-language models without fine-tuning, to our ensemble to increase performance. For context, training a BLIP model on 200,000 images took 10 hours to run a single epoch, causing our original plan to train on 2 million images to change. Furthermore, we were unable to train our models for more epochs, resulting in a cosine similarity of 68% for the ensemble, which could be improved by running for more epochs.

Our project was motivated by the quest to understand the relationships between prompts and the images they generate. We generated prompt embeddings using different models, but a reliable model that could reverse the diffusion process was required to answer this question with certainty. To address this, we suggest training the model on larger data sets and for a more extended period as future research.

Our current results show how the predicted prompts affect the generated image by Stable Diffusion. For instance, in the robot scenario, the generated prompts produced a robot that closely resembled the original. However, further research is needed to answer questions like how modifying prompt embeddings, such as adding perturbations, affects the diffusion model. Our future work will aim to explore these questions.

5 Conclusion

Our goal in this work was to produce models that can reverse the diffusion process that generates a given image from a prompt and to gain an understanding of how reversible the latent relationship is between images and prompts. We trained an ensemble of 4 models to convert images to prompt embeddings. We generated an ensemble embedding by performing a weighted sum of the individual model embeddings. Despite the significant training time required for the models, our ensemble was able to produce similar images to the original ones. However, to gain a better understanding of the relationship between prompts and the images they generate, a more reliable model that can reverse the diffusion process is needed. Our study has provided valuable insights into prompt engineering, and we believe that it will stimulate further research in this exciting field.

References

[1] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. DOI: 10.48550/ARXIV.2112.10752. URL: https://arxiv.org/abs/2112.10752.

[2] Will Cukierski Ashley Chow inversion. *Stable Diffusion - Image to Prompts*. 2023. URL: https://kaggle.com/competitions/stable-diffusion-image-to-prompts.

[3] Junnan Li et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. DOI: 10.48550/ARXIV.2201.12086. URL: https://arxiv.org/abs/2201.12086.

[4] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. DOI: 10.48550/ARXIV.2103.00020. URL: https://arxiv.org/abs/2103.00020.

[5] Jascha Sohl-Dickstein et al. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. DOI: 10.48550/ARXIV.1503.03585. URL: https://arxiv.org/abs/1503.03585.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. DOI: 10.48550/ARXIV.2006.11239. URL: https://arxiv.org/abs/2006.11239.

[7] Zijie J. Wang et al. "Large-Scale Prompt Gallery Dataset for Text-to-Image Generative Models". In: *arXiv:2210.14896 [cs]* (2022). URL: https://arxiv.org/abs/2210.14896.

[8] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[9] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].

[10] *BLIP+CLIP — CLIP Interrogator*. https://www.kaggle.com/code/leonidkulyk/lb-0-45836-blip-clip-clip-interrogator.

[11] *Diversify photo database with Clip Interrogator*. https://medium.com/@silkworm/diversify-photo-database-with-clip-interrogator-5dd1833be9f5.