Stable Diffusion

Images to Prompts

Group 07
Aman Patel
Kamal Panchal
Praveen Thanniru
Rahul Parupati

Agenda

- Introduction
- Competition Overview & Goal
- Data Sources
- Model selection
- Evaluation
- Conclusion

Introduction

- Stable Diffusion is a deep learning model that generates detailed images based on text descriptions.
- It can also be used for tasks like inpainting and outpainting.
- The model was developed by Stability AI in collaboration with academic researchers and non-profit organizations.

Image generated by stable diffusion

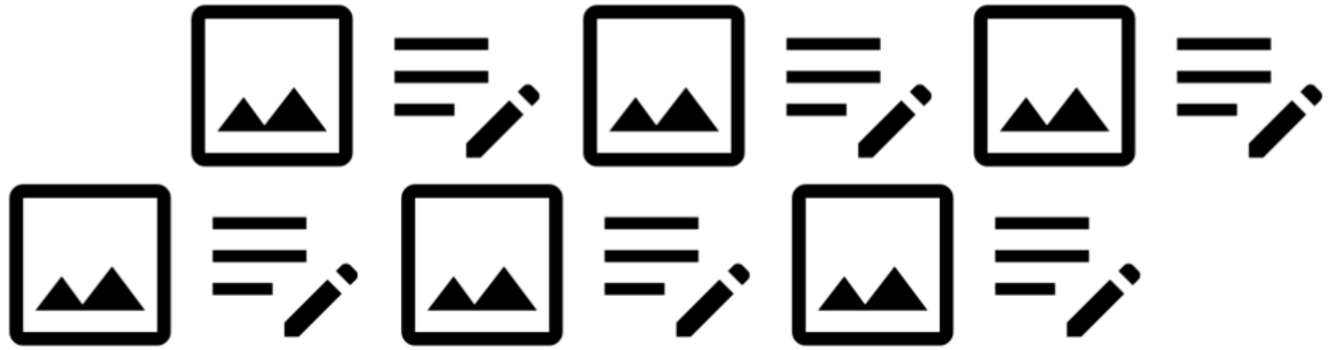
Prompt used :

"a photograph of an astronaut riding a horse"

Goal of Competition

Image to Prompts

Competition overview & Goal



- To reverse the direction of generative text to image model.
- Instead of generating image from prompt, we have to predict prompt from generated image.
- Creating a model that can reliably invert diffusion process that generated to a given image.
- Text embeddings to the generated prompts to be submitted as deliverables.

Data Sources

'ultrasaurus holding a black bean taco in the woods, near an identical cheneosaurus'

Data Explorer

3.24 MB

- images
  - 20057f34d.png
  - 227ef0887.png
  - 92e911621.png
  - a4e1c55a9.png
  - c98f79f71.png
  - d8edf2e40.png
  - f27825b2c.png
- prompts.csv
- sample_submission.csv

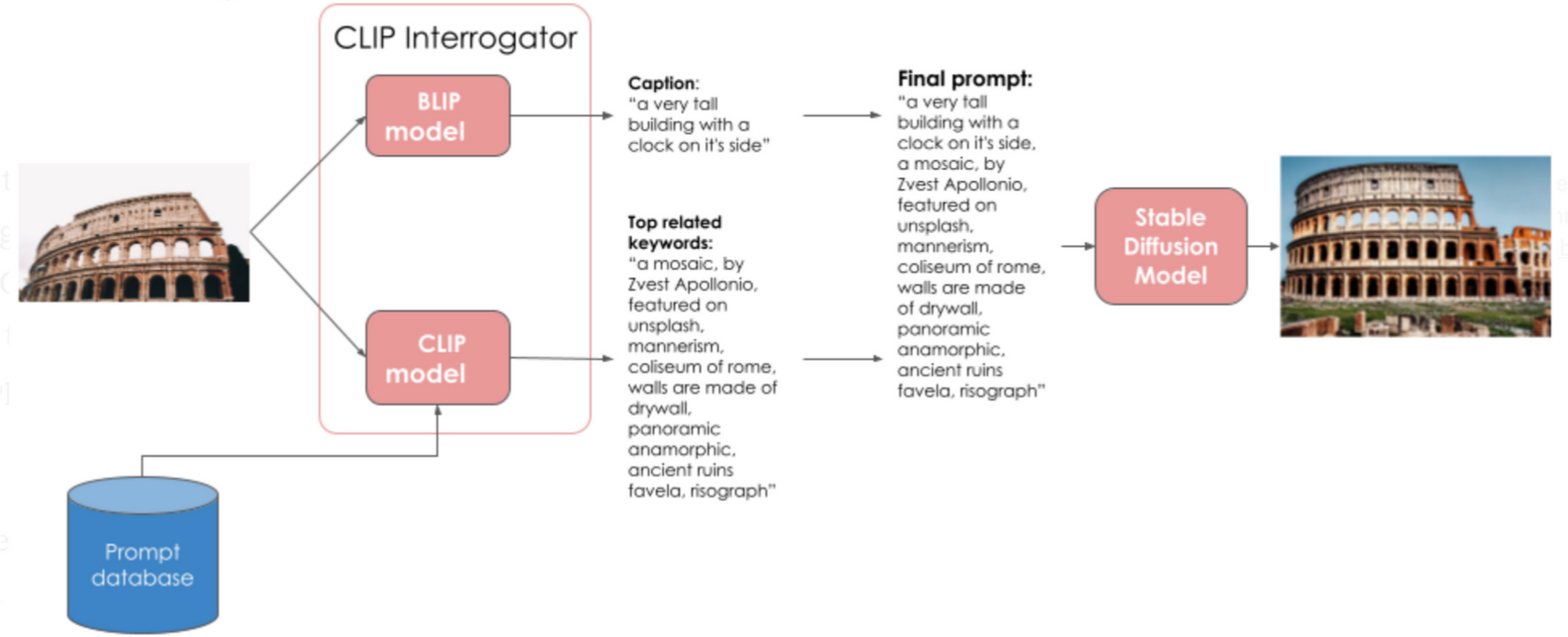
Related Work

CLIP Interrogator

Contrastive Language Image Pre-training

The CLIP Interrogator is a prompt engineering tool that combines OpenAI’s CLIP and Salesforce’s BLIP to optimize text prompts to match a given image.

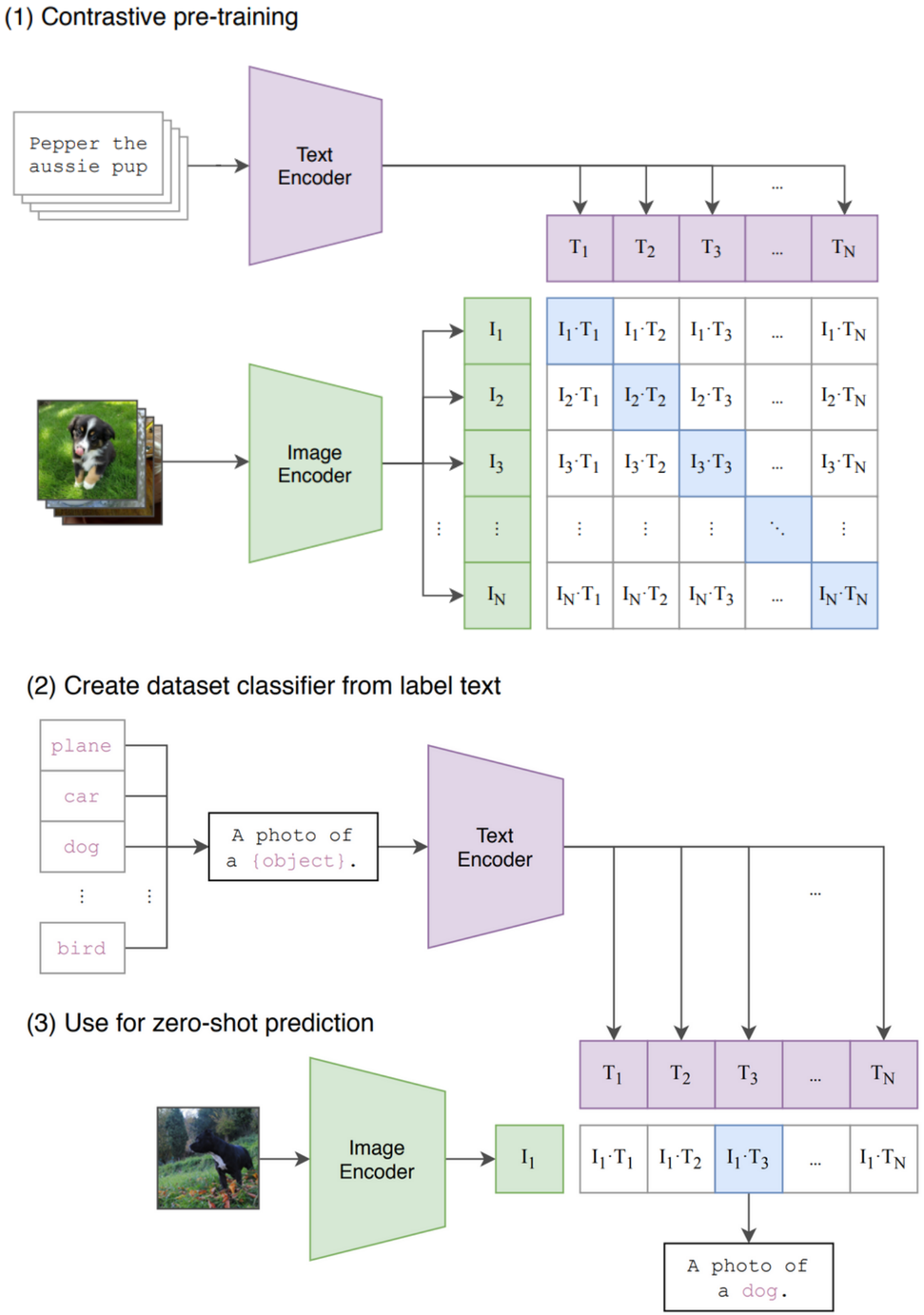
CLIP interrogator



- The BLIP model receives an input image and creates a caption.
- The CLIP model is used for text-image retrieval. CLIP model searches the Prompt database for the top-ranking keywords that match the content of the input image.
- The caption from BLIP and the top-related words from CLIP will be merged to form the final prompt.

CLIP explained

1. The model receives a batch of N <image-text> pairs.
2. The Text Encoder is a standard Transformer model with GPT2-style modifications. The Image Encoder can be either a ResNet or a Vision Transformer.
3. For every image in the batch, the Image Encoder computes an image vector. The first image corresponds to the I1 vector, the second to I2 , and so on.
4. Similarly, the textual descriptions are squashed into text embeddings [T1, T2 … TN]
5. In a contrastive fashion, a matrix will be created such that the best possible image prompt pairs will be formed on diagonal.
6. Zero shot predictions can be done on trained CLIP model by passing label texts to the text encoder and testing with unseen image.



CLIP Interrogator

Want to figure out what a good prompt might be to create new images like an existing one?
The CLIP Interrogator is here to get you answers!

You can skip the queue by duplicating this space and upgrading to gpu in settings: Duplicate Space

Prompt | Analyze

Image

CLIP Model

ViT-L (best for Stable Diffusion 1.*)

Mode

best | fast | classic | negative

Submit

Output

a painting of a cat in a dark room, granny weatherwax, unsettling image, game icon asset, by Wes Anderson, from berserk, inspired by Odd Nerdrum, its name is greeny, cushart, the artist has used bright, sansa, anime visual of a cute cat, gryffindor

Training Data

- We used 154320 image prompt pairs available in kaggle to finetune our custom proposed models.
- These images are generated by those corresponding prompts using Stable Diffusion 2.0

	filepath	prompt
0	/kaggle/input/diffusiondb-2m-part-0001-to-0100...	a portrait of a female robot made from code, v...
1	/kaggle/input/diffusiondb-2m-part-0001-to-0100...	dream swimming pool with nobody
2	/kaggle/input/diffusiondb-2m-part-0001-to-0100...	a beautiful paint of cultists dancing surround...
3	/kaggle/input/diffusiondb-2m-part-0001-to-0100...	frontal portrait of ragged, worried twin women...
4	/kaggle/input/diffusiondb-2m-part-0001-to-0100...	a stunning portrait of an asian samurai with l...
...	...	...
154315	/kaggle/input/diffusiondb-2m-part-1901-to-2000...	obama transformed into a penguin, a combinatio...
154316	/kaggle/input/diffusiondb-2m-part-1901-to-2000...	new york invaded by nazis, concept art
154317	/kaggle/input/diffusiondb-2m-part-1901-to-2000...	a owlish, aquiline picture of an owl sitting o...
154318	/kaggle/input/diffusiondb-2m-part-1901-to-2000...	a owlish, elaborate painting of an owl sitting...
154319	/kaggle/input/diffusiondb-2m-part-1901-to-2000...	a rose with the face of jerry garcia

154320 rows × 2 columns

Sentence Transformer

- SentenceTransformers is a Python framework for state-of-the-art sentence, text and image embeddings.

all-MiniLM-L12-v2

Description:	All-round model tuned for many use-cases. Trained on a large and diverse dataset of over 1 billion training pairs.
Base Model:	microsoft/MiniLM-L12-H384-uncased
Max Sequence Length:	256
Dimensions:	384
Normalized Embeddings:	true
Suitable Score Functions:	dot-product (`util.dot_score`), cosine-similarity (`util.cos_sim`), euclidean distance
Size:	120 MB
Pooling:	Mean Pooling
Training Data:	1B+ training pairs. For details, see model card.
Model Card:	https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

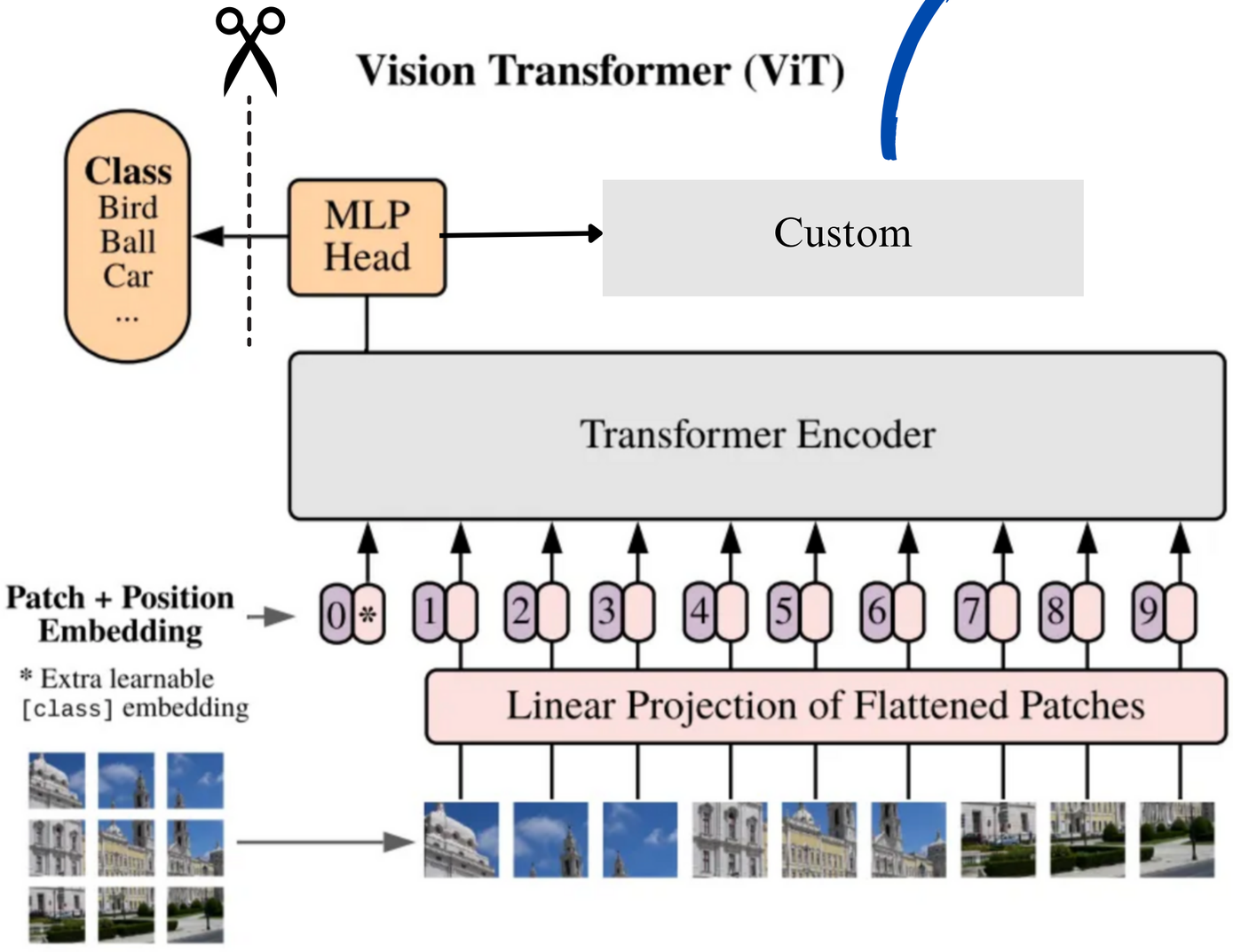
Proposed Model

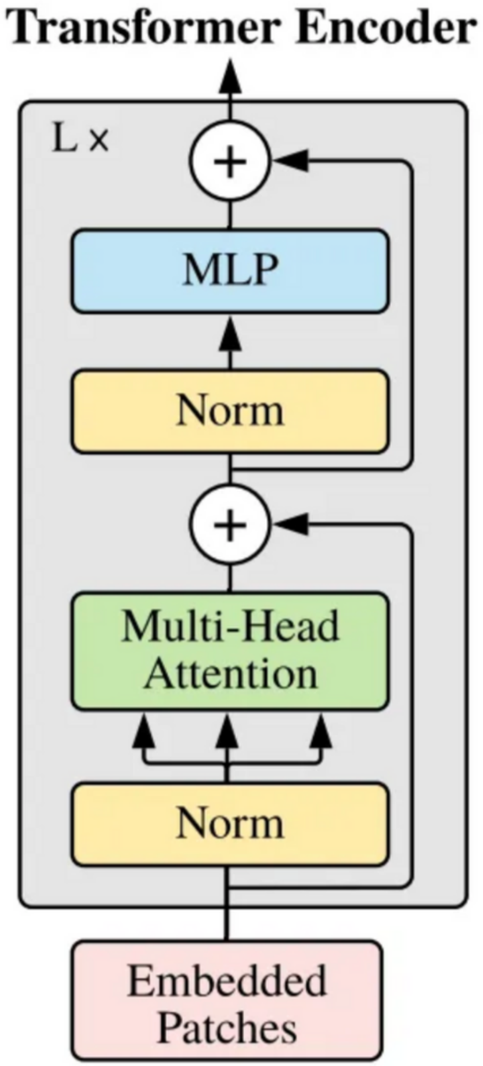
Customized Vision Transformer - replaced final classification head with 3 fully connected layers separated by Dropout and GELU activation function.

```
VisionTransformer-225                  [-1, 768]                   0
          Linear-226                 [-1, 1024]             787,456
         Dropout-227                 [-1, 1024]                   0
            GELU-228                 [-1, 1024]                   0
          Linear-229                  [-1, 512]             524,800
         Dropout-230                  [-1, 512]                   0
            GELU-231                  [-1, 512]                   0
          Linear-232                  [-1, 384]             196,992
================================================================
Total params: 87,155,840
Trainable params: 87,155,840
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 437.44
Params size (MB): 332.47
Estimated Total Size (MB): 770.49
----------------------------------------------------------------
```

ViT Backbone + Customized fully connected layers

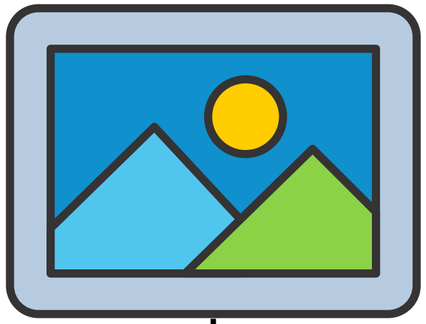
3 fully connected layers with
final layer consisting 384 units



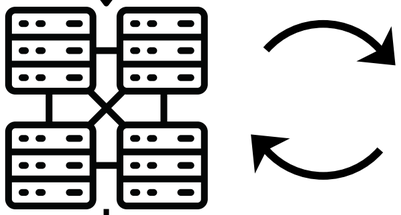


Custom ViT training

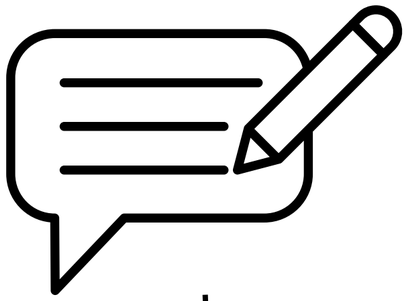
Image

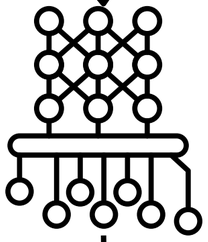


Custom ViT



Prompt





Sentence Transformer

Image Embedding

Text Embedding

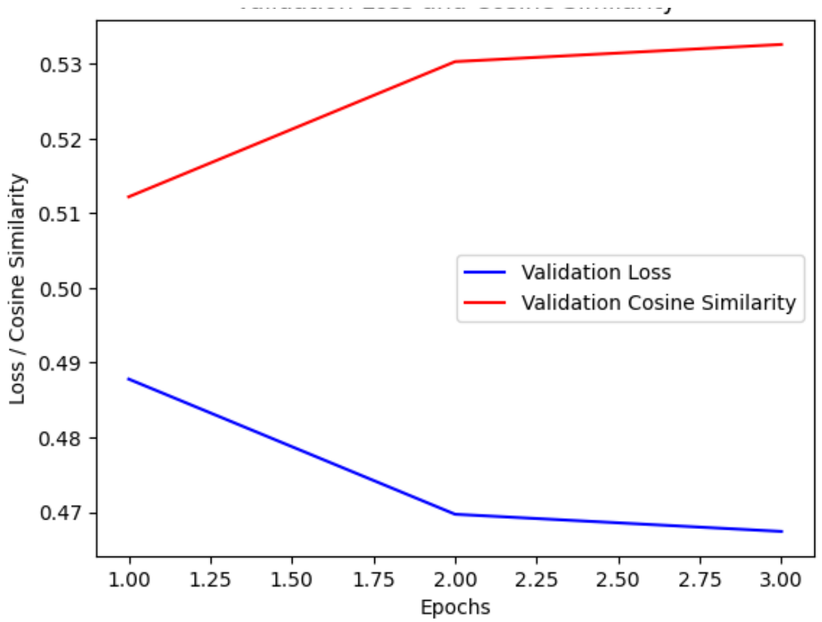
0.2	0.4	0.9	--

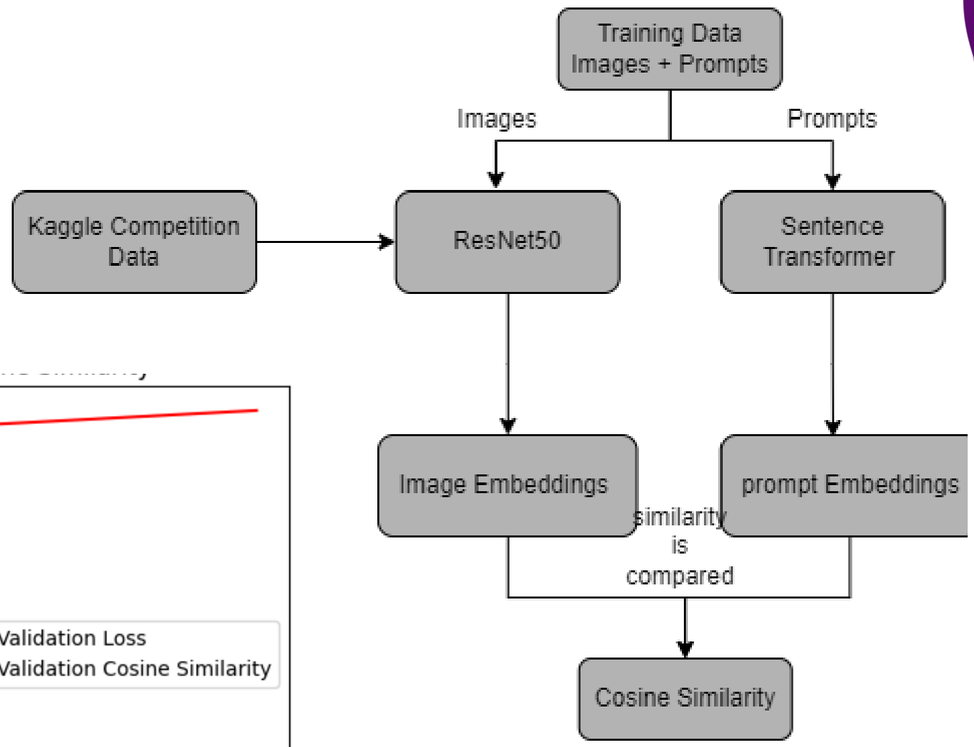
≈

0.6	0.3	0.8	--

Modeling(ResNet)

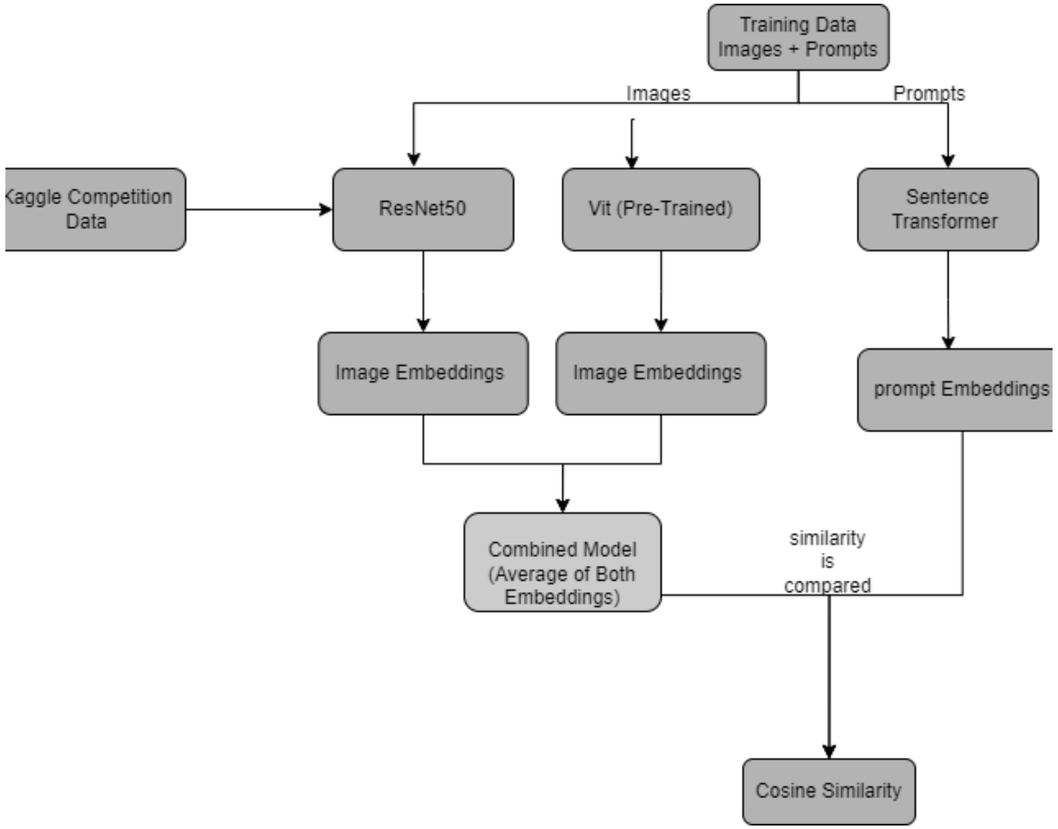
```
==========================================================
Layer (type:depth-idx)                   Param #
==========================================================
├─ResNet: 1-1                            --
|    └─Conv2d: 2-1                       9,408
|    └─BatchNorm2d: 2-2                  128
|    └─ReLU: 2-3                         --
|    └─MaxPool2d: 2-4                    --
|    └─Sequential: 2-5                   --
|    |    └─Bottleneck: 3-1              75,008
|    |    └─Bottleneck: 3-2              70,400
|    |    └─Bottleneck: 3-3              70,400
|    └─Sequential: 2-6                   --
|    |    └─Bottleneck: 3-4              379,392
|    |    └─Bottleneck: 3-5              280,064
|    |    └─Bottleneck: 3-6              280,064
|    |    └─Bottleneck: 3-7              280,064
|    └─Sequential: 2-7                   --
|    |    └─Bottleneck: 3-8              1,512,448
|    |    └─Bottleneck: 3-9              1,117,184
|    |    └─Bottleneck: 3-10             1,117,184
|    |    └─Bottleneck: 3-11             1,117,184
|    |    └─Bottleneck: 3-12             1,117,184
|    |    └─Bottleneck: 3-13             1,117,184
|    └─Sequential: 2-8                   --
|    |    └─Bottleneck: 3-14             6,039,552
|    |    └─Bottleneck: 3-15             4,462,592
|    |    └─Bottleneck: 3-16             4,462,592
|    └─AdaptiveAvgPool2d: 2-9            --
|    └─Sequential: 2-10                  --
|    |    └─Linear: 3-17                 1,049,088
|    |    └─BatchNorm1d: 3-18            1,024
|    |    └─ReLU: 3-19                   --
|    |    └─Dropout: 3-20                --
|    |    └─Linear: 3-21                 196,992
==========================================================
Total params: 24,755,136
Trainable params: 24,755,136
Non-trainable params: 0
----------------------------------------------------------
```

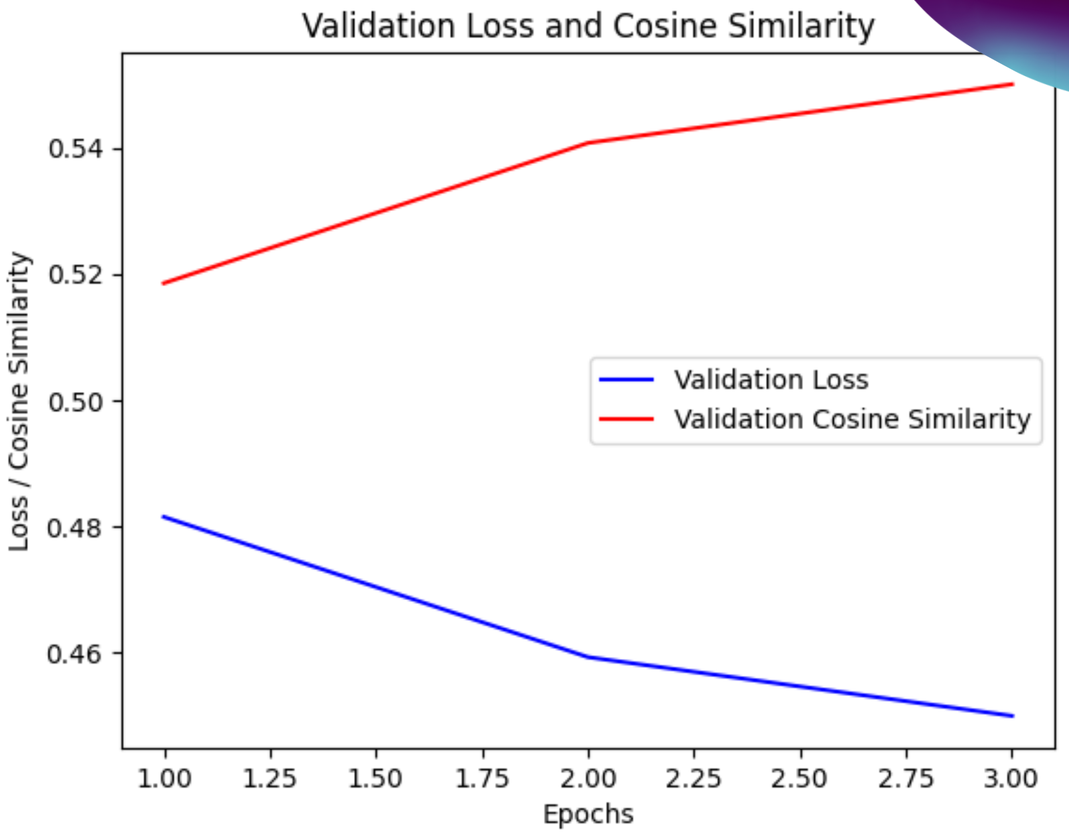




- Using skip connections, ResNet can learn deeper representations of images and produce more informative embeddings
- it can handle the vanishing gradient problem, finetuned on 154320 images.
- Adaptive average pooling layer,Batch Normalization and Drop out layers are introduced in final layers with ReLu activation.
- Resnet has stronger inductive bias towards local connections, it may not capture all the relevant information in the image, especially for complex image datasets.

Modeling(ResNet + Vit)





- **ResNet50 has a strong structural bias towards local connections between adjacent layers due to its residual blocks.**
- **ViT has a higher inductive bias towards the input structure.**
- **The attention mechanism used in visual transformers allows them to selectively focus on relevant image regions.**
- **ViT uses a multi-head self-attention mechanism, which allows it to focus on different parts of the input image and capture spatial relationships between them.**
- **Visual transformers are able to learn global dependencies between image regions, while ResNet-50 is limited by its local receptive field size**

Cosine Similarity

- Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that calculates the cosine of the angle between them, ranging from -1 (completely dissimilar) to 1 (completely similar).
- In our work, cosine similarity is used as a performance metric to evaluate the quality of the image embeddings generated by the model, measuring the similarity between the model's output and the ground truth text embeddings.
- Higher cosine similarity values indicate better model performance, as it means the generated embeddings are closer to the true embeddings, effectively representing the prompts.

Model Evaluation

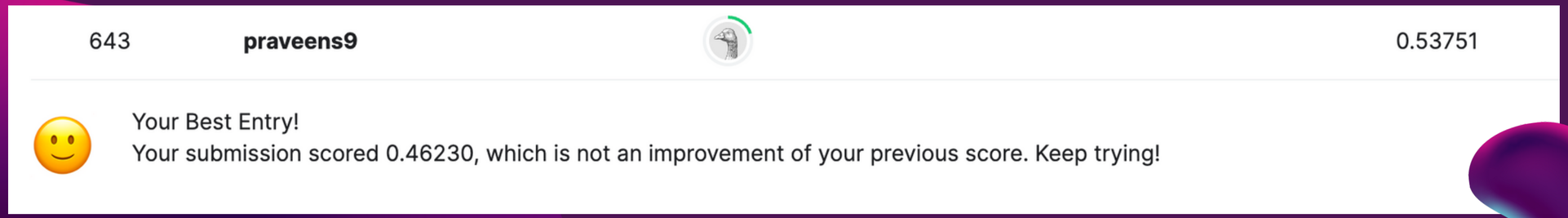
Original prompt:
ultrasaurus holding a black bean taco in the woods, near an identical cheneosaurus

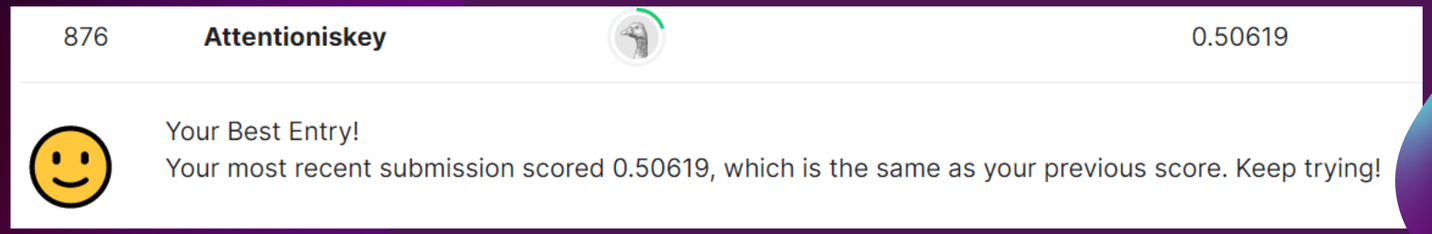
Generated prompt:
a cartoon dinosaur with a piece of cheese in its mouth, an illustration of, sumatraism, mmmmm, buttercup eating pizza, pastry lizard

- The fine-tuned model is then used to predict embeddings for the 7 test images provided by Kaggle.
- Generated 384 embeddings per image and in total 2688 embeddings were generated and submitted to the competition.

Results

- On comparing both the all the models, the ensemble of CLIP and ViT gave the highest score on Kaggle competition after successful submission
- Final embeddings = 0.25*(CLIP embeddings) + 0.75*(Custom ViT embeddings)





Conclusion

- **Objective**: We aimed to reverse the text-to-image generative process using the Stable Diffusion 2.0 model.
- **Approach**: Our ensemble model combined the CLIP Interrogator and a Custom Vision Transformer.
- **Performance**: We achieved a score of 0.53751, ranking #643 on the Kaggle leaderboard.
- **Key Insights**: Our work revealed insights into the reversibility of text-to-image models and the latent relationships between text prompts and images.
- This project highlights the potential of ensemble models and customized architectures to tackle complex tasks like reversing text-to-image generative processes.