Image Captioning & Story Generation

Group 19
Affan Bin Usman | Dhaval Nalin Shah | Eshita Sanjay Khandelwal | Kaushal Rathi | Viraj Thakkar

Contents

- Introduction
- Problem Description
- Methodology
- Results
- Conclusion & Future Work

Introduction

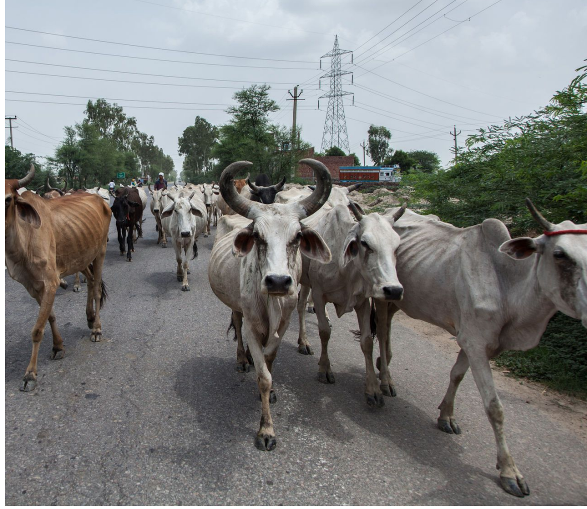
1. Image Captioning

Image captioning

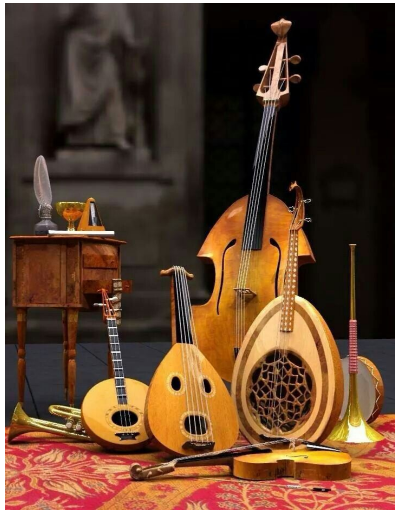
- Process enabling the computer to generate one or more sentences
- Describes the visual content of the image
- Deals with how humans and systems interpret information
- Combination of Computer Vision and NLP

Goals

- Creating meaningful sentences to describe the image
- Re-create the model for image captioning
  - Integrating best practices from various models
  - Fine tune parameters
- Commonly utilized datasets
  - Google Conceptual Captions Dataset
  - Microsoft COCO Dataset
  - Flickr 30k Dataset
  - Flickr 8k Dataset
- Model evaluation and comparisons (custom vs existing)



Original caption: a herd of cattle crowd the highway
Generated caption: a herd of cattle walking down a street



Original caption:
here are some of our instruments on the left side of the music room
Generated caption:
an old fashioned musical instrument sitting on top of a table

Introduction

1. Image Captioning
2. Story Generation
3. Image Generation (Story to images)

Story Generation

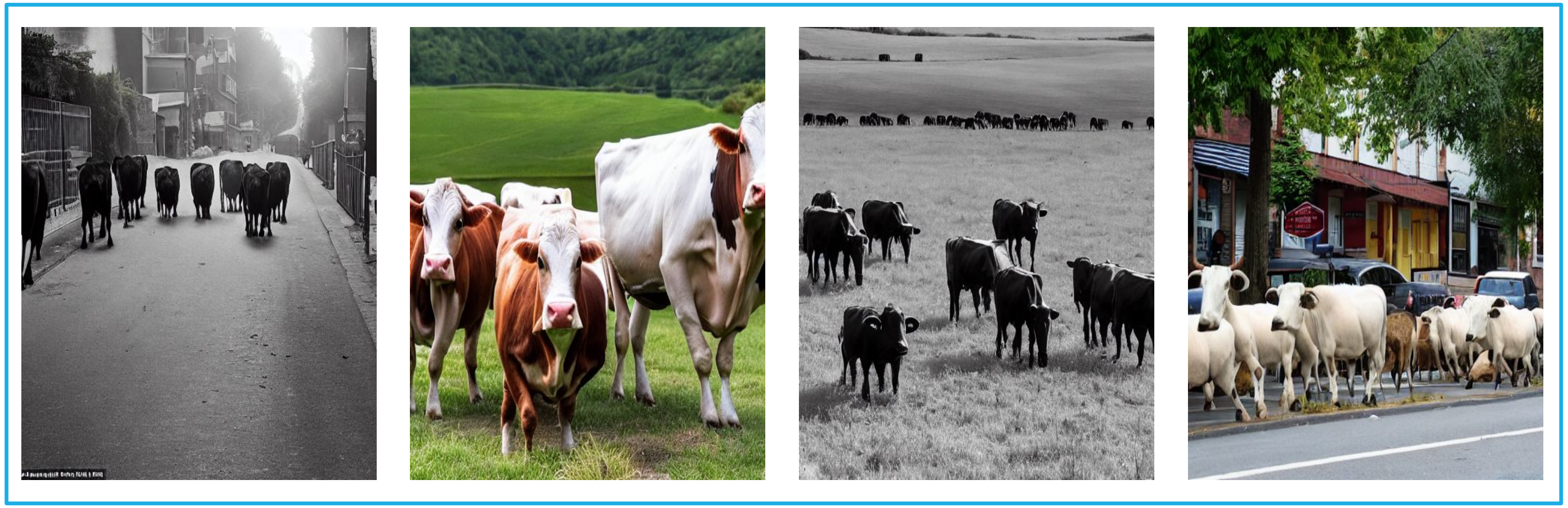
- Creating a story from the generated caption
- The story is comprised of a few sentences, forming a paragraph
- Implemented using Playground API by OpenAI

Image Generation

- Transforms a text statement to an image
- Latent diffusion model by StabilityAI
- Utilizes text embeddings for guided image generation

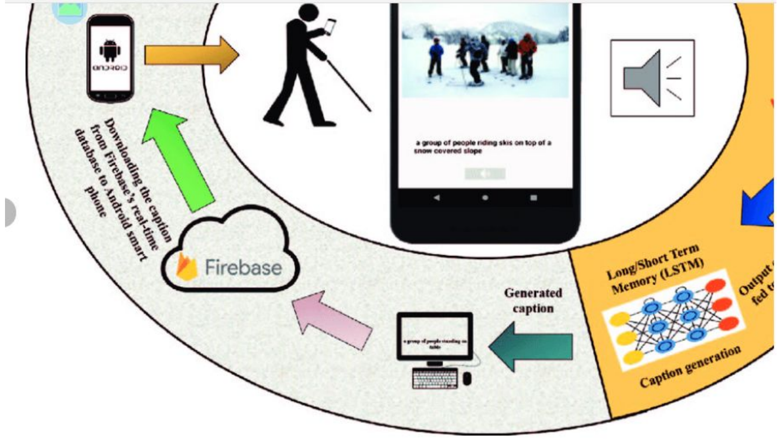
Caption:
A herd of cattle walking down a street

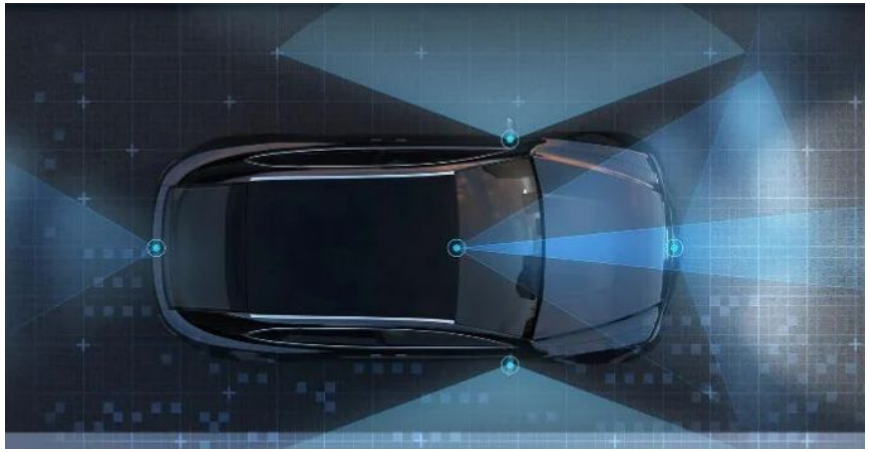
Story generated:
The herd of cattle slowly made their way down the street, their hooves clopping against the pavement. The cows lowed softly to each other as they walked, their tails swishing back and forth. The farmers followed behind the herd, making sure that none of the cows strayed too far from the group. The herd continued down the street, passing by houses and shops.



Problem Description

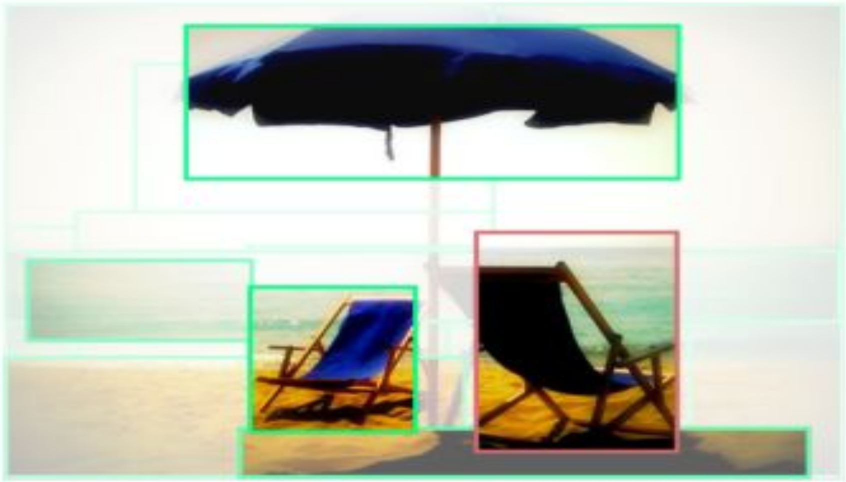
1. As seen for the images in introduction, it’s very easy for us humans to caption the image, to have a glance and describe the image
2. Difficult for the system to understand the image and caption as humans
3. It is used at multiple real world applications like:
   - ○ Aid for the blind
   - ○ Self driving cars
   - ○ Google image search
   - ○ Web development
   - ○ CCTVs


ANDROID
a group of people riding skis on top of a snow covered slope
Downloading the caption from Firebase's real-time database to Android smart phone
Firebase
a group of people standing on table
Generated caption
Long/Short Term Memory (LSTM)
Caption generation
Output fed t



Problem Description

1. The primary goal is generating captions for our images. We want to develop a process which requires object recognition and one which develops a caption related to that object.
2. Add more personalized context to the image, we want to achieve something shown in the figures below.
3. Image captioning systems are example of big data systems as they focus on the volume aspect of the data. For example MS COCO, Flickr30K

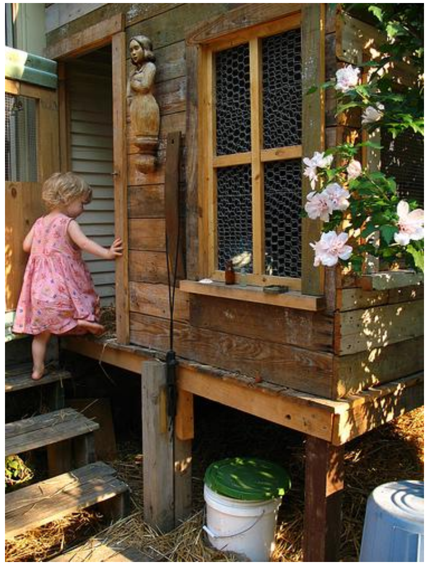


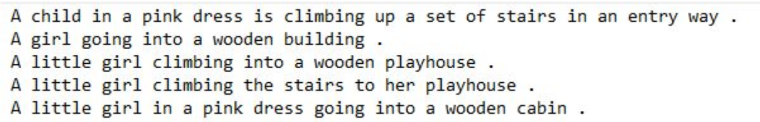
Generated Caption: *two beach chairs under an umbrella on the beach*

Understanding the Datasets

Apart from the massive size difference (Fickr8k ~8000 images & Google CC ~3.3M). There is a stark difference in the type as well. Flickr8k is a synthetic database and GCC is a real world database!

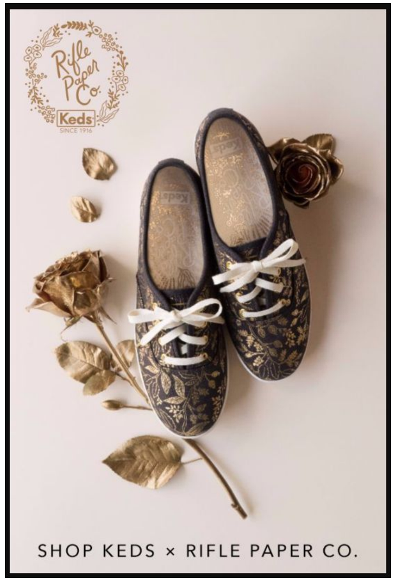
Flickr8k





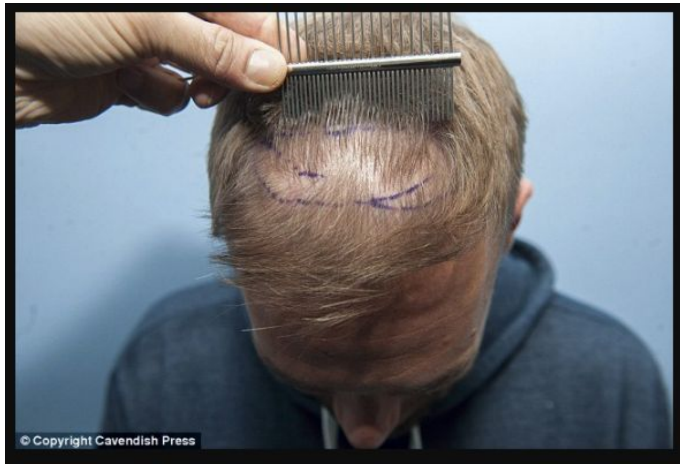
```
A child in a pink dress is climbing up a set of stairs in an entry way .
A girl going into a wooden building .
A little girl climbing into a wooden playhouse .
A little girl climbing the stairs to her playhouse .
A little girl in a pink dress going into a wooden cabin .
```

Google Conceptual Captions



A wearable work of art.

94ca81c1800301ac9a254ea1fa43
12f0.jpg (600×900) (pinimg.com)



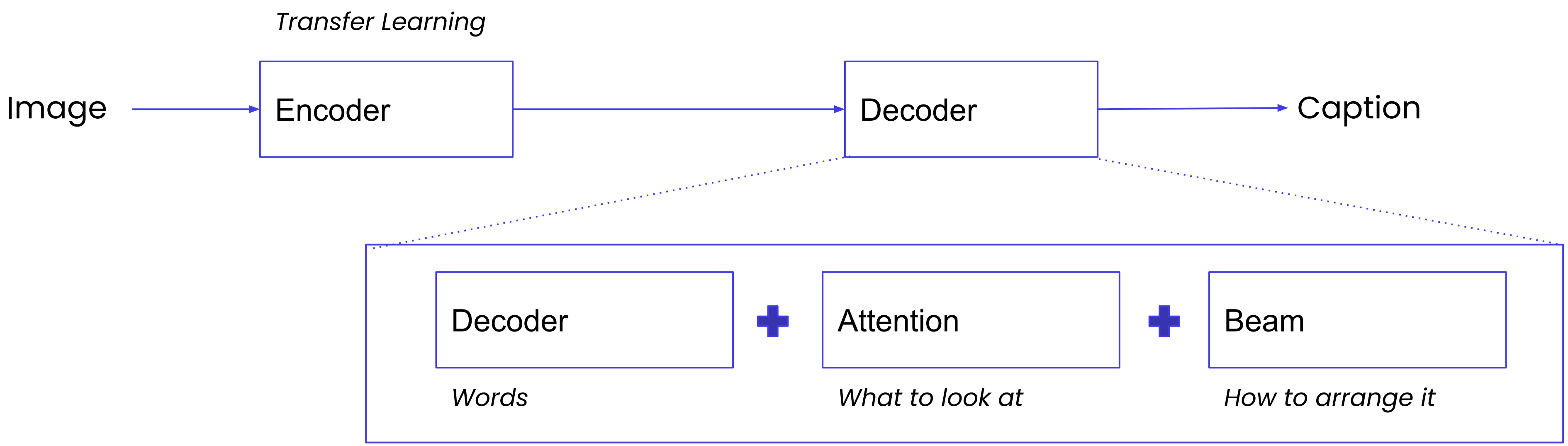
hair today : the former model has bolstered his thinning locks with a £ 6,000 operation

article-2539080-1AA61BF700000578-611_634x422.jpg (634×422)
(dailymail.co.uk)

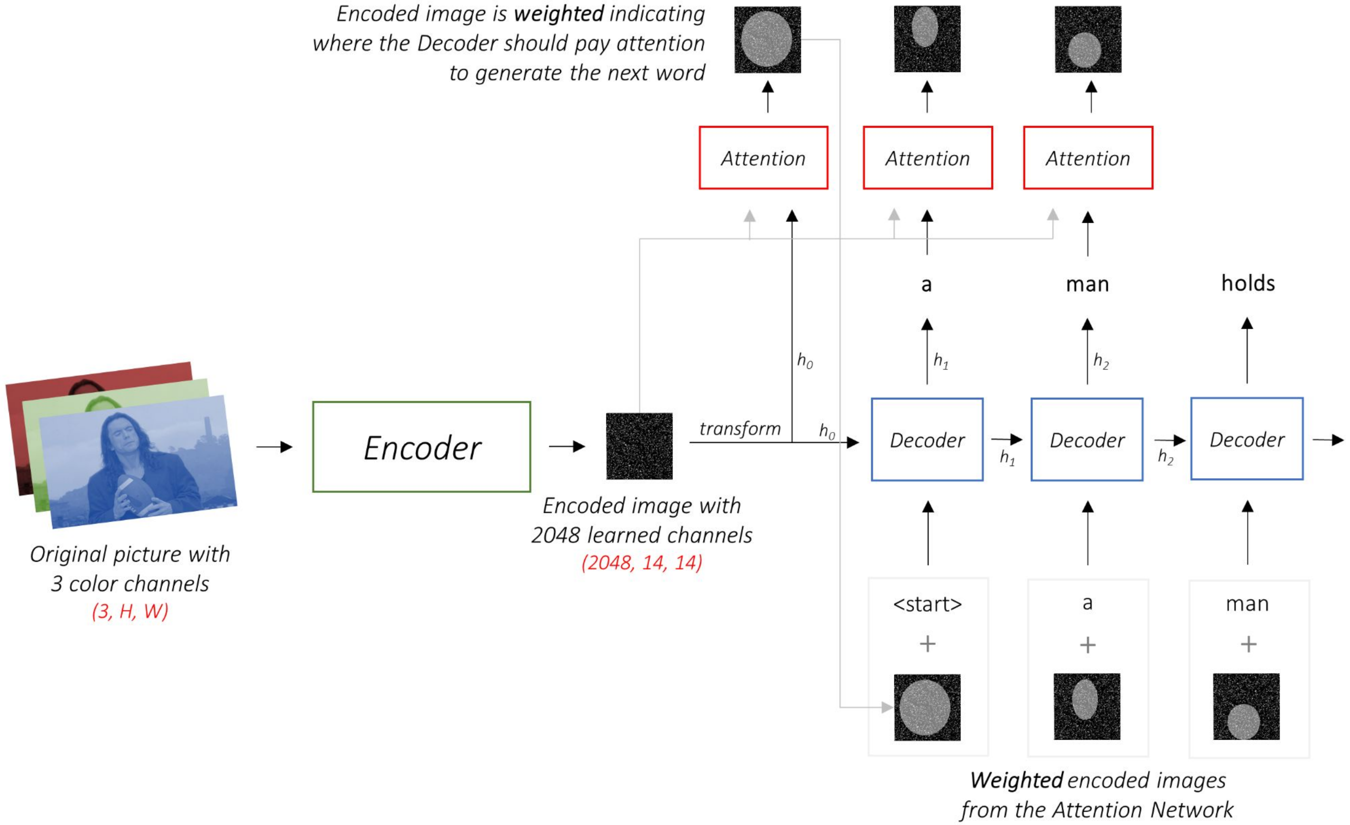
Methodology - Some Terminology

- Transfer Learning:
  - To take a pre-trained model, discard the top layer(s) and repurpose it for your use case!
- Encoder:
  - Primary purpose is to convert data into the required format.
  - In our case, it is an image converted to smaller image with more dimensions
- Decoder:
  - Primary purpose is to convert data formatted by encoder to what we want
  - In our case, it is converting a multi-dimensional image from encoder into a sensible statement
- Attention Networks:
  - They will help focus on certain parts of the image that make the most sense
  - eg. Image of a person holding a ball, it will focus on person, hands and ball
- Beam Search:
  - It helps order the words in a sensible sequence

Methodology - The Model



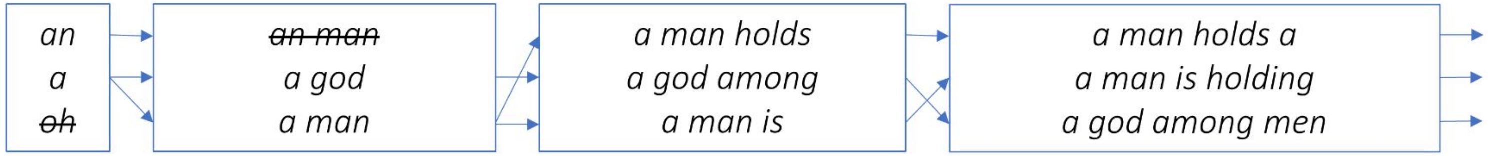
Methodology - Attention



Methodology - Beam search

Beam Search with k = 3

Choose top 3 sequences at each decode step.
Some sequences fail early.
Choose the sequence with the highest score after all 3 chains complete.



an
a
~~*oh*~~

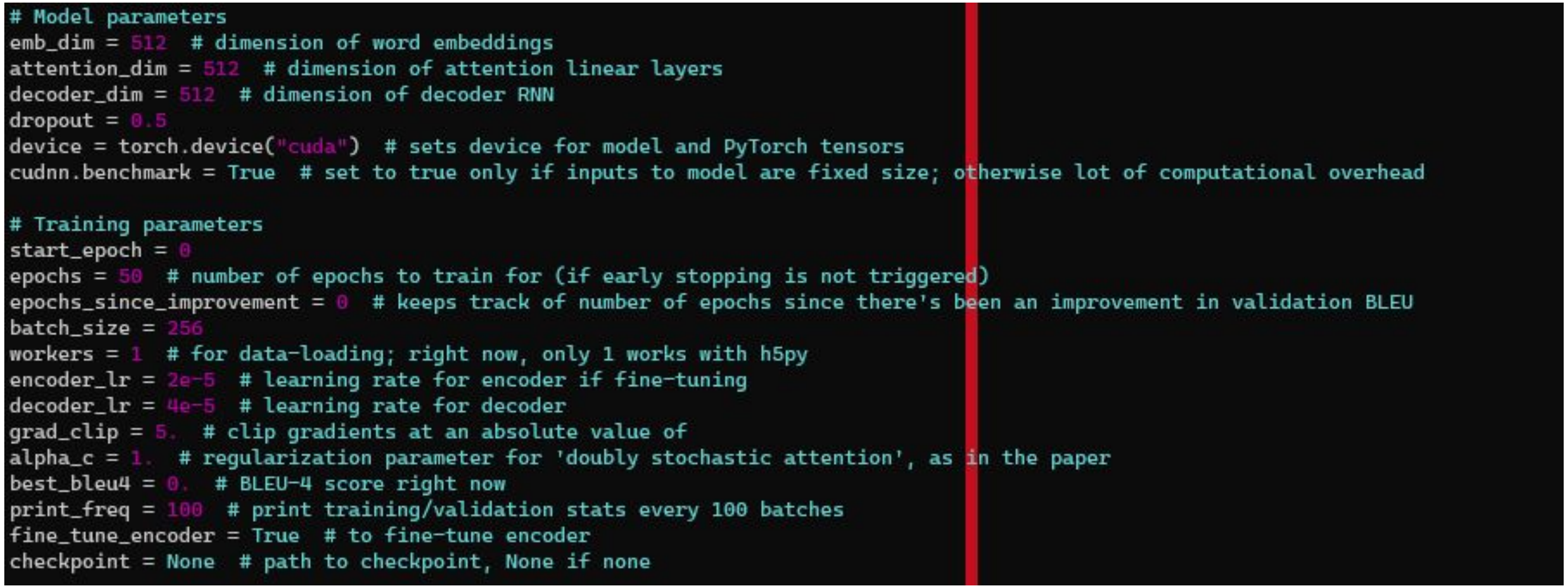
~~*an man*~~
a god
a man

a man holds
a god among
a man is

a man holds a
a man is holding
a god among men

Methodology - Fine Tuning

Multiple fine tuning attempts (6 in total) were made: Changes attempted in: epochs, batch size, learning rates (both encoder and decoder)



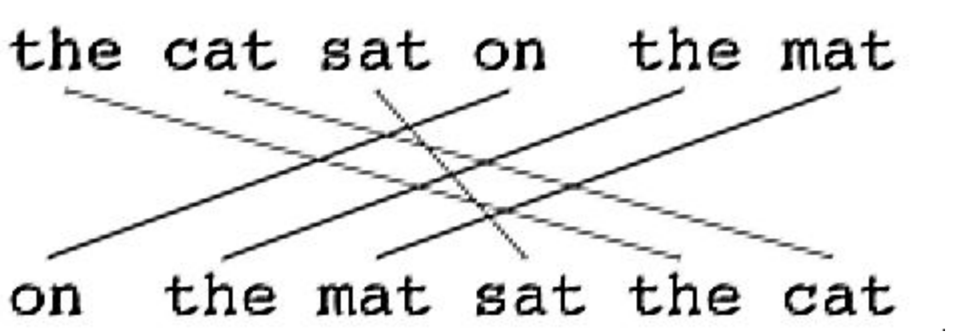
```
# Model parameters
emb_dim = 512  # dimension of word embeddings
attention_dim = 512  # dimension of attention linear layers
decoder_dim = 512  # dimension of decoder RNN
dropout = 0.5
device = torch.device("cuda")  # sets device for model and PyTorch tensors
cudnn.benchmark = True  # set to true only if inputs to model are fixed size; otherwise lot of computational overhead

# Training parameters
start_epoch = 0
epochs = 50  # number of epochs to train for (if early stopping is not triggered)
epochs_since_improvement = 0  # keeps track of number of epochs since there's been an improvement in validation BLEU
batch_size = 256
workers = 1  # for data-loading; right now, only 1 works with h5py
encoder_lr = 2e-5  # learning rate for encoder if fine-tuning
decoder_lr = 4e-5  # learning rate for decoder
grad_clip = 5.  # clip gradients at an absolute value of
alpha_c = 1.  # regularization parameter for 'doubly stochastic attention', as in the paper
best_bleu4 = 0.  # BLEU-4 score right now
print_freq = 100  # print training/validation stats every 100 batches
fine_tune_encoder = True  # to fine-tune encoder
checkpoint = None  # path to checkpoint, None if none
```

Evaluation

Calculate the similarity between reference and predicted sentence by:

- Capturing common words
- Getting correct alignment
- Capturing semantics

Reference sentence



Predicted sentence

Functioning of Metrics

- **Convert sentences to words:**
  Words are referred as: unigram (1 word), bigram (2 words), n-grams (n words)
  ⟶ Solves the alignment issue.

- **Calculate precision**
  precision=number of a n-gram occurs in reference and hypothesis/number of n-grams occurs in the hypothesis/candidate translation.
  ⟶ number of words matched with respect to hypothesis.

- **Calculate recall**
  recall=number of n-gram occurs in reference and hypothesis/number of n-gram already in reference.
  ⟶ number of words matched with respect to reference.

- **Calculate F1-score**
  F1-score= harmonic mean of recall and precision.

Results

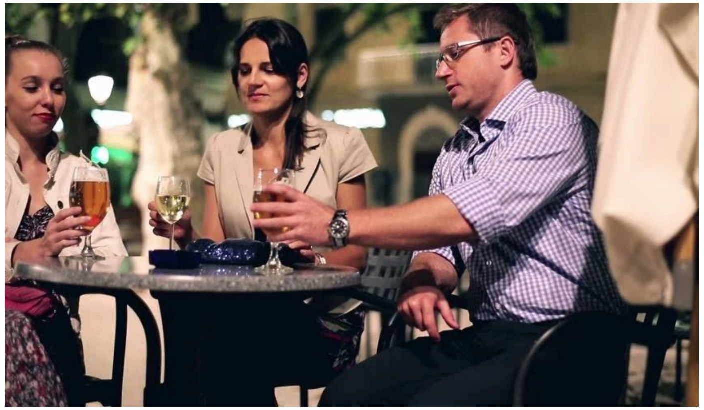
Test data set used was Google Conceptual Captions (GCC)

Metrics	**Vit-gpt2** **Trained on Coco dataset**	**Our model** **Trained of Flickr 8k dataset**
SacreBlue score (P,Brevity penalty)	18.58 %	33.44 %
Rogue metric (P,R,F1)	Rouge-1: 14.75 % Rouge-2: 36.12 % Rouge-L: 14.1 %	Rouge-1: 20.004 % Rouge-2: 20.004 % Rouge-L: 20 %
Meteor score (P,R,F1)	3.5 %	3.6 %
Cider metric (cosine similarity)	2.88 %	1.25%

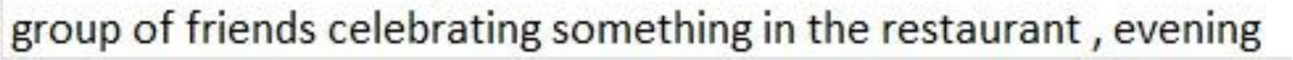
For reference:

- Blue score >60% is considered quality often better than human and highest performing model trained on Coco dataset itself has 46.5% score!
- GCC highest performing model itself has Rouge-L 27.79% !

Conclusion and Future Work

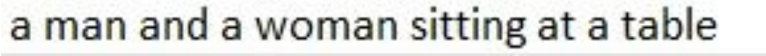


GCC sample test image



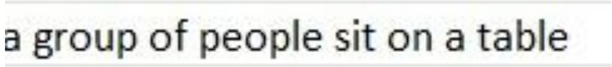
group of friends celebrating something in the restaurant , evening

GCC test/reference sentence



a man and a woman sitting at a table

Vit-gpt2 model predicted sentence



a group of people sit on a table

Our model predicted sentence

Conclusion: Since there is cumulative excessive variation we still need human judgement as the last step for evaluation!

Future work:

- Preserving subjects: What if we preserve the data from the attention model and pass it onto image generation, can we avoid cases where a cat becomes a human in the next image?
- Video generation: Imagine input of an image and getting a smooth video out of it!

References

Understanding and Implementation (Image Captioning)

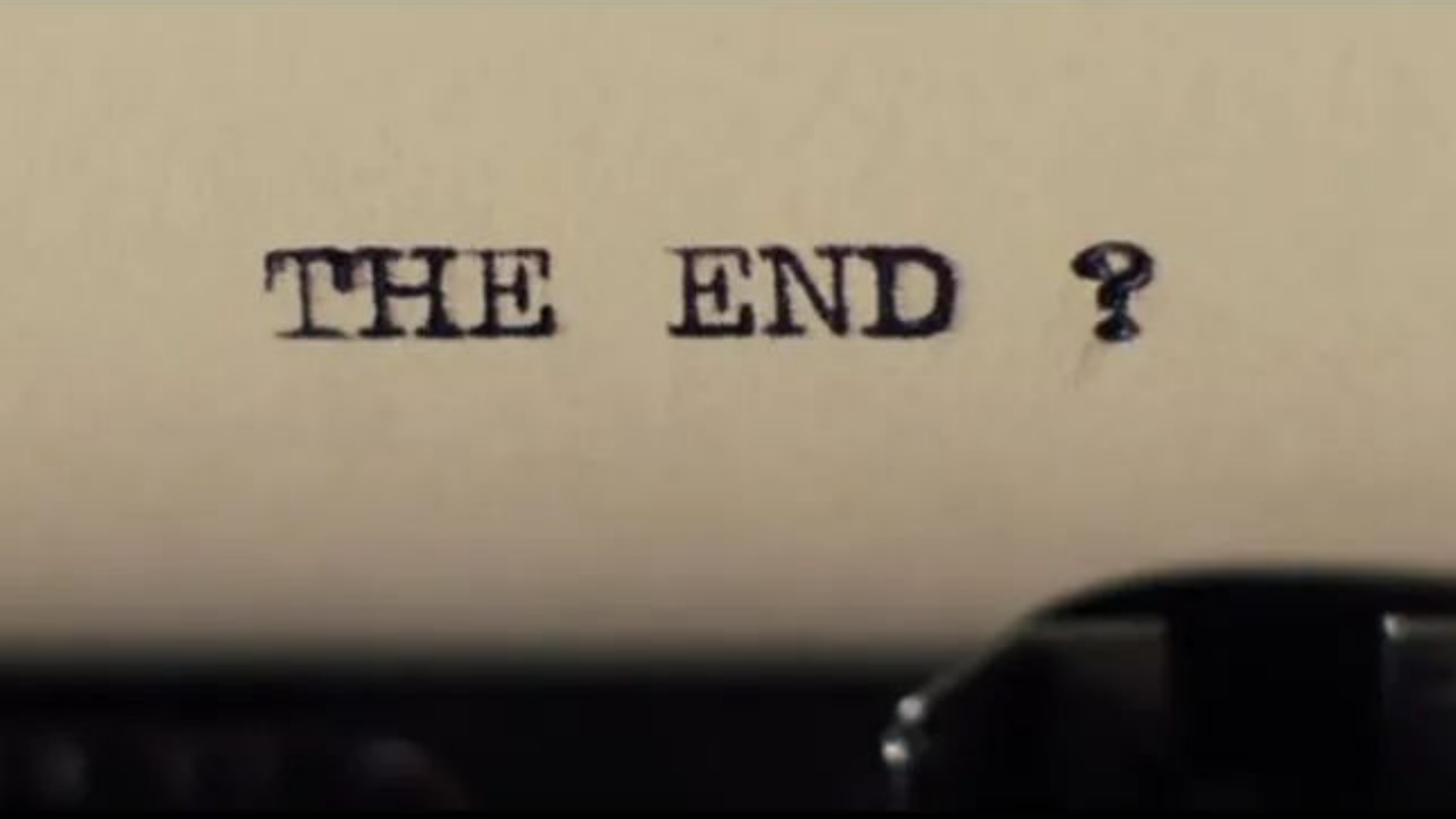
1. ydshieh/vit-gpt2-coco-en-ckpts · Hugging Face
2. GitHub - sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning: Show, Attend, and Tell
3. Understanding and visualizing ResNets | by Pablo Ruiz | Towards Data Science
4. Understanding Encoders-Decoders with Attention Based Mechanism | DataX Journal
5. How Does Attention Work in Encoder-Decoder Recurrent Neural Networks - MachineLearningMastery.com

Understanding and Implementation (Story Generation & Image Generation)

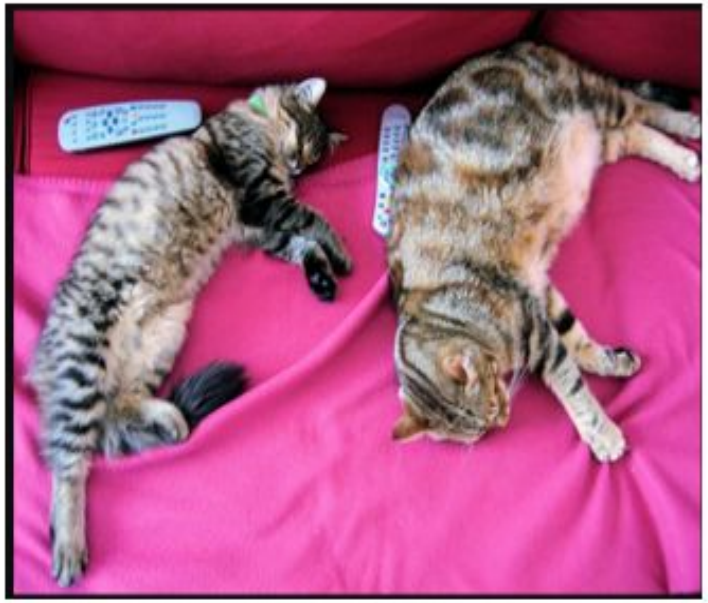
1. Playground - OpenAI API
2. Stable Diffusion - Dreambooth

Evaluation

1. CIDEr: Consensus-based Image Description Evaluation
2. Evaluation Metrics for High-Correlation with Human Rankings of Machine Translation Output
3. Cross-validating Image Description Datasets and Evaluation Metrics
4. Understanding BLUE score
5. Meteor Universal: Language Specific Translation Evaluation for Any Target Language
6. ROUGE: A Package for Automatic Evaluation of Summaries



THE END ?

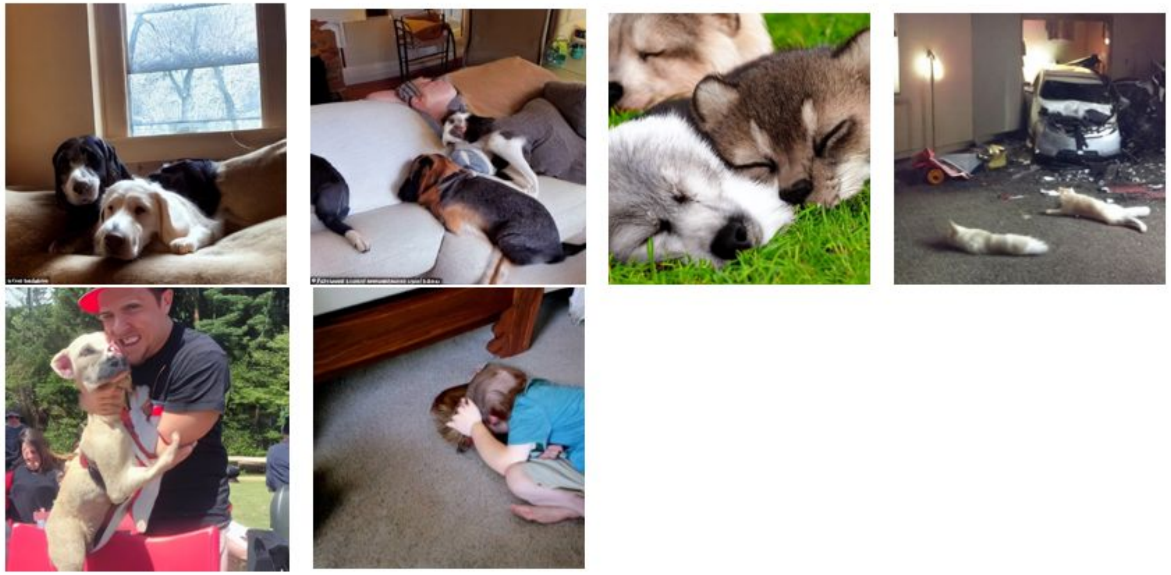


['a cat laying on top of a couch next to another cat']

['Luna and her sister, Lily, loved to lay on the couch together and watch the world go by outside the window', ' They loved each other dearly and spent most of their time together', ' On this particular day, they were both content to lay on the couch, with Luna on top of the cushions and Lily next to her', ' They dozed off and were soon fast asleep',

Suddenly, there was a loud crash outside and both cats woke with a start', ' They jumped off the couch and ran to hide under the bed', ' They were both terrified and shaking', ' After a few minutes, they cautiously peeked out from under the bed to see what had caused the noise',

It was just a bird that had flown into the window', ' Luna and Lily both breathed a sigh of relief and slowly came out from under the bed', " They both jumped back up on the couch and snuggled close together, content to be safe and sound in each other's company"]



Some Results

(chosen with utmost care)

Any Questions??