# Image Captioning Using ResNet-50 and LSTMs

Alexander Holland
University of Victoria
Victoria, Canada
alexanderwholland@uvic.ca

Bhavanvir Rai
University of Victoria
Victoria, Canada
brai@uvic.ca

Kevin, Mathew
University of Victoria
Victoria, Canada
kevingmathew@uvic.ca

***Abstract*—This report outlines how neural networks could be used to predict the captions of images based on its features. The dataset for this project, sourced from Kaggle, from Flickr consists of 8,000 images and 40,000 complimentary captions [1]. After the image pre-processing step was complete, a long short-term memory network (LSTM) was employed and trained on the data; more specifically, a residual neural network (ResNet). The results revealed that neural networks are more than capable of predicting such targets, though their efficacy is highly dependent on model architecture and overall dataset quality, where to produce highly accurate predictions it must be sufficiently large, diverse and optimized enough for the task.**

## I. Introduction

Image captioning models aim to bridge the gap between visual content and natural language by generating descriptive captions for images. These models combine advancements in computer vision and natural language processing to automatically analyze the visual features of an image and generate coherent and contextually relevant textual descriptions. Typically, image captioning models employ deep learning architectures, such as convolutional neural networks (CNNs) for visual feature extraction and recurrent neural networks (RNNs), such as LSTM networks, for generating sequential captions. More recently, transformer-based models, such as the vision-transformer (ViT) models, have gained prominence due to their ability to capture global image context and semantic relationships. The development of these image captioning models has revolutionized various domains, including accessibility, content indexing, and contextual understanding, opening up new possibilities for leveraging visual content in a textual form.

**Task.** The primary objective of this project is to develop an image captioning system that automatically generates descriptive and contextually relevant captions for images. The task at hand involves leveraging the synergies between computer vision and natural language processing to bridge the gap between visual content and textual comprehension. Through the utilization of deep learning techniques, we aim to train a model capable of accurately associating visual features with meaningful textual descriptions.

To achieve our goal, we will collect a comprehensive dataset comprising a diverse range of images along with corresponding captions. This dataset will serve as the foundation for training and evaluating our image captioning model. Throughout the project, we will experiment with various techniques and methodologies, fine-tuning the model's parameters to optimize its performance. We will evaluate the generated captions using established metrics, such as BLEU, to measure the quality and relevance of the model's output. Our ultimate aim is to develop an image captioning system that not only accurately describes the visual content of an image but also captures the contextual nuances and relationships depicted within the image.

## II. Questions

1) What kind of model can we use to effectively tackle this problem?
2) What kind of dataset could we use to train the model?
3) How do we measure the effectiveness of the model?
4) What are the practical applications of this project?
5) What are some potential directions for improvement?
6) What are the characteristics of the dataset? (e.g., number of images, number of captions per image)?
7) Can I leverage pre-trained models (e.g., pre-trained on ImageNet, COCO, etc.) for image representation?
8) What pre-processing steps are necessary for the images and captions before feeding them into the model?
9) Are there any legal or ethical considerations in obtaining image-caption pairs?
10) What measure can be taken to improve the effectiveness and quality of the model?

## III. Analysis

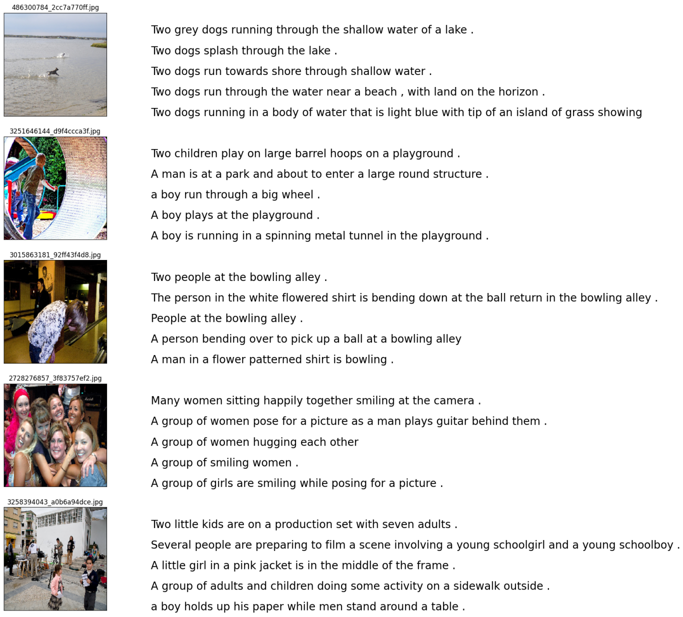
Data analysis before creating an image captioning model is crucial. It provides insights into the dataset's characteristics, such as image distribution, caption lengths, and vocabulary. This information informs model design, addressing specific challenges and biases. By understanding the data, we can pre-process it appropriately and develop more accurate and relevant captioning models.

### *A. Exploratory Data Analysis*

**Dataset.** The Flickr8k dataset consists of 8,000 images paired with five captions each, resulting in a total of 40,000 captions. The images cover a wide range of subjects and scenarios, but no specific categories or tags are available. Analyzing the captions, we find that they vary in length, with an average of approximately 10 to 15 words. The minimum caption length is three words, while the maximum reaches around 20 words [1].

The vocabulary in the dataset is rich and diverse, with an estimated size of 8,000 to 10,000 unique words. Common words such as ”a,” ”the,” ”on,” ”in,” and ”and” indicate the presence of generic captions describing basic elements of the images, i.e. stopwords with no inherent value. To assess the diversity and quality of the dataset, a random sample of images and their associated captions was examined, providing insights into the various image-caption pairs.

**Image Sampling.** Sampling images with their respective captions in the Flickr8k dataset is crucial for evaluating caption quality, assessing alignment between visuals and text, identifying challenges, and understanding distribution patterns. By randomly selecting images and examining their associated captions, the quality and diversity of the dataset can be assessed, providing insights into descriptive power and creativity. Aligning images with captions enables the evaluation of caption accuracy and relevance, ensuring meaningful descriptions. Sampling also helps identify issues such as inconsistencies or repetition, guiding improvement strategies. Furthermore, understanding distribution patterns aids in modeling decisions, ensuring the system handles various image types effectively. In Figure 1 we can see the sample images and their captions.


486300784_2cc7a770ff.jpg
3251646144_d9f4ccca3f.jpg
3015863181_92ff43f4d8.jpg
2728276857_3f83757ef2.jpg
3258394043_a0b6a94dce.jpg

Two grey dogs running through the shallow water of a lake .
Two dogs splash through the lake .
Two dogs run towards shore through shallow water .
Two dogs run through the water near a beach , with land on the horizon .
Two dogs running in a body of water that is light blue with tip of an island of grass showing

Two children play on large barrel hoops on a playground .
A man is at a park and about to enter a large round structure .
a boy run through a big wheel .
A boy plays at the playground .
A boy is running in a spinning metal tunnel in the playground .

Two people at the bowling alley .
The person in the white flowered shirt is bending down at the ball return in the bowling alley .
People at the bowling alley .
A person bending over to pick up a ball at a bowling alley
A man in a flower patterned shirt is bowling .

Many women sitting happily together smiling at the camera .
A group of women pose for a picture as a man plays guitar behind them .
A group of women hugging each other
A group of smiling women .
A group of girls are smiling while posing for a picture .

Two little kids are on a production set with seven adults .
Several people are preparing to film a scene involving a young schoolgirl and a young schoolboy .
A little girl in a pink jacket is in the middle of the frame .
A group of adults and children doing some activity on a sidewalk outside .
a boy holds up his paper while men stand around a table .

Fig. 1. Sample images with their 5 respective captions.

**Word frequency.** By examining the distribution of word occurrences in the caption dataset, we can gain valuable insights into the common vocabulary used. This analysis helps us identify frequently occurring words, such as articles and prepositions, which can guide pre-processing steps like stopword removal. Additionally, it allows us to identify important and domain-specific terms that are relevant to the image captioning task. Understanding word frequency enables us to make informed decisions regarding vocabulary size, word embeddings, and language modeling techniques, ultimately improving the accuracy and coherence of generated captions. We observed in Figure 2 that the most frequent words were common parts of speech articles such as ”a”, ”in”, ”the”, ”and”. Additionally these common words provide us with potential stop words when pre-processing the data. Likewise in Figure 4 we can see some other common words using a word cloud where we can see common subjects of each image, such as ”woman”, ”man”, ”dog”, ”beach”, etc. On the other hand, observing the least frequent words in Figure 3 provides little insight into word frequencies, but does show that captions do include very image specific description words.

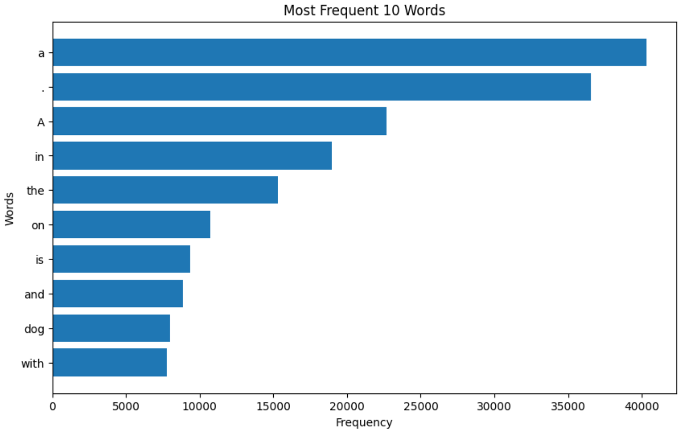


Fig. 2. Bar graph of the 10 most frequent words.

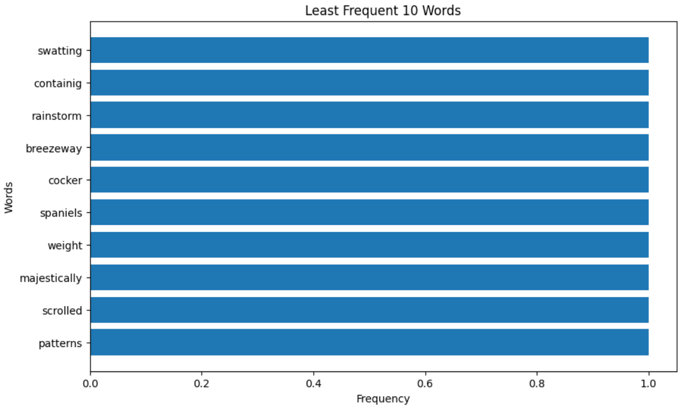


Fig. 3. Bar graph of the 10 least frequent words.

**Caption length.** Analyzing the length of captions can offer insights into determining an optimal balance between conciseness and informativeness. By avoiding excessive specificity, captions can effectively capture the essence of an image while remaining concise. In Figure 5 we can see that the histogram exhibits a symmetric distribution with a

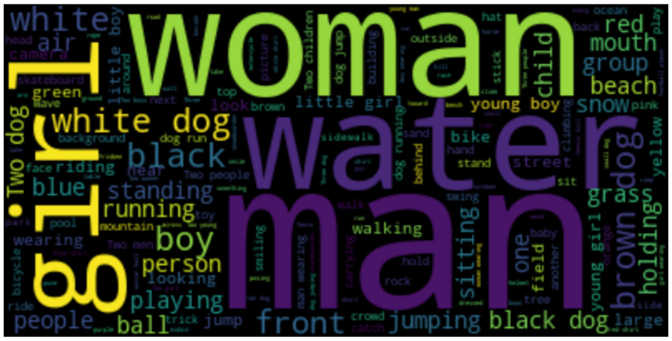


Fig. 4. Word Cloud of caption texts.

prominent peak centered around a caption length of 10 words.

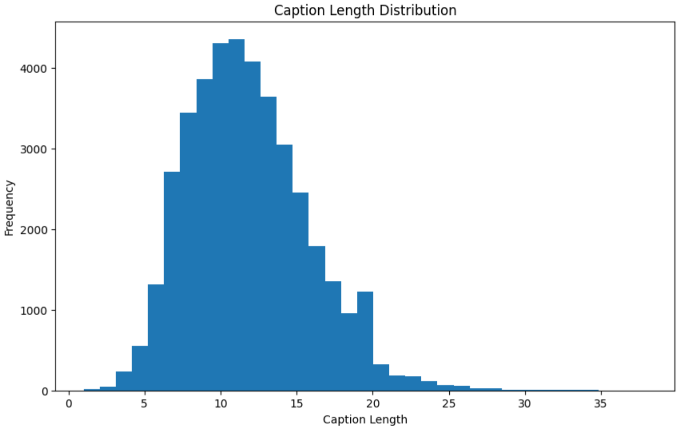


Fig. 5. Histogram of the caption lengths in the Flickr8k dataset.

**Part-of-speech tagging.** the Natural Language Toolkit (NLTK), a widely-used Python library, offers a range of functionalities for text processing, including part-of-speech (POS) tagging. By leveraging NLTK's POS tagging capabilities, we can assign grammatical categories, such as nouns, verbs, and adjectives, to each word in each caption. This process is enables us to comprehend the syntactic structure of the caption and use it to generate more accurate and contextually appropriate descriptions for associated images. In Table I we can see the POS tags with their respective meaning and the frequency of these tags used in the entire dataset in Figure 6 [3].

### B. Model Selection

**Model variations.** We had the choice between a multitude of different pre-trained models for the sole purpose of feature extraction from our set of training images. However, we developed a shortlist based on the following criteria:

1) **Architecture:** The model's architecture should be well-suited for image feature extraction. We looked for models that have demonstrated strong performance in visual recognition tasks.

TABLE I
NLTK POS Tags

| Tag | Meaning |
|---|---|
| NN | noun, singular (cat, tree) |
| VBZ | verb, present tense with 3rd person singular (bases) |
| VBG | verb gerund (judging) |
| NNS | noun plural (desks) |
| JJ | adjective (large) |
| VBP | verb, present tense not 3rd person singular (wrap) |
| VBN | verb past participle (reunified) |
| NNP | proper noun, singular (sarah) |
| VB | verb (ask) |
| VBD | verb past tense (pleaded) |
| JJR | adjective, comparative (larger) |
| JJS | adjective, superlative (largest) |
| NNPS | proper noun, plural (indians or americans) |

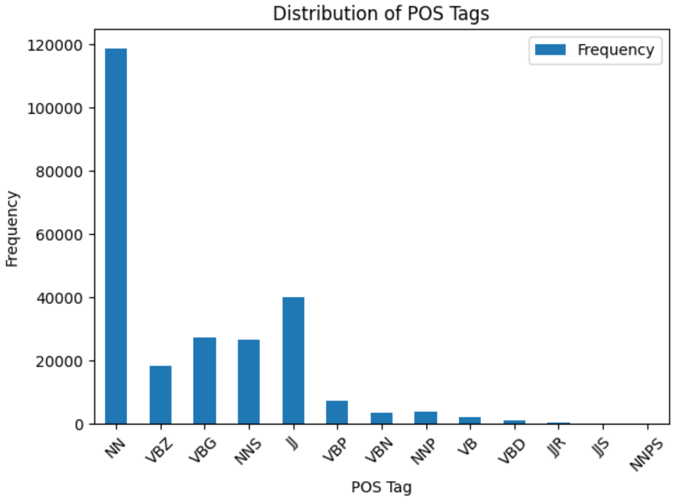


Fig. 6. Bar graph of the POS tag distributions.

2) **Pre-training:** It was important for the model to have been pre-trained on large-scale datasets such as ImageNet or COCO. Pre-training allows the model to learn generic visual representations, which can be fine-tuned for our specific image captioning task.
3) **Transfer learning:** We focused on models that support transfer learning. Transfer learning enables us to leverage the knowledge gained from pre-training on large datasets and apply it to our own task, improving efficiency and performance.
4) **Availability:** We considered models that are readily available and accessible through popular deep learning frameworks like TensorFlow or PyTorch. This ensured ease of implementation and compatibility with our existing infrastructure.

However, availability emerged as a significant constraint during our model selection process. Implementing a state-of-the-art model from an obscure technical paper was not feasible within our time frame. Consequently, we narrowed down our choices to well-known and widely used CNNs as encoders for image captioning, including VGG, ResNet, Xception, Inception, and DenseNet. Specifically, we tested VGG16, ResNet-50, ResNet-152, and Xception, evaluating the quality of the generated captions. Our results aligned with the findings in the 2021 paper by Alam et al. [2], where ResNet outperformed the other models. Considering both our experimental results and the literature, we decided to adopt ResNet-50 as our feature extraction model. While ResNet-50 is shallower than ResNet-152, we believed it had better potential for generalization in our transfer learning application. We considered its balance between model depth and computational efficiency to be advantageous for our specific needs. By selecting ResNet-50 as our feature extraction model, we aimed to leverage its strengths in capturing visual features and facilitating transfer learning to enhance the quality of our image captioning system.

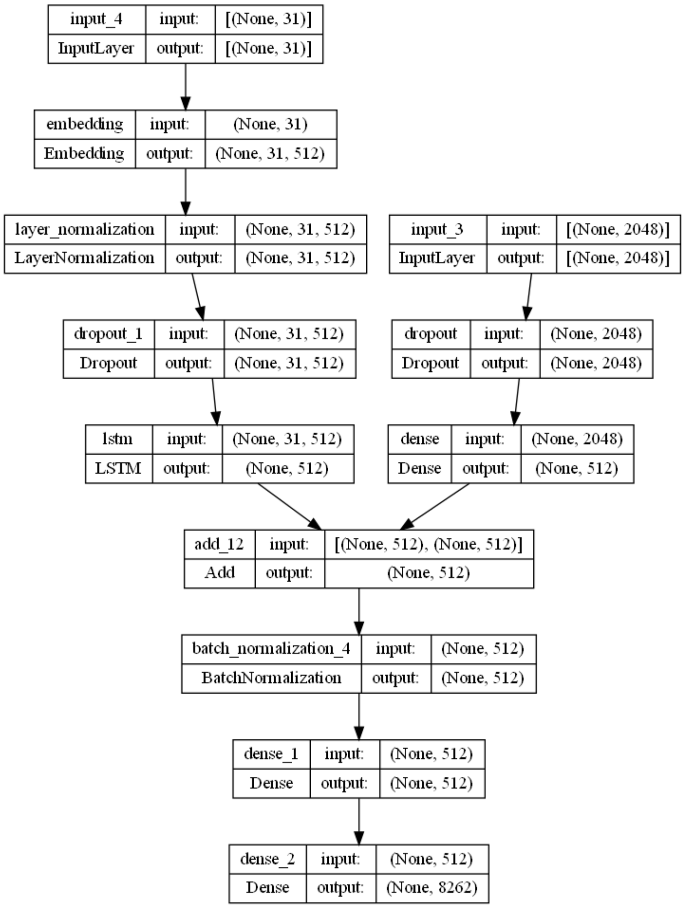
### *C. Model Training*

Once we selected the ResNet-50 model as our feature extraction model, we proceeded to train the image captioning system by combining the extracted image features with a language model component based on LSTM networks.

*1) Dataset Preparation:* We prepared our training dataset by pairing images with corresponding captions. Each image, reshaped to be of size (224, 224, 3) as is optimal for ResNet architectures, was passed through the ResNet-50 model to extract a fixed-length feature vector. The captions were preprocessed by first removing non-alphanumeric characters, extra spaces, and single characters, before being tokenized into words and representing them numerically. We also decided to limit our over 8200 word vocabulary to only the top 5000 words as it was observed to offer better performance for image captioning; we think this is in part to the smaller total corpus being easier to handle by our encoder-decoder model structure.

*2) Feature Engineering:* We prepared our training data set by pairing images with corresponding captions. As mentioned before we reduced the word vocabulary to 5000 words to only consider the most important words. We also converted captions to lowercase, removed non-alphabetic characters, removed extra spaces, removed single-character words, and added $< start >$ and $< end >$ anchors to mark the beginning and end of each caption. These techniques helped improve the quality and consistency of the caption text by removing irrelevant characters, reducing redundancy, and enhancing the overall readability. The prepossessed captions can then be used as input for our models.

*3) Architecture Design:* The architecture of our image captioning model consisted of two main components: the encoder and the decoder. The encoder utilized the pre-trained ResNet-50 model as the feature extractor. We froze the weights of the ResNet-50 layers to preserve the learned visual representations and prevent them from being updated during training. The output of the ResNet-50 model was fed into a fully connected layer to reduce the dimensionality of the features and capture high-level semantic information. For the decoder, we used an LSTM network to generate the captions based on the image features. The LSTM was designed to learn the sequential dependencies between words and generate captions word by word. The LSTM network was initialized with an embedding layer to convert the numerical representation of words into continuous vectors. We used Layer Normalization in our encoder Layer to help standardize and normalize the dataset. This helps avoid problems such as exploding gradients and also reduces overall training time. We also used batch normalization [4] in the Decoder Layer which helps normalize inputs throughout the batches. This becomes important as we progress through layers in our model and the input needs to be normalized again.



input_4 | input: | [(None, 31)]
InputLayer | output: | [(None, 31)]
embedding | input: | (None, 31)
Embedding | output: | (None, 31, 512)
layer_normalization | input: | (None, 31, 512)
LayerNormalization | output: | (None, 31, 512)
input_3 | input: | [(None, 2048)]
InputLayer | output: | [(None, 2048)]
dropout_1 | input: | (None, 31, 512)
Dropout | output: | (None, 31, 512)
dropout | input: | (None, 2048)
Dropout | output: | (None, 2048)
lstm | input: | (None, 31, 512)
LSTM | output: | (None, 512)
dense | input: | (None, 2048)
Dense | output: | (None, 512)
add_12 | input: | [(None, 512), (None, 512)]
Add | output: | (None, 512)
batch_normalization_4 | input: | (None, 512)
BatchNormalization | output: | (None, 512)
dense_1 | input: | (None, 512)
Dense | output: | (None, 512)
dense_2 | input: | (None, 512)
Dense | output: | (None, 8262)

Fig. 7. CNN-RNN model architecture.

*4) Training Process:* We initialized the weights of the LSTM and embedding layers randomly and trained the entire image captioning model end-to-end. The objective was to minimize the discrepancy between the predicted captions and the ground truth captions in the training dataset. We used a variant of the cross-entropy loss, called the categorical cross-entropy loss, to compute the training loss. The loss was calculated between the predicted captions and the ground truth captions, considering each word as a separate class. To optimize the model, we employed back-propagation along with the Adam optimizer. We iteratively updated the model's weights by back-propagating the gradients of the loss through the network and adjusting the weights in the direction that minimized the loss.

*5) Hyperparameter Tuning:* During the training process, we carefully tuned the hyperparameters to achieve the best performance of our image captioning system. Some of the key hyperparameters we considered were:

- **Learning rate:** We experimented with different learning rates to find an optimal balance between convergence speed and stability.
- **Batch size:** We varied the batch size to control the number of training examples processed in each training iteration; we found that the larger the batch size the faster time to convergence was, and lower our per-epoch loss.
- **Number of LSTM units:** We adjusted the number of LSTM units to influence the model's capacity to capture sequential dependencies and generate captions effectively.
- **Dropout rate:** We applied dropout regularization to prevent overfitting. We tuned the dropout rate to strike a balance between model regularization and preserving important information.
- **Training epochs:** We determined the number of training epochs, considering factors such as convergence and model performance on validation data.

*6) Model Evaluation:* To assess the performance of our trained image captioning model, we utilized a separate validation dataset. During the training process, we periodically evaluated the model on the validation set to monitor its progress and prevent overfitting. For evaluation, we employed a metric commonly used in image captioning tasks: bilingual evaluation understudy (BLEU). This metric provided quantitative measures of the quality and similarity of the generated captions compared to the ground truth captions. By closely monitoring the model's performance on the validation set and iteratively adjusting the hyperparameters, we aimed to achieve the best possible captioning performance and ensure the model's ability to generalize to unseen images.

*7) Inference:* Once the model training was complete, we used the trained image captioning model for inference. Given a new image, we fed it through the ResNet-50 encoder to obtain the image features. These features were then passed as input to the trained LSTM decoder, which generated a caption word by word. The generated caption was post-processed to improve readability and coherence if necessary. The trained image captioning model was ready for deployment and capable of generating captions for new images, thereby providing a seamless integration of visual information with textual descriptions. By following these steps, we successfully trained an image captioning model using the ResNet-50 feature extractor and LSTM-based language model, allowing us to generate captions that capture the content and context of images in a meaningful way.

## IV. Results

To predict captions for the images using our trained model we used two decoding algorithms: Greedy Search and Beam Search

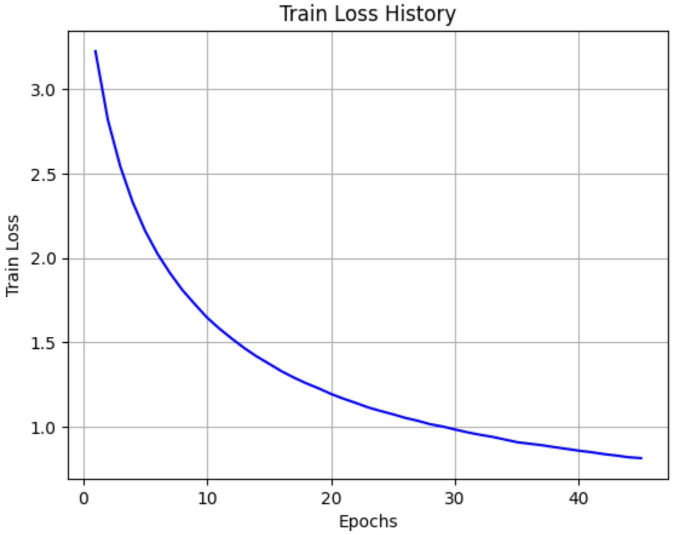


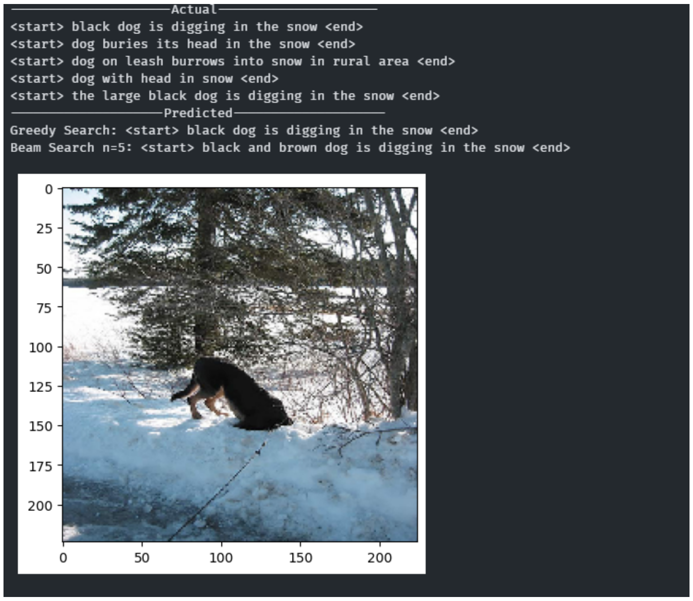
Fig. 8. Training loss history over 30 epochs.

- **Greedy search:** Greedy search is a simple and efficient decoding algorithm. It works by selecting the word with the highest probability at each step of the decoding process. In caption generation, at each time step, the model predicts the next word based on the current input and selects the word with the highest probability as the output. Greedy search does not consider future possibilities or explore alternative paths, resulting in a locally optimal choice at each step. However, this can lead to sub-optimal overall sequences, as the model may get stuck in a sub-optimal path early on.
- **Beam search:** Beam search is an enhanced decoding algorithm that explores a predefined number of candidate sequences simultaneously, similar to that of a breadth first search algorithm (BFS). It maintains a set of the top-k most likely partial sequences and expands each of them by considering all possible next words. The expanded sequences are then ranked based on their probabilities, and the top-k sequences are kept for further expansion. Beam search allows for considering multiple possibilities, increasing the chances of finding a globally better sequence. The beam width parameter determines the number of candidate sequences to consider.

We found greedy search to be faster than beam search but beam search provided a much more descriptive caption. Below is an example of a caption generated using greedy and beam search; we can see that beam search was able to pick up on the varied colour of the dog's coat, where greedy search treated the coat as a single colour.

## V. Discussion

We were able to develop a successful deep learning-based image captioning system that utilized a sequential approach. The model consisted of two main components: an Encoder layer and a Decoder layer. The encoder layer utilized a CNN



```
─────────────────────Actual─────────────────────
<start> black dog is digging in the snow <end>
<start> dog buries its head in the snow <end>
<start> dog on leash burrows into snow in rural area <end>
<start> dog with head in snow <end>
<start> the large black dog is digging in the snow <end>
────────────────────Predicted────────────────────
Greedy Search: <start> black dog is digging in the snow <end>
Beam Search n=5: <start> black and brown dog is digging in the snow <end>
```

Fig. 9. First Random Prediction.

to extract meaningful features from the input images, while the decoder layer employed an LSTM network to generate a descriptive caption for a given input image.

Overall, the model performed remarkably well in generating accurate and meaningful image captions, considering the relative simplicity and plug-and-play nature of the architecture. The combination of CNNs and LSTMs allowed the model to effectively capture the visual features of the images and generate coherent and contextually relevant captions.

To enhance the performance of the model, we incorporated various techniques such as dropout, layer normalization, and batch normalization. These techniques helped in reducing overfitting, improving the generalization capability of the model, and ensuring the stability of the training process. By pruning and normalizing inputs between layers, we achieved better convergence and optimized the overall performance of the model.

While our model achieved satisfactory results, there are potential avenues for further improvement. One promising approach is the utilization of transformer-based architectures, such as the ViT model mentioned briefly early on in the paper. Transformers have demonstrated superior performance in various natural language processing tasks, including language translation and text generation. By incorporating transformers into our image captioning model, we could potentially enhance the model's ability to capture long-range dependencies and improve the quality of the generated captions.

Additionally, expanding the dataset used for training could also contribute to better performance. A larger and more diverse dataset would enable the model to learn from a wider range of visual and textual patterns, leading to more accurate and contextually rich captions.

## VI. Conclusion

In conclusion, this project successfully utilized deep learning techniques, combining computer vision and NLP, to generate descriptive captions for images. The sequential deep learning model, comprising an of encoder layer based on CNN and a decoder layer based on an LSTM, proved effective in capturing image features and generating human-like captions.

The model was trained and validated using the Flickr8k dataset, consisting of 8,000 images with 5 captions each. Features were extracted from the images using ResNet50, and the model was trained using these features. Evaluation was performed using BLEU scores, a metric for assessing the quality of machine-translated text.

Two caption generation algorithms, greedy and beam search, were employed. While greedy search was faster, beam search provided more descriptive captions, albeit at a slower speed. Techniques such as dropout, layer normalization, and batch normalization were implemented to enhance model performance and stability.

Overall, the project yielded successful outcomes, enhancing our understanding of deep learning and neural networks. Given more time, integrating transformer-based architectures could be explored to further enhance the model's efficiency and performance.

In summary, this project demonstrated the effectiveness of deep learning in generating accurate and contextually relevant image captions. The combination of computer vision and NLP techniques, along with the sequential deep learning model, holds promise for various applications in image captioning and related fields.

## VII. Acknowledgments

This paper was part of the final project for the Summer 2023 session of SENG 474, Data Mining; a course taught by Dr. Alex Thomo. We would like to thank Dr. Thomo for his guidance throughout the courses entirety.

## References

[1] A. Jain, "Flickr 8K dataset," Kaggle, https://www.kaggle.com/datasets/adityajn105/flickr8k (accessed Jun. 5, 2023).

[2] Alam et al., "Comparison of Different CNN Model used as Encoders for Image Captioning", ResearchGate, Comparison of Different CNN Model used as Encoders for Image Captioning (accessed Jun. 26, 2023).

[3] 5. categorizing and tagging words, https://www.nltk.org/book/ch05.html (accessed Jun. 21, 2023).

[4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv.org, https://doi.org/10.48550/arXiv.1502.03167 (accessed Jun. 25, 2023).