IMAGE CAPTION GENERATION USING CNN+GRU

Team Members:

Bhanu Sai Praneeth Sarva
Ashwin Muthuraman
Bhoomika Nanjaraja

CONTENTS

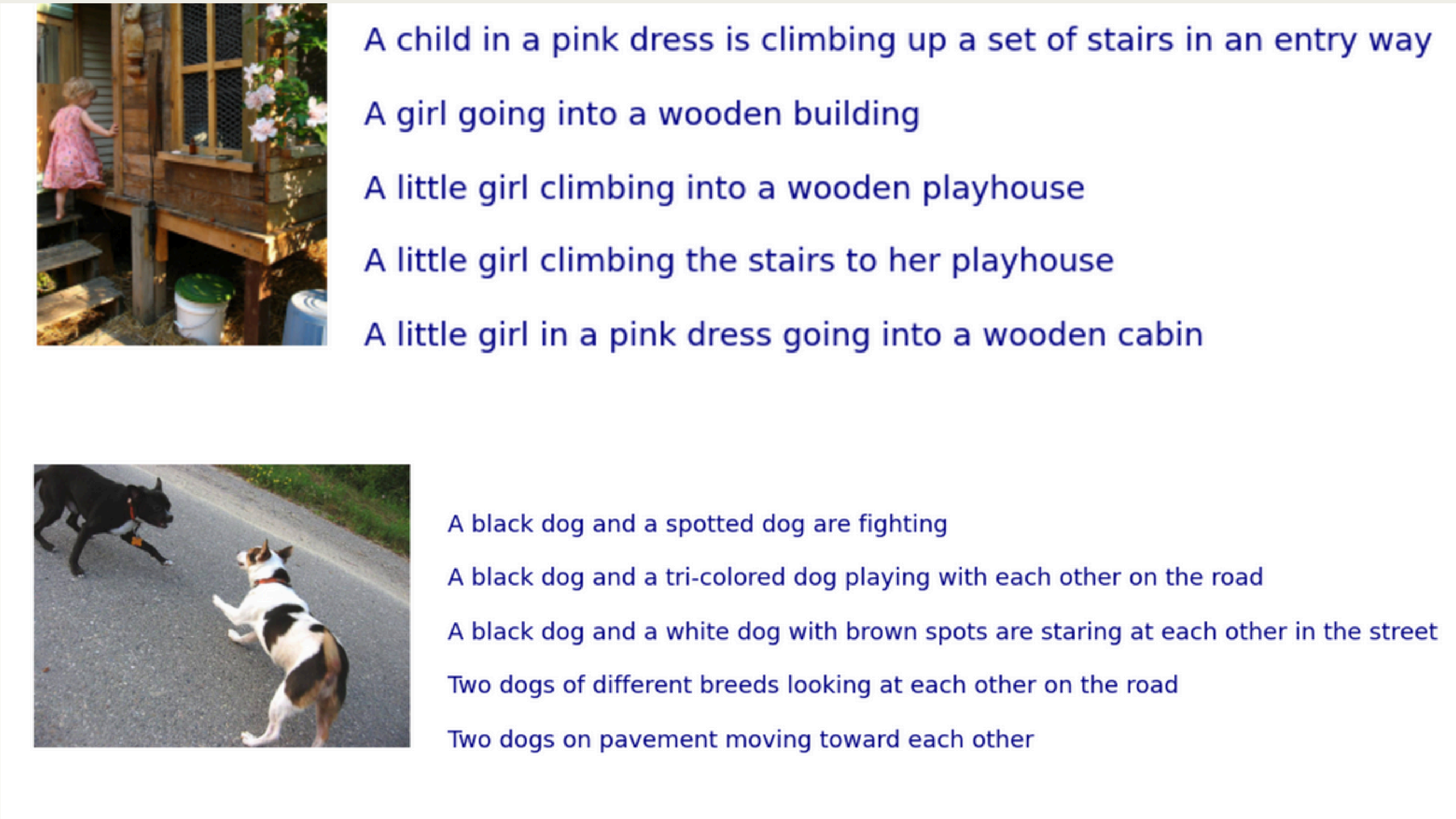
- Initial Setup
- Data Understanding and Visualization (Flicker8k)
- Data Cleaning
- Data Preprocessing
  - Captions Preprocessing (tokenizer , mappings)
  - Images Preprocessing (Resizing, Normalizing)
- Dataset Creation
- Model Building (Encoder , Decoder, Attention Model)
- Model Training and Optimization (hyperparameter tuning, optimizer, loss function)
- Model Evaluation (Greedy Search & BLEU score)

OBJECTIVE

- The objective of this project is to develop an image captioning model that generates descriptive captions for images. The model utilizes pretrained CNNs (e.g Inception-V3) for feature extraction to encode visual information and an encoder-decoder architecture with attention mechanisms to focus on specific image regions while generating captions. The decoder, typically implemented using GRUs, processes textual data and generates contextually relevant captions. The aim is to achieve a seamless integration of visual and textual understanding.

DATA UNDERSTANDING

- Dataset Overview: Flickr8k dataset used for image captioning tasks.
- Images Folder: Contains 8,091 total images.
- Captions File: captions.txt includes image names and corresponding captions.
- Caption Details: Each image is associated with 5 captions.
- Total Dataset Size: 40,455 samples (images × captions).



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

A black dog and a spotted dog are fighting

A black dog and a tri-colored dog playing with each other on the road

A black dog and a white dog with brown spots are staring at each other in the street

Two dogs of different breeds looking at each other on the road

Two dogs on pavement moving toward each other

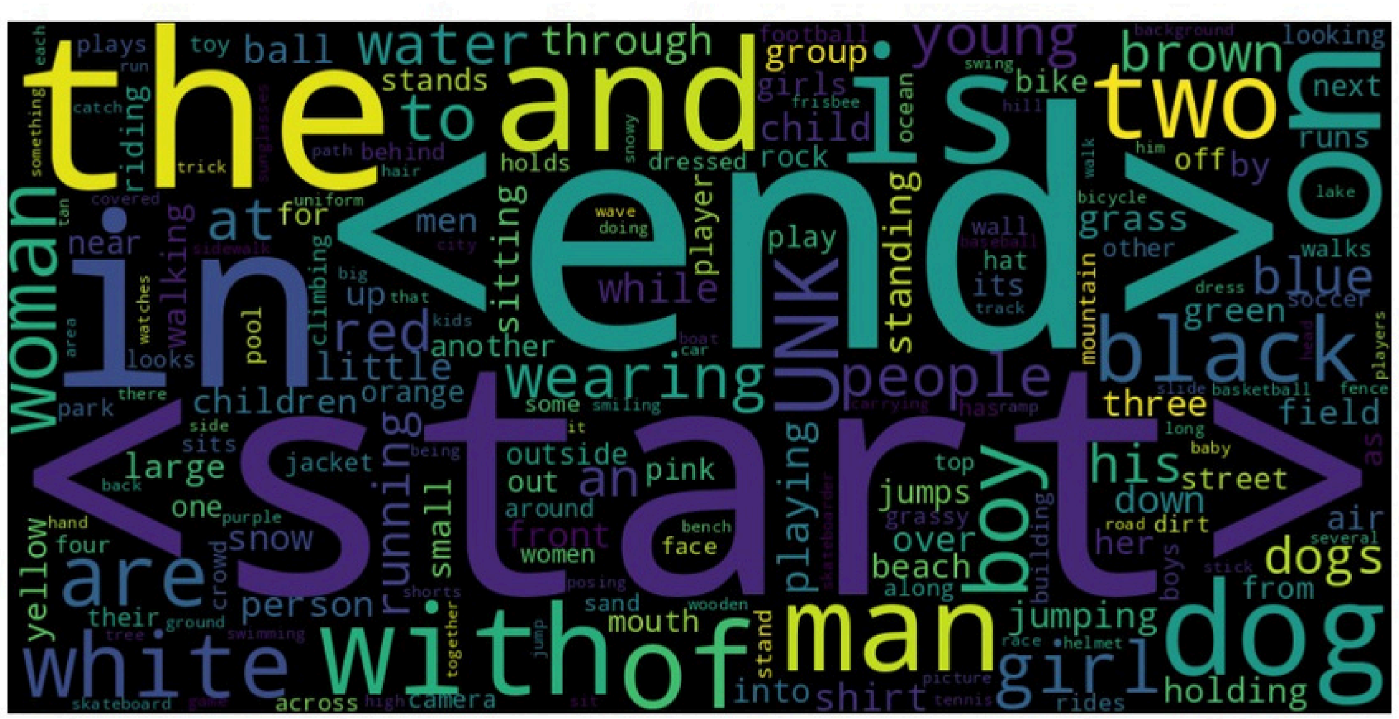
DATA CLEANING

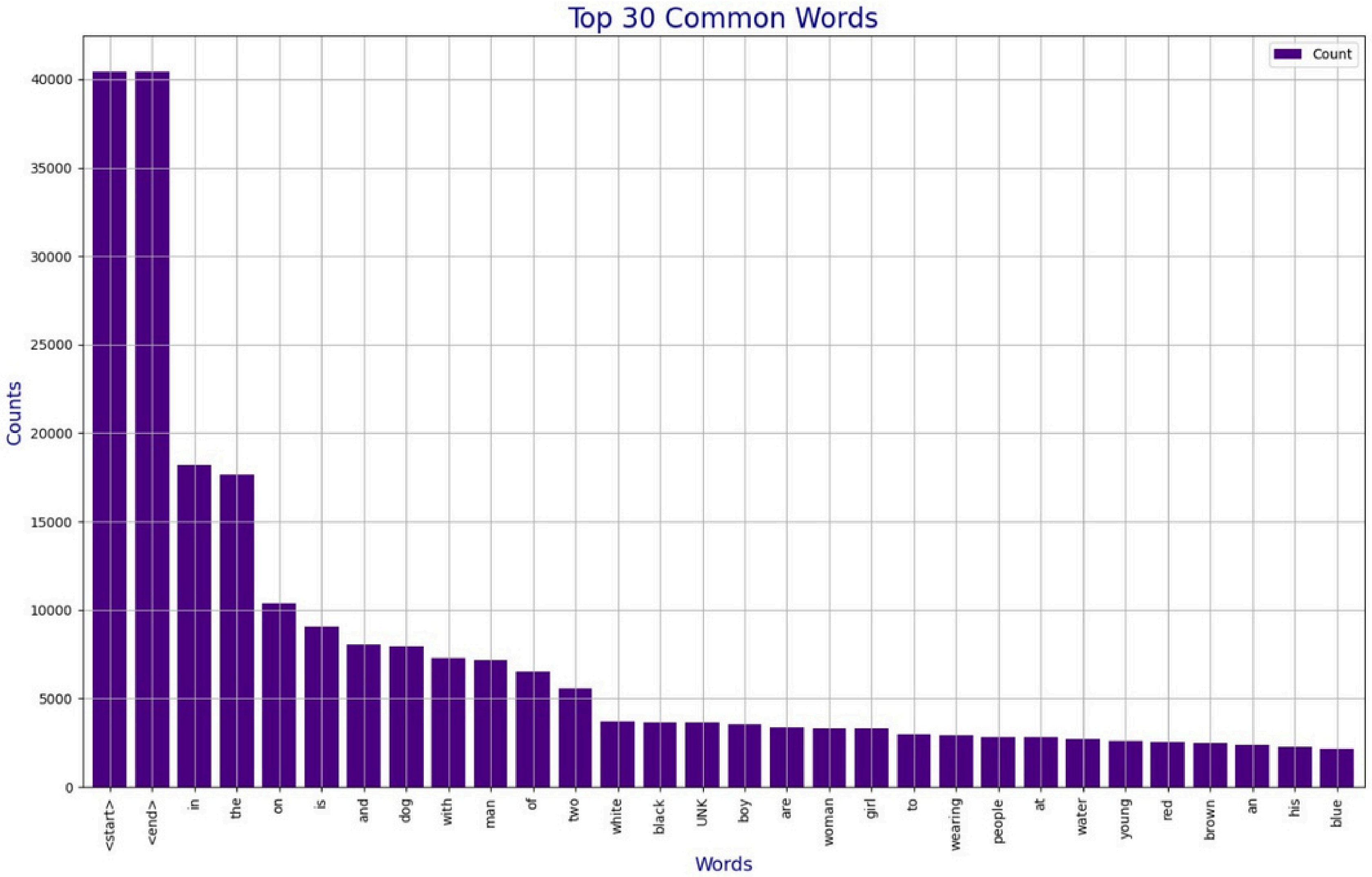
- Remove punctuations
- Convert captions to lowercase
- Retain words and eliminate numeric values

DATAPREPROCESSING

- Data Annotation -'<start> child in pink dress is climbing up set of stairs in an entry way <end>'
- Tokenize Captions
- Replace Rare Words with "UNK"
- Create Word Mappings
- Pad Sequences

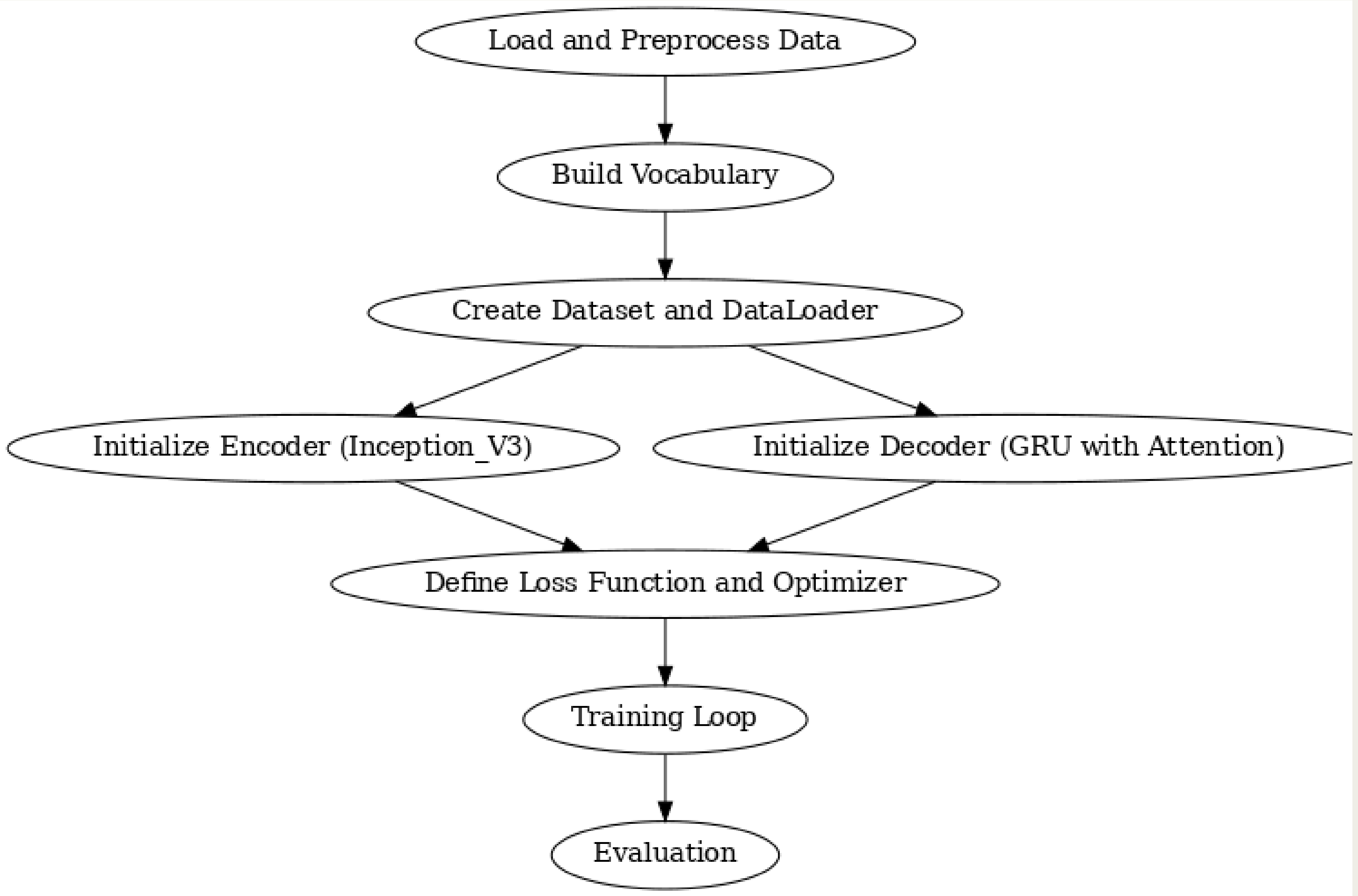
VISUALIZATION



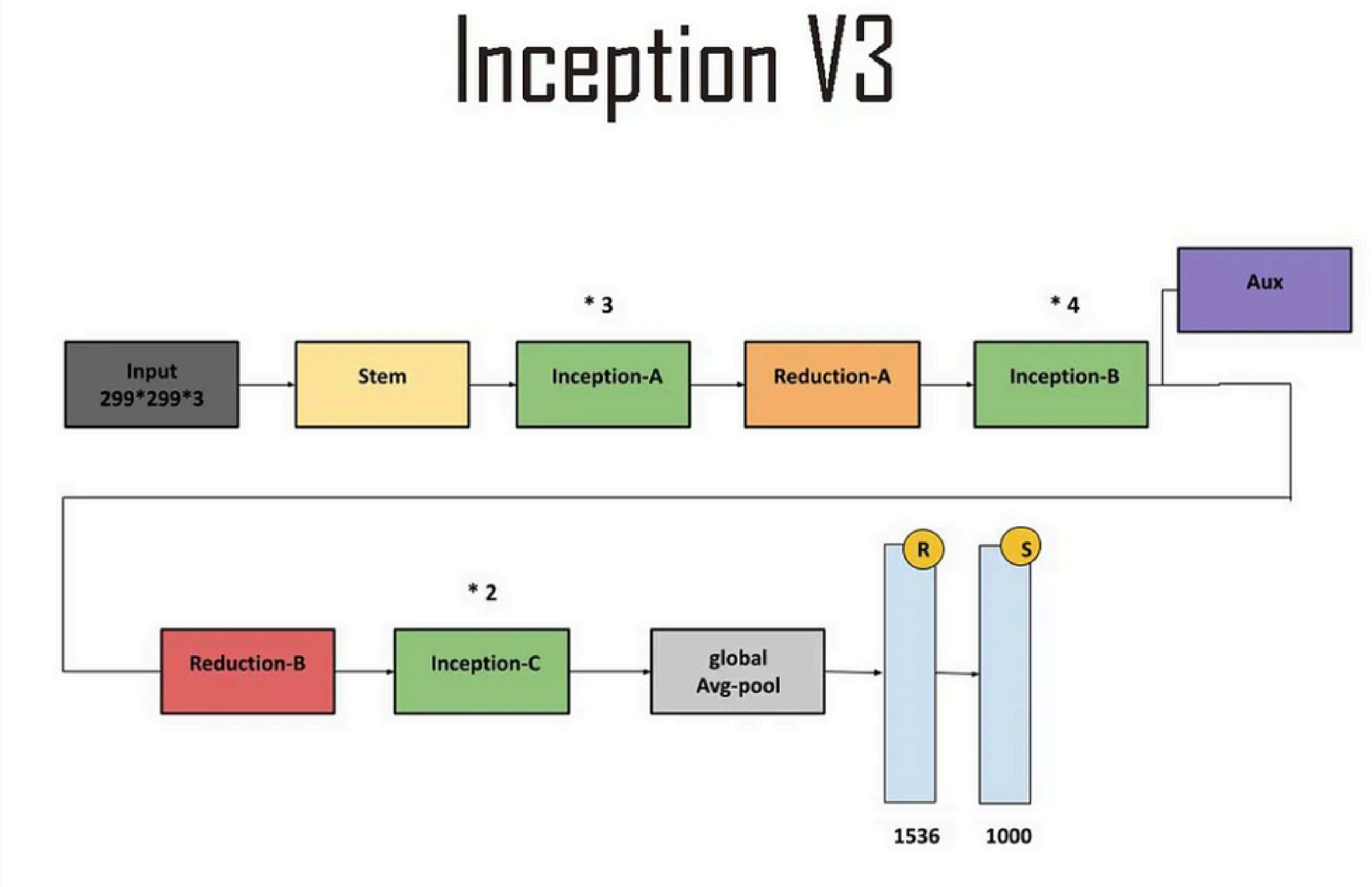


DATASET CREATION AND FLOW CHART

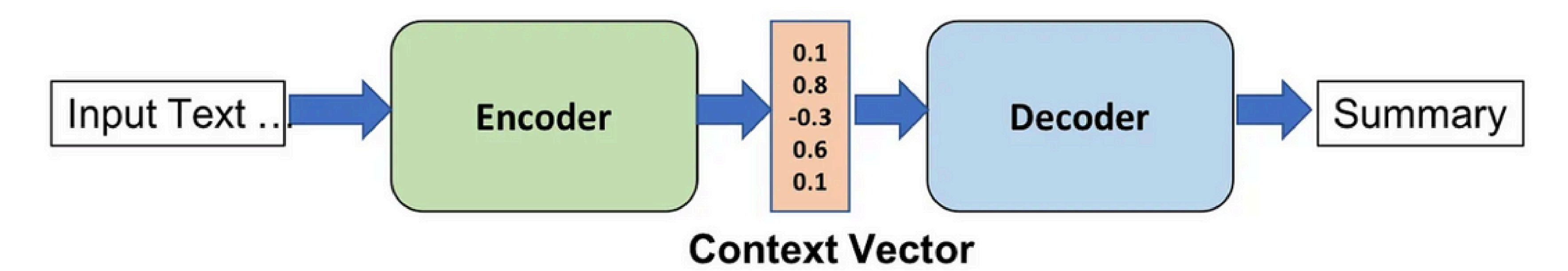
- Image Preprocessing -
  Resize images to shape : (299, 299)
  Normalize the image within the range of -1 to 1
- INCEPTION-V3 - Feature extraction of the images using pre trained weights from the ImageNet dataset
- Feature maps of images and tokens of text mapped together
- Passing the feature maps with mapped tokens to the model for training


Load and Preprocess Data
Build Vocabulary
Create Dataset and DataLoader
Initialize Encoder (Inception_V3)
Initialize Decoder (GRU with Attention)
Define Loss Function and Optimizer
Training Loop
Evaluation

MODEL ARCHITECTURE



FEATURES OF THE INPUT IMAGE

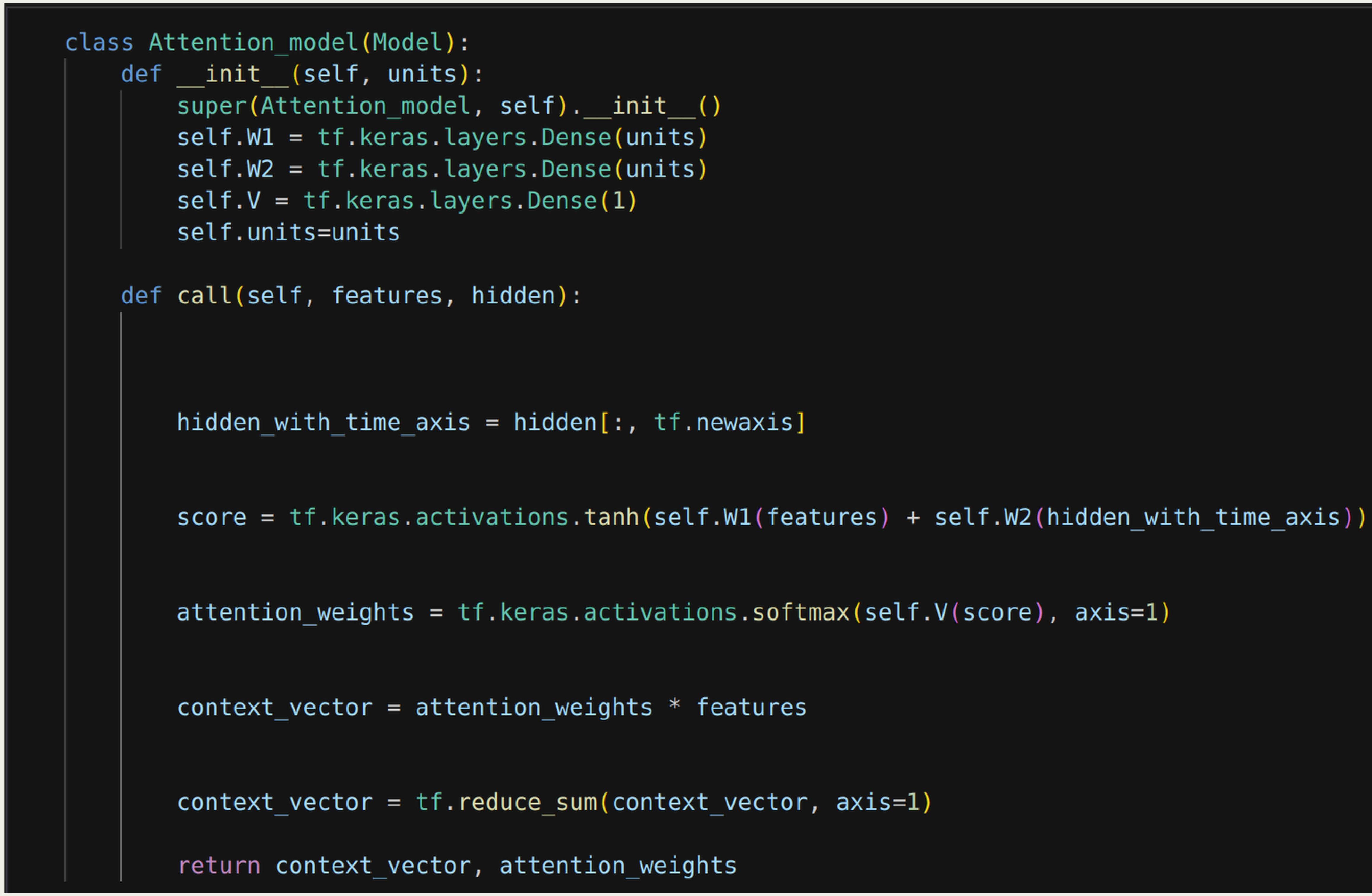


MODEL ARCHITECTURE

Attention serves as a bridge between the encoder and decoder, enabling the model to focus on relevant parts of the input image dynamically for each timestamp.

Instead of feeding the entire image to the decoder repeatedly, attention provides an adaptive context vector, improving efficiency and accuracy.

This mechanism overcomes the limitations of traditional CNN-RNN models by selectively passing only the most relevant image features to the decoder.



```
class Attention_model(Model):
    def __init__(self, units):
        super(Attention_model, self).__init__()
        self.W1 = tf.keras.layers.Dense(units)
        self.W2 = tf.keras.layers.Dense(units)
        self.V = tf.keras.layers.Dense(1)
        self.units=units

    def call(self, features, hidden):



        hidden_with_time_axis = hidden[:, tf.newaxis]


        score = tf.keras.activations.tanh(self.W1(features) + self.W2(hidden_with_time_axis))


        attention_weights = tf.keras.activations.softmax(self.V(score), axis=1)


        context_vector = attention_weights * features


        context_vector = tf.reduce_sum(context_vector, axis=1)

        return context_vector, attention_weights
```

MODEL DEFINITION

- Optimizer - Adam
- Loss Function - SparseCategoricalCrossentropy
- Callbacks - EarlyStopping with patience = 5
- We define functions to do
    - shuffling - Reshuffles the dataset order at the start of each epoch for better generalization.
    - batching - gradients are averaged over multiple samples , to process multiple inputs simultaneously for effective computation.

TECHNIQUES

- DataLoader (Batching): Created a DataLoader with the custom collate function to generate batches of image tensors and padded caption tensors.

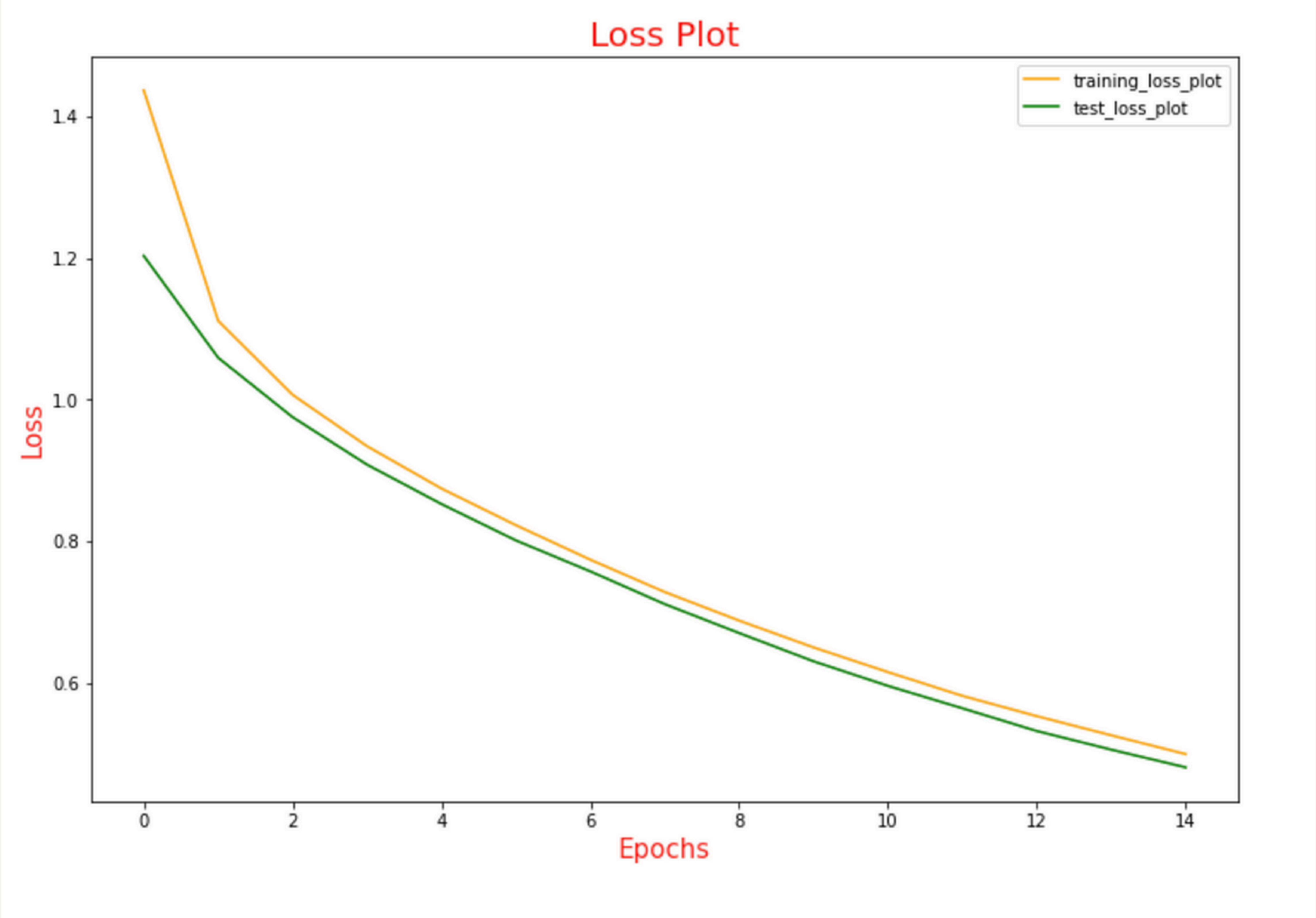
- Used Validation Dataset Along with Train to generalize the model

- Masking - Without masking, the model treats padded inputs as valid data, leading to incorrect predictions and increased loss.

- Attention Mechanism: Attention mechanism is uniquely designed to process image features extracted by a CNN, ensuring that spatial information from the image is preserved while dynamically focusing on regions relevant to the caption generation at each decoding step.

- Teacher Forcing: Employed teacher forcing with a specified ratio to stabilize training.

LOSS CURVE


Loss Plot
training_loss_plot
test_loss_plot
Loss
1.4
1.2
1.0
0.8
0.6
0
2
4
6
8
10
12
14
Epochs

MODEL EVALUATION

Greedy Search:

- A simple text generation method where the model selects the word with the highest probability at each step.
- Fast and computationally efficient but may miss better overall sequences as it doesn't explore alternatives.
- Used to generate captions word-by-word until the <end> token is reached.

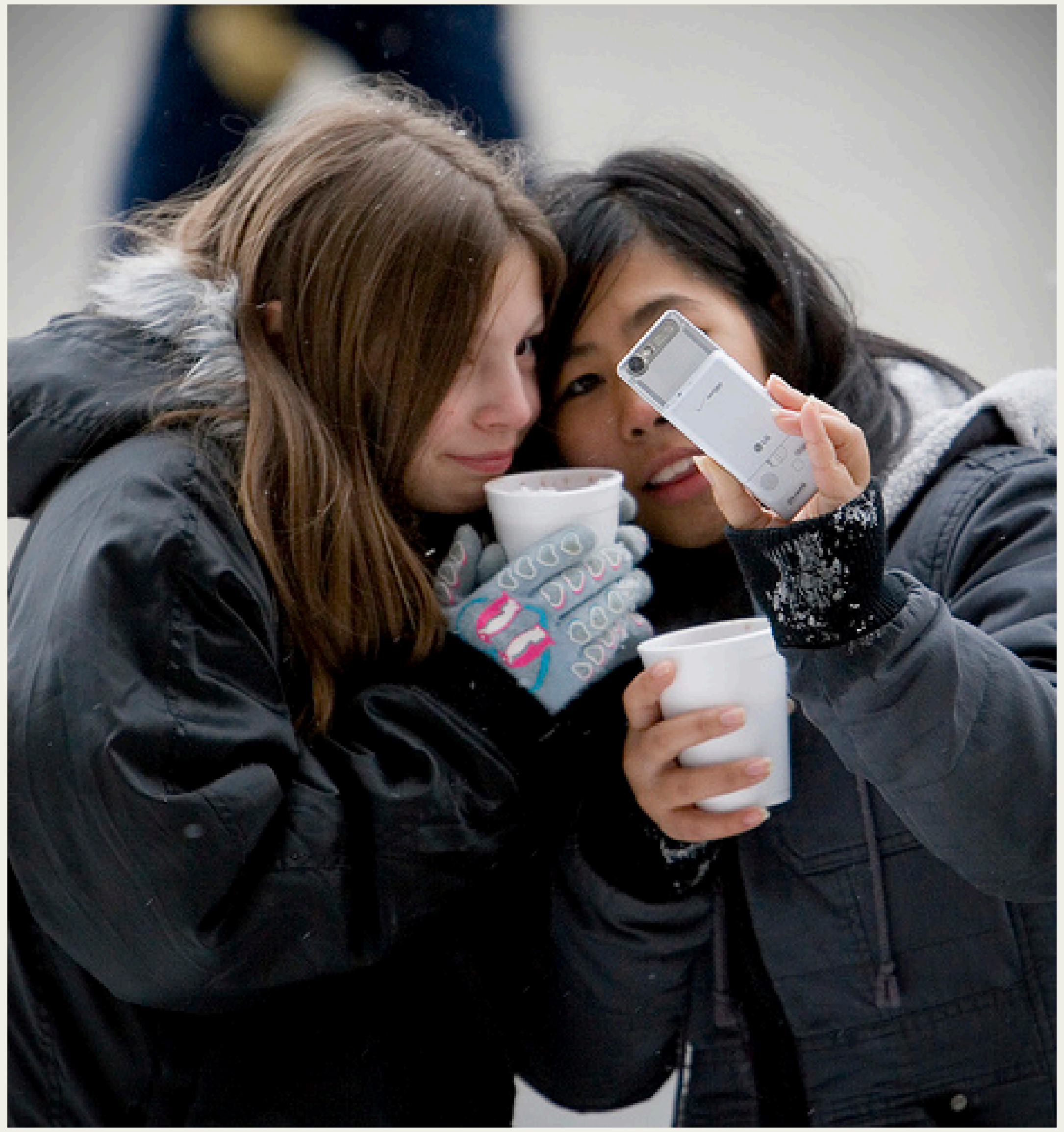
BLEU Score:

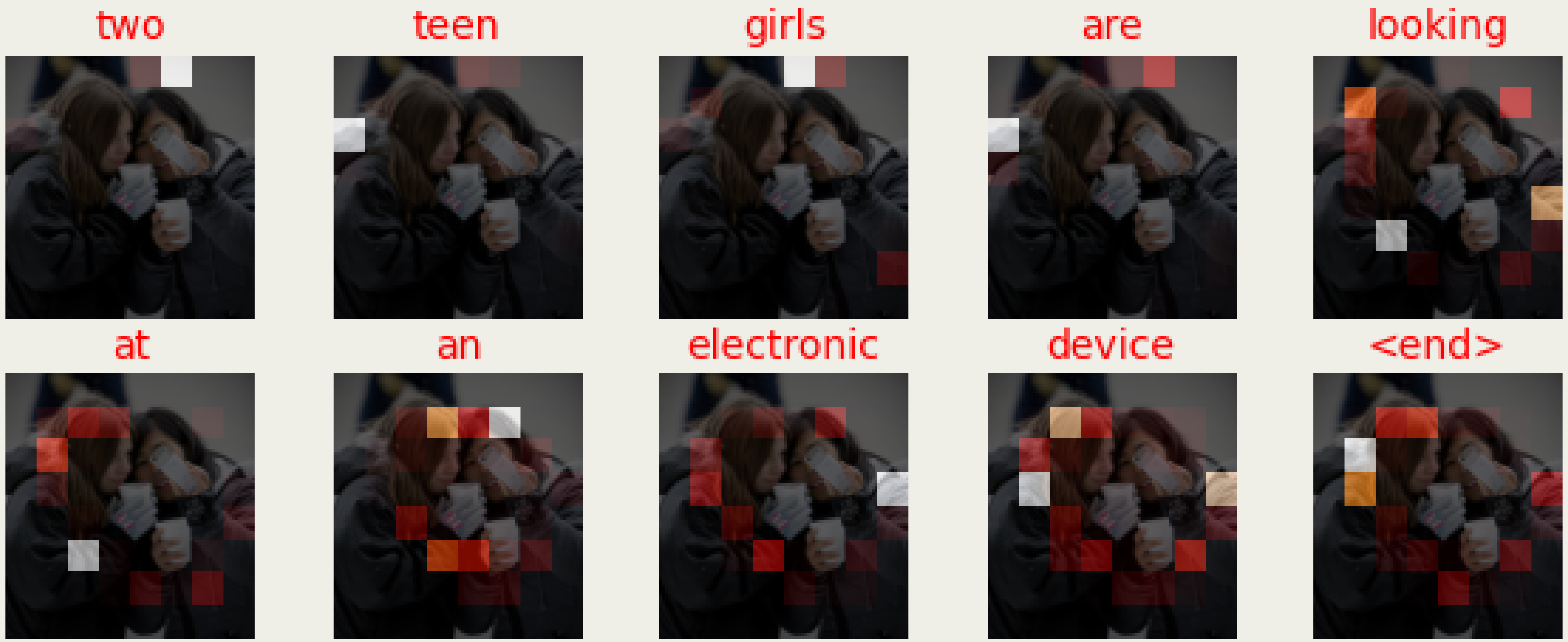
- A metric to evaluate the quality of machine-generated text compared to human-written reference text.
- Measures how many words and phrases in the generated text match the reference.
- Scores range from 0 to 1 (or 0% to 100%), with higher scores indicating better quality.
- Commonly used in tasks like image captioning and machine translation.

MODEL EVALUATION

- Weights Used in BLEU Score:
- (0.5, 0.5, 0, 0):
  - 50% importance to unigrams (individual words).
  - 50% importance to bigrams (pairs of consecutive words).
  - Ignores trigrams and 4-grams.
- (0.25, 0.25, 0, 0):
  - 25% importance to unigrams and bigrams.
  - Ignores longer sequences (trigrams and 4-grams).
- Results:
- Achieved accuracy between 50–70% for test images.
- These weights prioritize matching individual words and short phrases, making them effective for short captions.

MODEL EVALUATION - BLEU SCORE



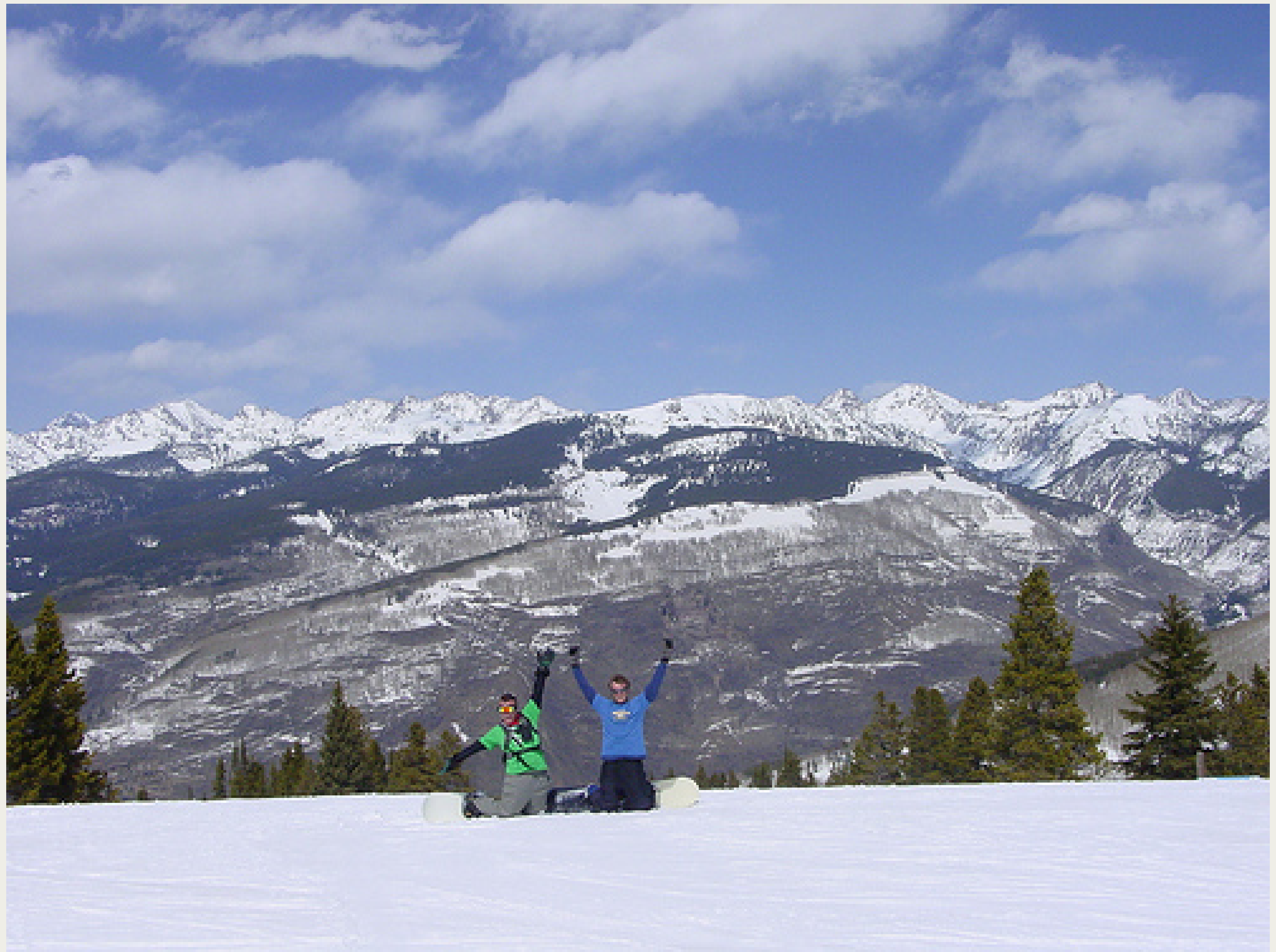


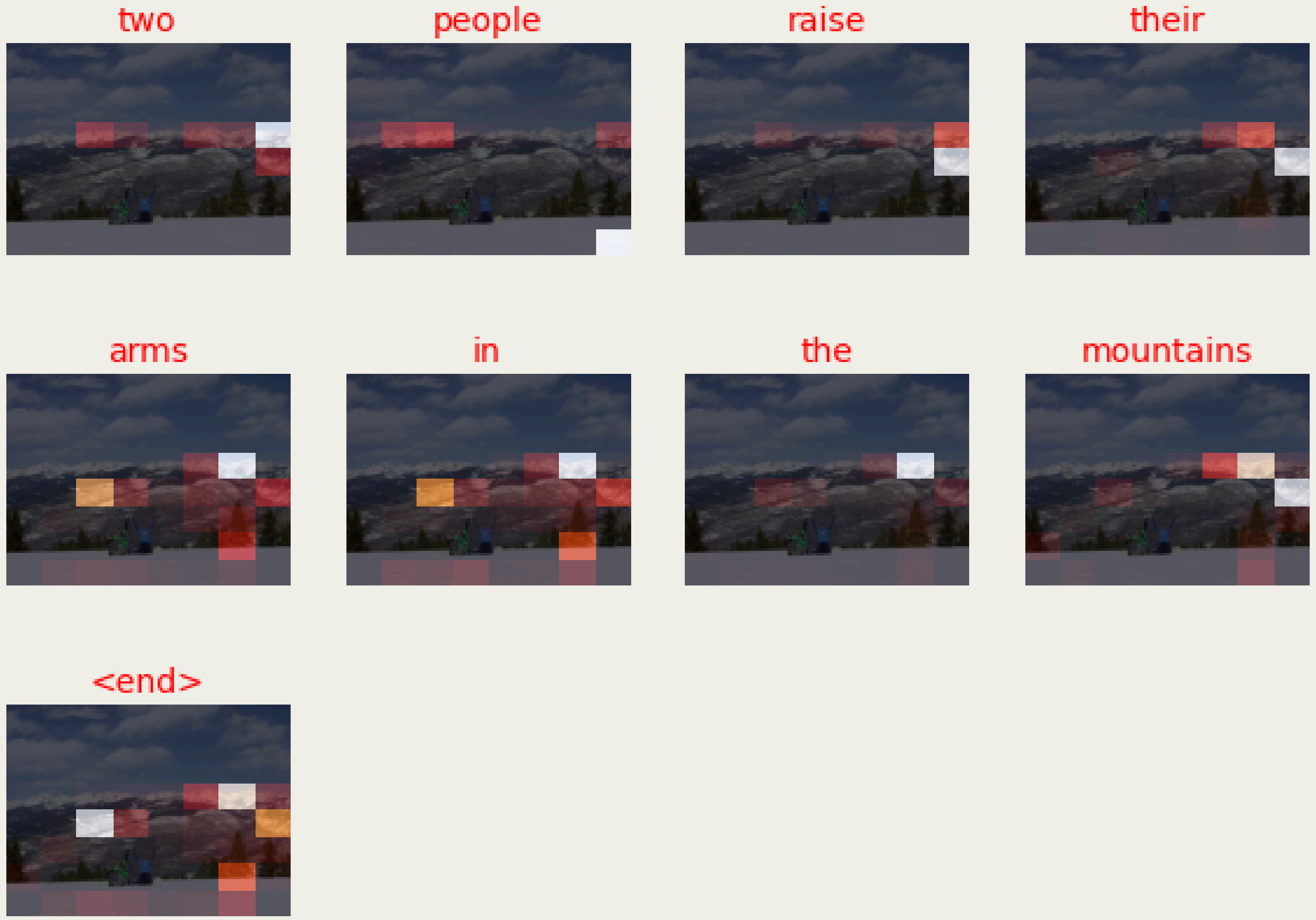
BLEU SCORE: 52.35

REAL CAPTION: TWO TEEN GIRLS ARE LOOKING AT SMALL ELECTRONIC DEVICE WHILE WEARING WINTER COATS

PREDICTED CAPTION: TWO TEEN GIRLS ARE LOOKING AT AN ELECTRONIC DEVICE

MODEL EVALUATION





BLEU SCORE: 66.13

ACTUAL CAPTION: TWO PEOPLE RAISE THEIR ARMS ON SNOWY HILL IN THE MOUNTAINS

PREDICTED CAPTION: TWO PEOPLE RAISE THEIR ARMS IN THE MOUNTAINS

STREAMLIT

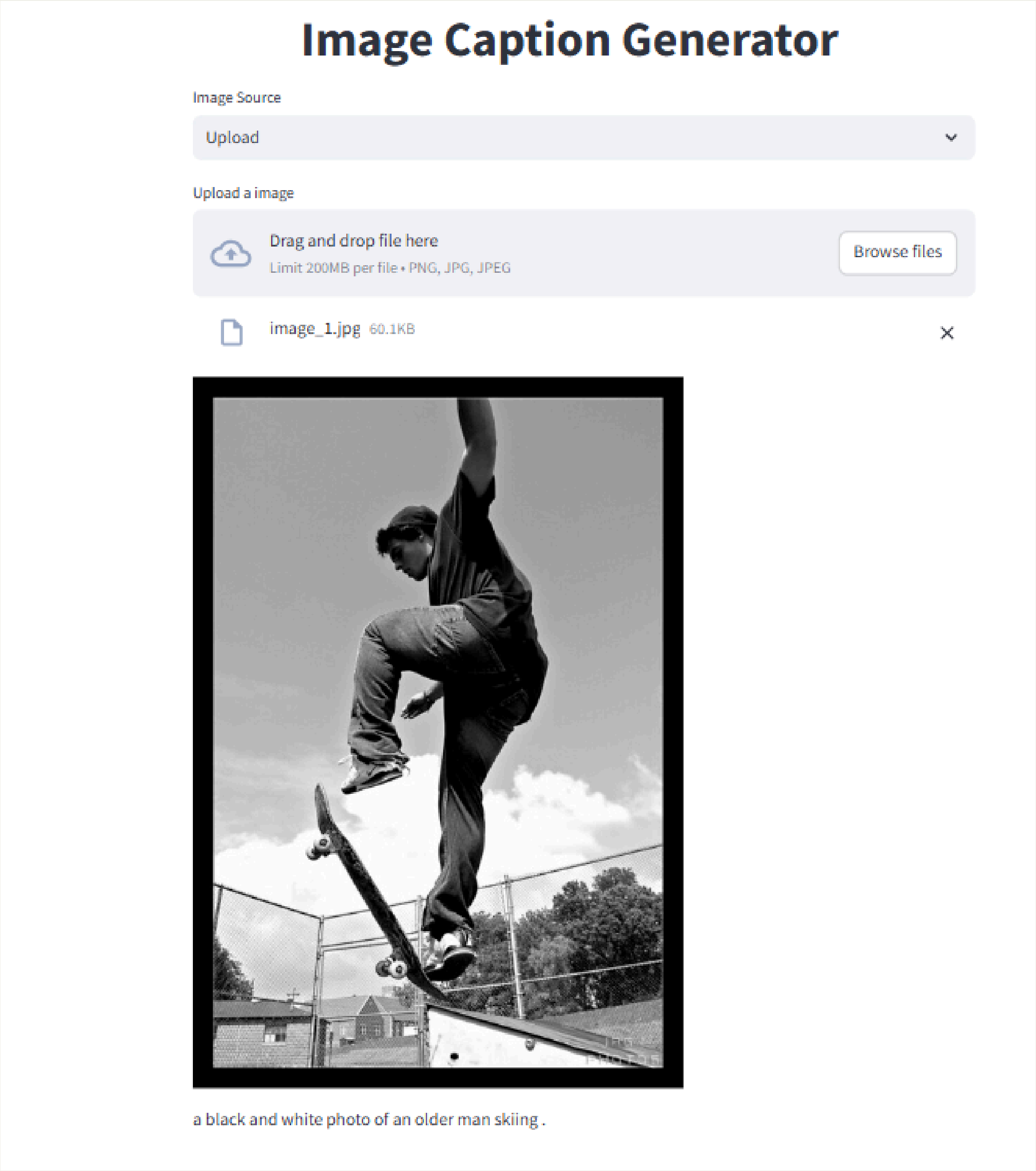


Image Caption Generator

Image Source: Upload

Upload a image

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

image_1.jpg 60.1KB

a black and white photo of an older man skiing .

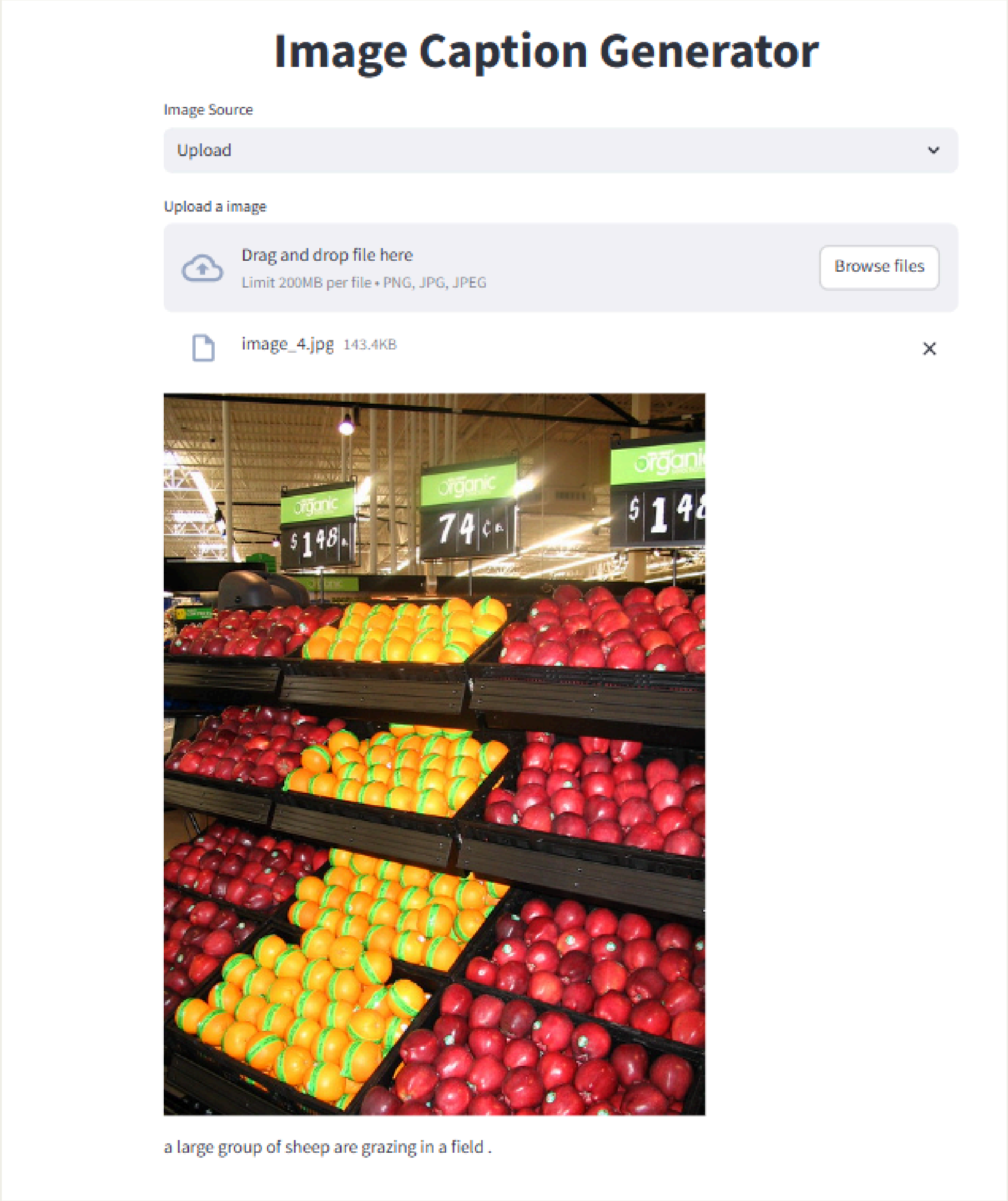


Image Caption Generator

Image Source: Upload

Upload a image

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG

image_4.jpg 143.4KB

a large group of sheep are grazing in a field .

Thank you!