e-caption-with-pytorch-resnet-lstm

August 24, 2024

[1]:
```
import numpy as np
import pandas as pd
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

[2]:
```
image_data_location = "../input/flickr8k/Images"
caption_data_location = "../input/flickr8k/captions.txt"
```

[3]:
```
df = pd.read_csv(caption_data_location)
```

[4]:
```
df.head()
```

[4]:
```
                         image  \
0  1000268201_693b08cb0e.jpg
1  1000268201_693b08cb0e.jpg
2  1000268201_693b08cb0e.jpg
3  1000268201_693b08cb0e.jpg
4  1000268201_693b08cb0e.jpg

                                             caption
0  A child in a pink dress is climbing up a set o…
1              A girl going into a wooden building .
2   A little girl climbing into a wooden playhouse .
3  A little girl climbing the stairs to her playh…
4  A little girl in a pink dress going into a woo…
```

[5]:
```
data_idx = 11
image_path = image_data_location + "/" + df.iloc[data_idx,0]
# print( df.iloc[data_idx,:])
img = mpimg.imread(image_path)
plt.imshow(img)
plt.show()

for i in range(data_idx, data_idx+5):
    print(f"Caption - {df.iloc[i,1]}")
```


```
Caption - A little girl is sitting in front of a large painted rainbow .
Caption - A small girl in the grass plays with fingerpaints in front of a white
canvas with a rainbow on it .
Caption - There is a girl with pigtails sitting in front of a rainbow painting .
Caption - Young girl with pigtails painting outside in the grass .
Caption - A man lays on a bench while his dog sits by him .
```

[6]:
```
import os
from collections import Counter
import spacy
import torch
from torch.nn.utils.rnn import pad_sequence
from torch.utils.data import DataLoader, Dataset
import torchvision.transforms as T
```

[7]:
```
spacy_eng = spacy.load('en_core_web_sm')
text = "This is a good place to find a city"
[token.text.lower() for token in spacy_eng.tokenizer(text)]
```

[7]:
```
['this', 'is', 'a', 'good', 'place', 'to', 'find', 'a', 'city']
```

[8]:
```
class Vocabulary:
    def __init__(self,freq_threshold):
        self.itos = {0:"<PAD>",1:"<SOS>",2:"<EOS>",3:"<UNK>"}
        self.stoi = {v:k for k,v in self.itos.items()}
        self.freq_threshold = freq_threshold
```

```
    def __len__(self):
        return len(self.itos)

    @staticmethod
    def tokenize(text):
        return [token.text.lower() for token in spacy_eng.tokenizer(text)]

    def build_vocab(self,sentence_list):
        frequencies = Counter()
        idx = 4
        for sentence in sentence_list:
            for word in self.tokenize(sentence):
                frequencies[word] += 1

                if frequencies[word] == self.freq_threshold:
                    self.stoi[word] = idx
                    self.itos[idx] = word
                    idx += 1

    def numericalize(self,text):
        tokenized_text = self.tokenize(text)
        return [self.stoi[token] if token in self.stoi else self.stoi["<UNK>"]␣
↪for token in tokenized_text]
```

[9]:
```
v = Vocabulary(freq_threshold=1)
v.build_vocab(["This is a new city"])
print(v.stoi)
print(v.numericalize("This is a new city"))
```

```
{'<PAD>': 0, '<SOS>': 1, '<EOS>': 2, '<UNK>': 3, 'this': 4, 'is': 5, 'a': 6,
'new': 7, 'city': 8}
[4, 5, 6, 7, 8]
```

[10]:
```
df = pd.read_csv(caption_data_location)
print(df["image"][0][::-1])
```

```
gpj.e0bc80b396_1028620001
```

[11]:
```
class CustomDataset(Dataset):
    def __init__(self,root_dir,captions_file,transform=None, freq_threshold=5):
        self.root_dir = root_dir
        self.df = pd.read_csv(captions_file)

        self.transform = transform
        self.imgs = self.df["image"]
        self.captions = self.df["caption"]
```

```
        self.vocab = Vocabulary(freq_threshold)
        self.vocab.build_vocab(self.captions.tolist())

    def __len__(self):
        return len(self.df)

    def __getitem__(self,idx):
        caption = self.captions[idx]
        img_name = self.imgs[idx]

        img_location = os.path.join(self.root_dir,img_name)
        img = Image.open(img_location).convert("RGB")

        if self.transform is not None:
            img = self.transform(img)

        caption_vec = []
        caption_vec += [self.vocab.stoi["<SOS>"]]
        caption_vec += self.vocab.numericalize(caption)
        caption_vec += [self.vocab.stoi["<EOS>"]]

        return img, torch.tensor(caption_vec)
```

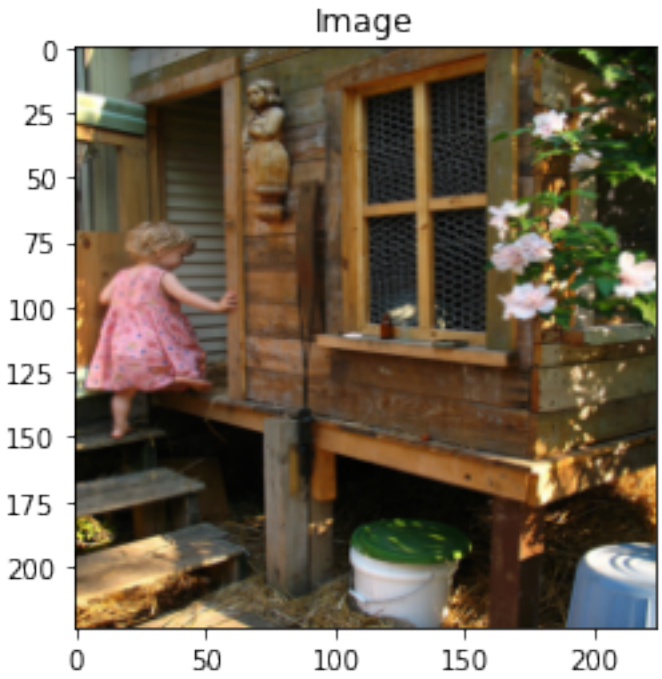
[12]:
```
#defing the transform to be applied
transforms = T.Compose([
    T.Resize((224,224)),
    T.ToTensor()
])
```

[13]:
```
def show_image(inp, title=None):
    """Imshow for Tensor"""
    inp = inp.numpy().transpose((1,2,0))
    plt.imshow(inp)
    if title is not None:
        plt.title(title)
    plt.pause(0.001)
```

[14]:
```
# testing the dataset
dataset = CustomDataset(
        root_dir = image_data_location,
        captions_file = caption_data_location,
        transform = transforms
)
```

[15]:
```
img, caps = dataset[0]
# print(caps)
```

```
show_image(img,"Image")
print("Token :",caps)
print("Sentence: ")
print([dataset.vocab.itos[token] for token in caps.tolist()])
```




```
Token : tensor([  1,   4,  28,   8,   4, 195, 151,  17,  32,  67,   4, 353,  11,
711,
          8,  24,   3, 496,   5,   2])
Sentence:
['<SOS>', 'a', 'child', 'in', 'a', 'pink', 'dress', 'is', 'climbing', 'up', 'a',
'set', 'of', 'stairs', 'in', 'an', '<UNK>', 'way', '.', '<EOS>']
```

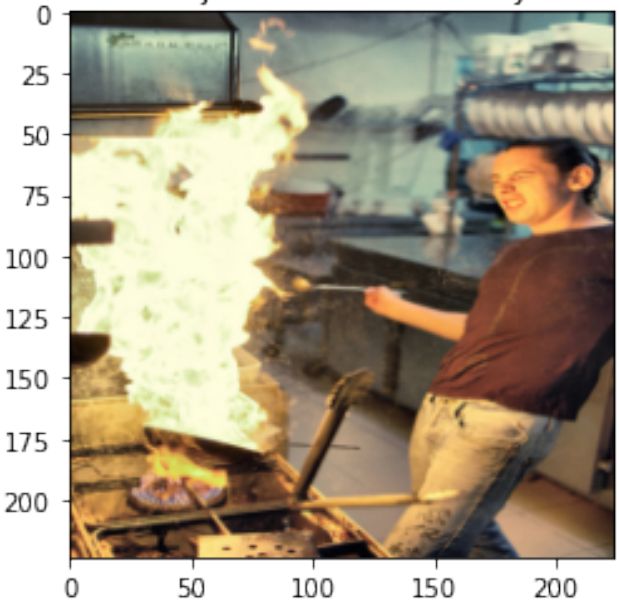
[16]:
```
class CapsCollate:
    def __init__(self,pad_idx,batch_first=False):
        self.pad_idx = pad_idx
        self.batch_first = batch_first

    def __call__(self,batch):
        imgs = [item[0].unsqueeze(0) for item in batch]
#         print(f"shape - {(imgs)}")
#         print("----"*22)
        imgs = torch.cat(imgs,dim=0)
#         print(f"shape - {imgs}")
#         print("------")
        targets = [item[1] for item in batch]
```

```
        targets = pad_sequence(targets, batch_first=self.batch_first,␣
 ↪padding_value=self.pad_idx)
        return imgs,targets
```

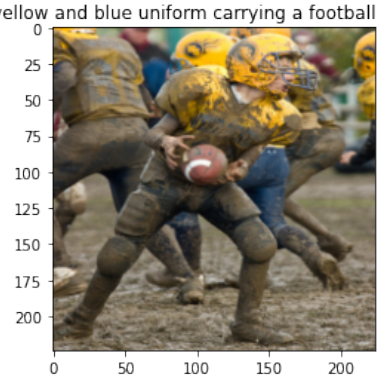
[17]:
```
#writing the dataloader
#setting the constants
BATCH_SIZE = 4
NUM_WORKER = 1

#token to represent the padding
pad_idx = dataset.vocab.stoi["<PAD>"]

data_loader = DataLoader(
    dataset=dataset,
    batch_size=BATCH_SIZE,
    num_workers=NUM_WORKER,
    shuffle=True,
    collate_fn=CapsCollate(pad_idx=pad_idx,batch_first=True)
)
```

[18]:
```
#generating the iterator from the dataloader
dataiter = iter(data_loader)

#getting the next batch
batch = next(dataiter)

#unpacking the batch
images, captions = batch

#showing info of image in single batch
for i in range(BATCH_SIZE):
    img,cap = images[i],captions[i]
#     print(f"captions - {captions[i]}")
    caption_label = [dataset.vocab.itos[token] for token in cap.tolist()]
    eos_index = caption_label.index('<EOS>')
    caption_label = caption_label[1:eos_index]
    caption_label = ' '.join(caption_label)
    show_image(img,caption_label)
    plt.show()
```

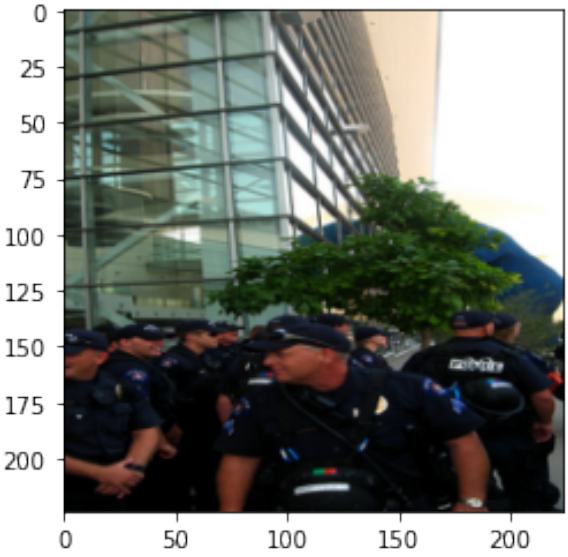
a man in a black shirt and jeans is <UNK> away from a fire on a stove .



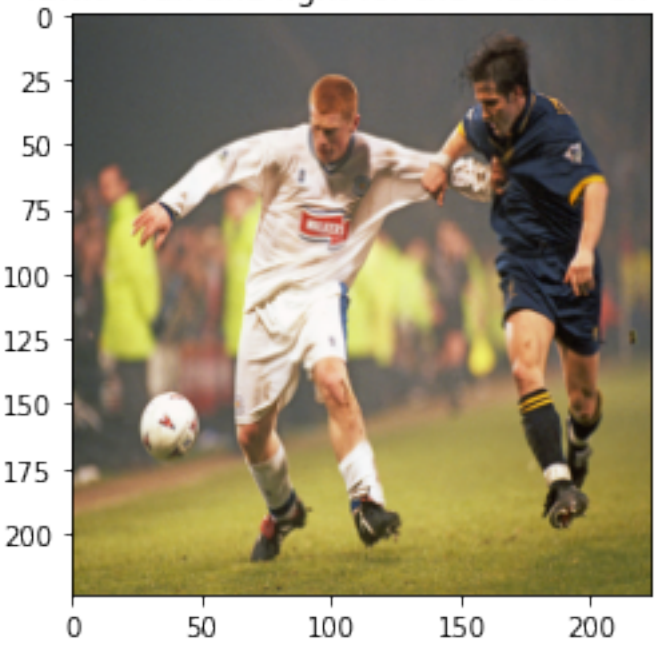
a young football player in a muddy , yellow and blue uniform carrying a football with other muddy players run behind him .



a group of police officers <UNK> the streets with a building in the background



two soccer <UNK> run and fight for the <UNK> on a grassy field .



[19]:
```
import torch
import torch.nn as nn
import torchvision.models as models
import torch.optim as optim
```

[20]:
```
# vgg16 = models.resnet50(pretrained=True)
# for param in vgg16.parameters():
#     param.requires_grad_(False)
# modules = list(vgg16.children())[:-1]
# print(dir(vgg16))
```

[21]:
```
class EncoderCNN(nn.Module):
    def __init__(self,embed_size):
        super(EncoderCNN,self).__init__()
        resnet = models.resnet50(pretrained=True)
        for param in resnet.parameters():
            param.requires_grad_(False)

        modules = list(resnet.children())[:-1]
        self.resnet = nn.Sequential(*modules)
        self.embed = nn.Linear(resnet.fc.in_features,embed_size)

    def forward(self,images):
        features = self.resnet(images)
#         print(f"resenet features shape - {features.shape}")
        features = features.view(features.size(0),-1)
#         print(f"resenet features viewed shape - {features.shape}")
        features = self.embed(features)
#         print(f"resenet features embed shape - {features.shape}")
        return features

class DecoderRNN(nn.Module):
    def␣
 ↪__init__(self,embed_size,hidden_size,vocab_size,num_layers=1,drop_prob=0.3):
        super(DecoderRNN,self).__init__()
        self.embedding = nn.Embedding(vocab_size,embed_size)
        self.lstm = nn.
 ↪LSTM(embed_size,hidden_size,num_layers=num_layers,batch_first=True)
        self.fcn = nn.Linear(hidden_size,vocab_size)
        self.drop = nn.Dropout(drop_prob)

    def forward(self,features, captions):
        # vectorize the caption
#         print(f"captions - {captions[:,:-1]}")
#         print(f"caption shape - {captions[:,:-1].shape}")
        embeds = self.embedding(captions[:,:-1])
#         print(f"shape of embeds - {embeds.shape}")
        # concat the features and captions
#         print(f"features shape - {features.shape}")
#         print(f"features unsqueeze at index 1 shape - {features.unsqueeze(1).
 ↪shape}")
        x = torch.cat((features.unsqueeze(1),embeds),dim=1)
```

```
#         print(f"shape of x - {x.shape}")
        x,_ = self.lstm(x)
#         print(f"shape of x after lstm - {x.shape}")
        x = self.fcn(x)
#         print(f"shape of x after fcn - {x.shape}")
        return x

    def generate_caption(self,inputs,hidden=None,max_len=20,vocab=None):
        # Inference part
        # Given the image features generate the captions

        batch_size = inputs.size(0)

        captions = []

        for i in range(max_len):
            output,hidden = self.lstm(inputs,hidden)
            output = self.fcn(output)
            output = output.view(batch_size,-1)


            #select the word with most val
            predicted_word_idx = output.argmax(dim=1)

            #save the generated word
            captions.append(predicted_word_idx.item())

            #end if <EOS detected>
            if vocab.itos[predicted_word_idx.item()] == "<EOS>":
                break

            #send generated word as the next caption
            inputs = self.embedding(predicted_word_idx.unsqueeze(0))

        #covert the vocab idx to words and return sentence
        return [vocab.itos[idx] for idx in captions]


class EncoderDecoder(nn.Module):
    def␣
 ↪__init__(self,embed_size,hidden_size,vocab_size,num_layers=1,drop_prob=0.3):
        super(EncoderDecoder,self).__init__()
        self.encoder = EncoderCNN(embed_size)
        self.decoder =␣
 ↪DecoderRNN(embed_size,hidden_size,vocab_size,num_layers,drop_prob)

    def forward(self, images, captions):
```

```
        features = self.encoder(images)
        outputs = self.decoder(features, captions)
        return outputs
# resenet features shape - torch.Size([4, 2048, 1, 1])
# resenet features viewed shape - torch.Size([4, 2048])
# resenet features embed shape - torch.Size([4, 400])
# caption shape - torch.Size([4, 14])
# shape of embeds - torch.Size([4, 14, 400])
# features shape - torch.Size([4, 400])
# features unsqueeze at index 1 shape - torch.Size([4, 1, 400])
# shape of x - torch.Size([4, 15, 400])
# shape of x after lstm - torch.Size([4, 15, 512])
# shape of x after fcn - torch.Size([4, 15, 2994])
```

[22]:
```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
device
```

[22]:
```
device(type='cuda')
```

[23]:
```
# Hyperparameters
embed_size = 400
hidden_size = 512
vocab_size = len(dataset.vocab)
num_layers = 2
learning_rate = 0.0001
num_epochs = 2
```

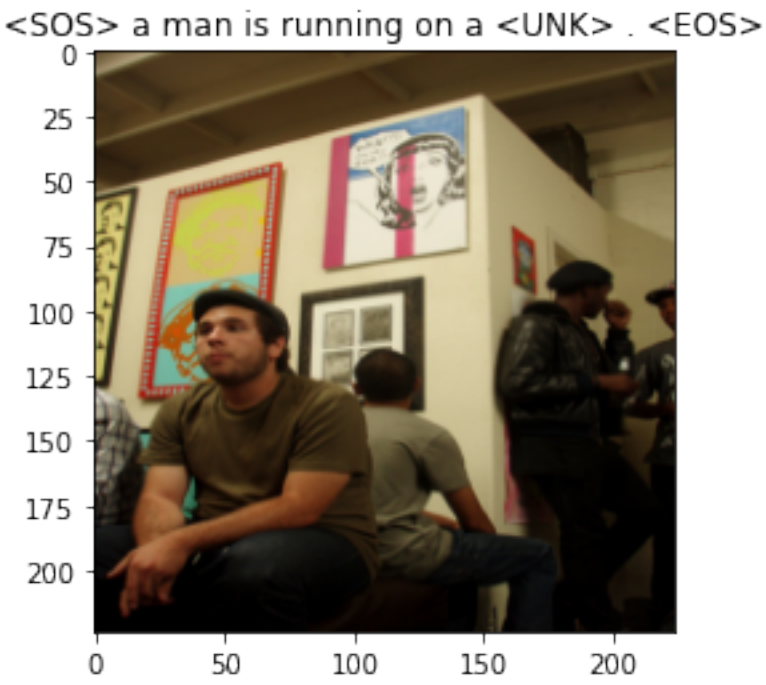
[24]:
```
# initialize model, loss etc
model = EncoderDecoder(embed_size, hidden_size, vocab_size, num_layers).
 ↪to(device)
criterion = nn.CrossEntropyLoss(ignore_index=dataset.vocab.stoi["<PAD>"])
optimizer = optim.Adam(model.parameters(), lr=learning_rate)
```

```
Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to
/root/.cache/torch/hub/checkpoints/resnet50-0676ba61.pth

  0%|          | 0.00/97.8M [00:00<?, ?B/s]
```

[25]:
```
num_epochs = 20
print_every = 2000

for epoch in range(1,num_epochs+1):
    for idx, (image, captions) in enumerate(iter(data_loader)):
        image,captions = image.to(device),captions.to(device)

        # Zero the gradients.
        optimizer.zero_grad()
```

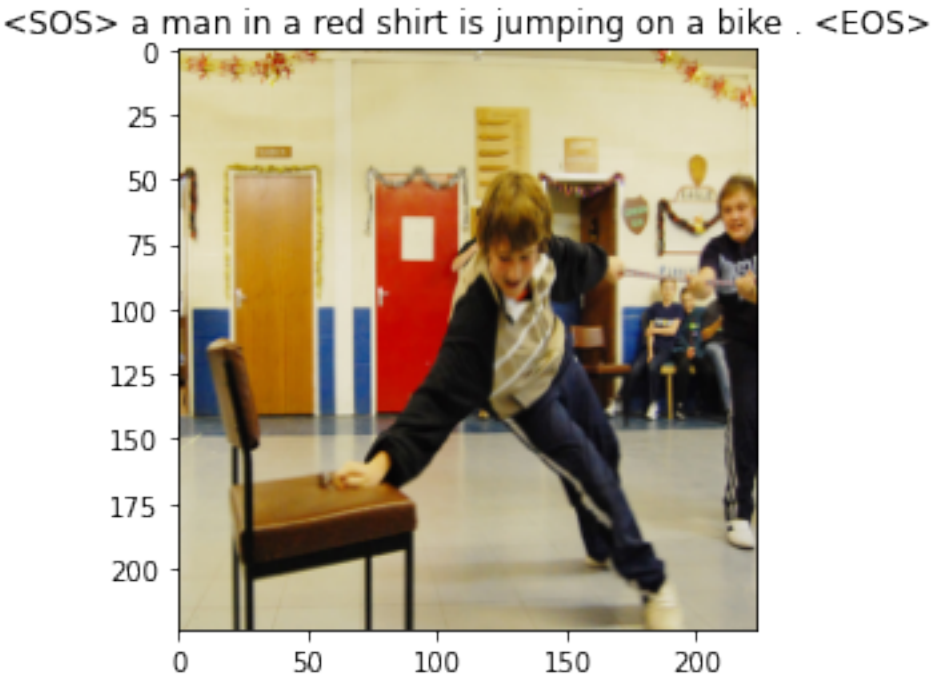
```
        # Feed forward
        outputs = model(image, captions)

        # Calculate the batch loss.
        loss = criterion(outputs.view(-1, vocab_size), captions.view(-1))


        # Backward pass.
        loss.backward()

        # Update the parameters in the optimizer.
        optimizer.step()

        if (idx+1)%print_every == 0:
            print("Epoch: {} loss: {:.5f}".format(epoch,loss.item()))


            #generate the caption
            model.eval()
            with torch.no_grad():
                dataiter = iter(data_loader)
                img,_ = next(dataiter)
                features = model.encoder(img[0:1].to(device))
                print(f"features shape - {features.shape}")
                caps = model.decoder.generate_caption(features.
↪unsqueeze(0),vocab=dataset.vocab)
                caption = ' '.join(caps)
                print(caption)
                show_image(img[0],title=caption)

            model.train()
```

```
Epoch: 1 loss: 4.09961
features shape - torch.Size([1, 400])
<SOS> a man is running on a <UNK> . <EOS>
```

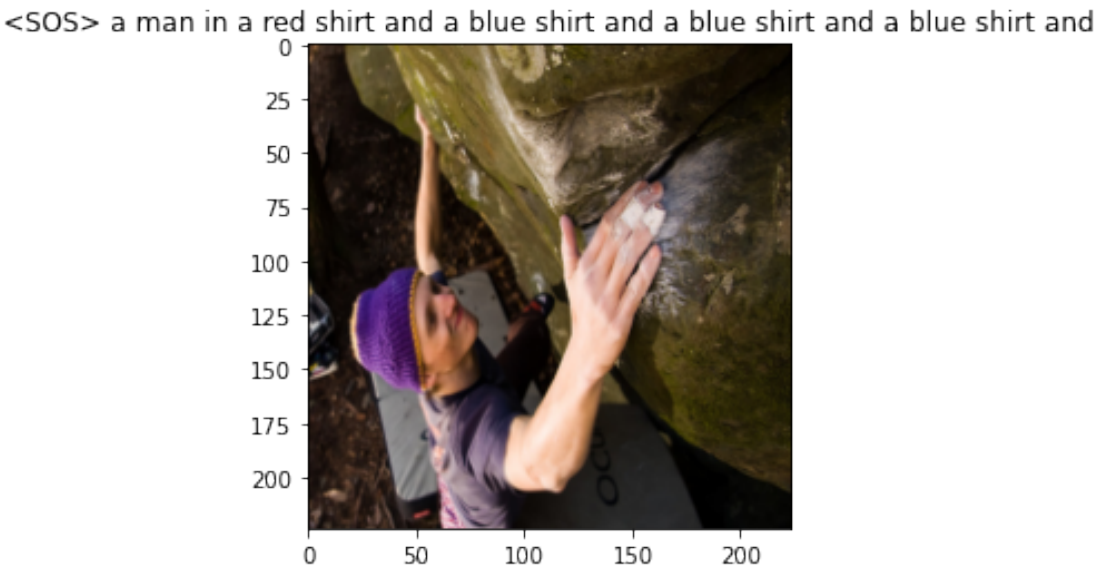



```
Epoch: 1 loss: 3.32659
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt is running on a field . <EOS>
```

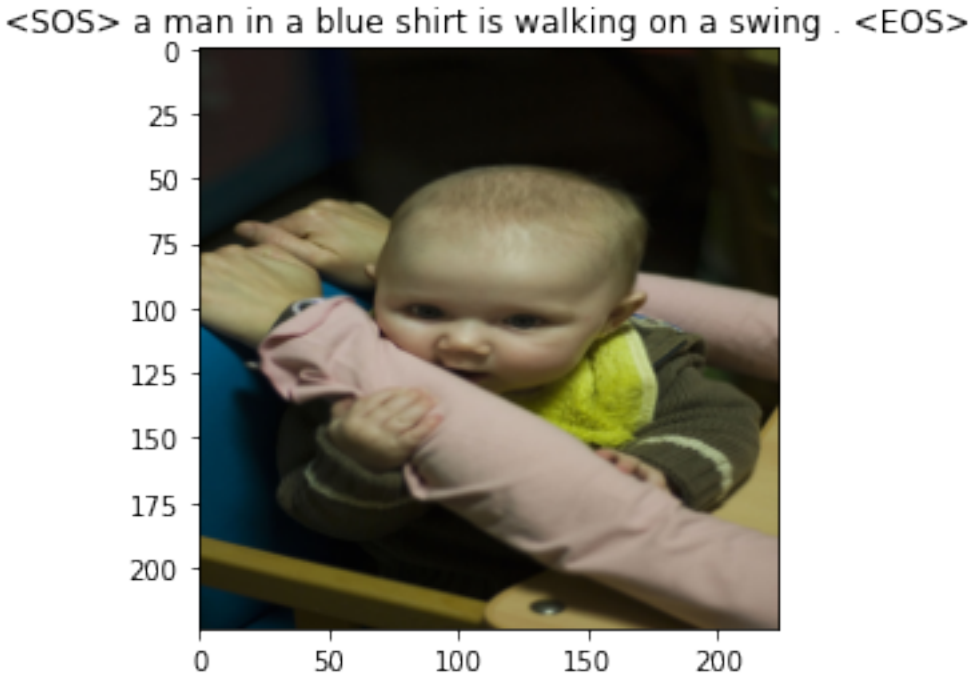
```
Epoch: 1 loss: 3.10428
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt is jumping on a bike . <EOS>
```




```
Epoch: 1 loss: 2.86790
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt and a blue shirt and a blue shirt and a blue shirt
and
```

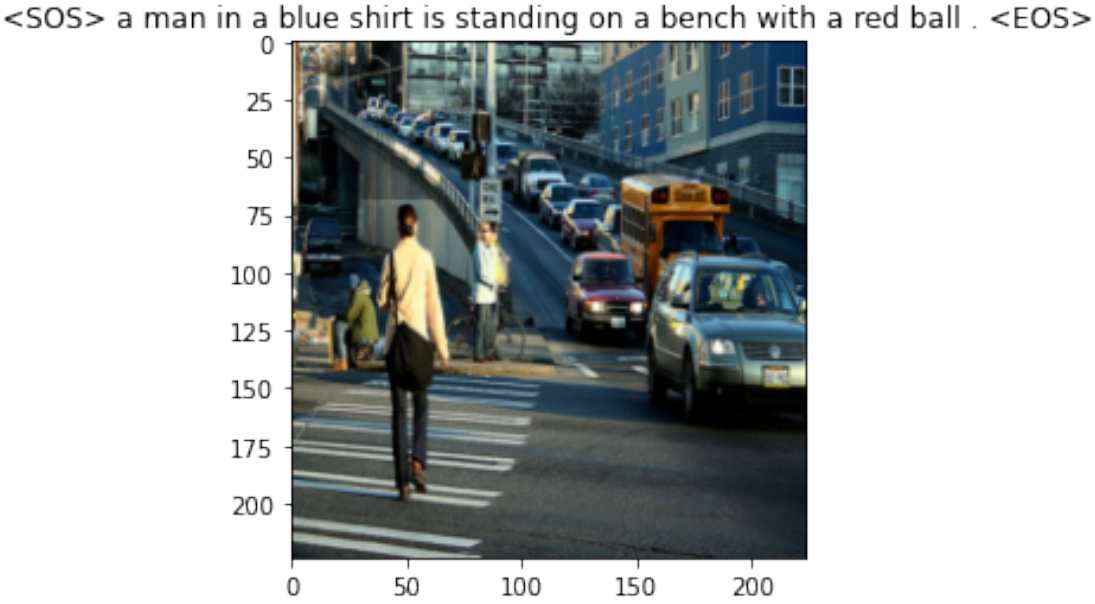



```
Epoch: 1 loss: 3.07261
features shape - torch.Size([1, 400])
<SOS> a man in a blue shirt is walking on a swing . <EOS>
```

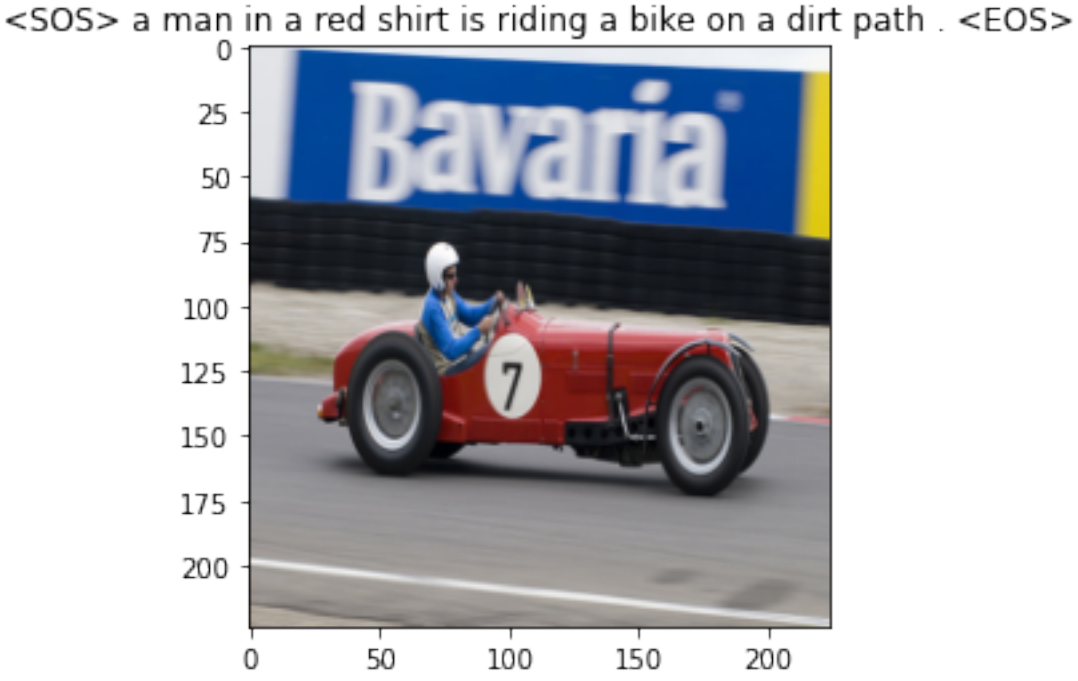



```
Epoch: 2 loss: 2.57285
features shape - torch.Size([1, 400])
```

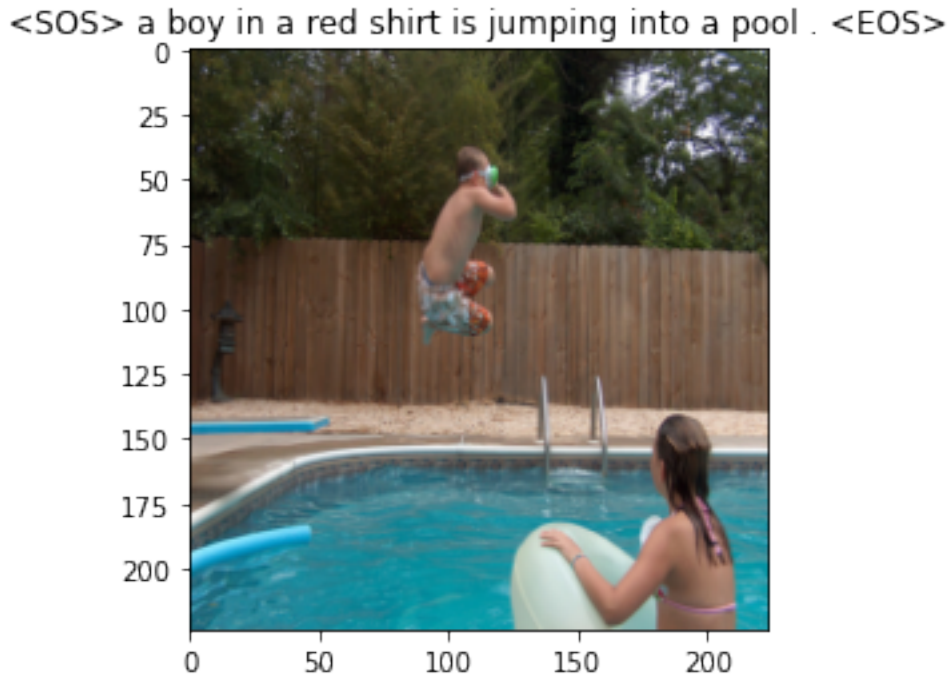
<SOS> a man in a blue shirt is standing on a bench with a red ball . <EOS>



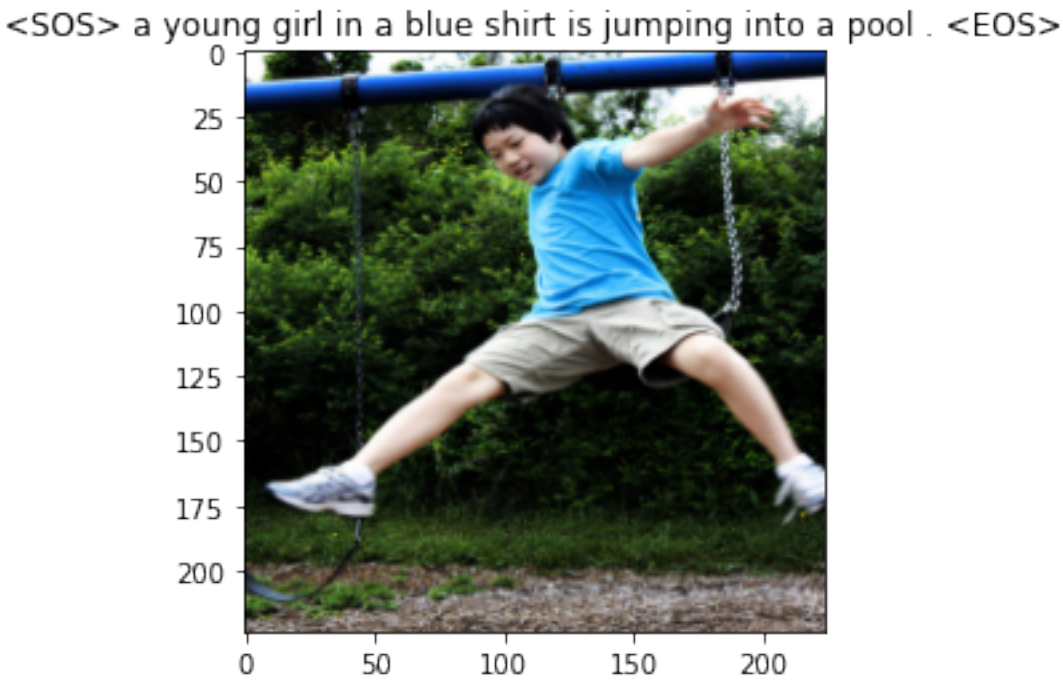
Epoch: 2 loss: 2.68556
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt is riding a bike on a dirt path . <EOS>



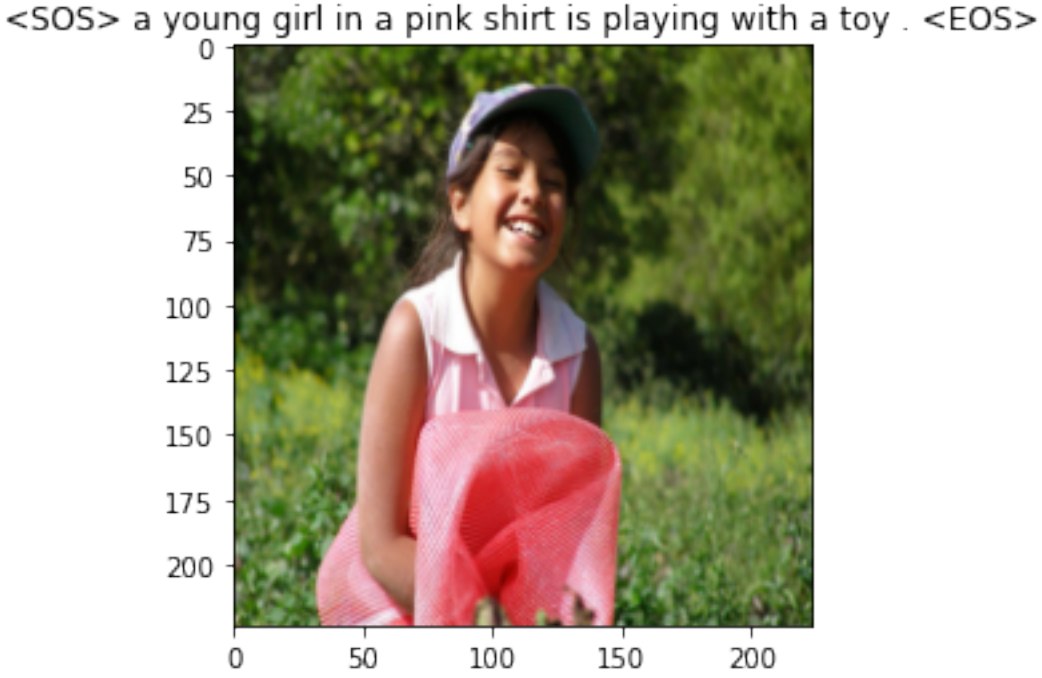
```
Epoch: 2 loss: 3.01316
features shape - torch.Size([1, 400])
<SOS> a boy in a red shirt is jumping into a pool . <EOS>
```



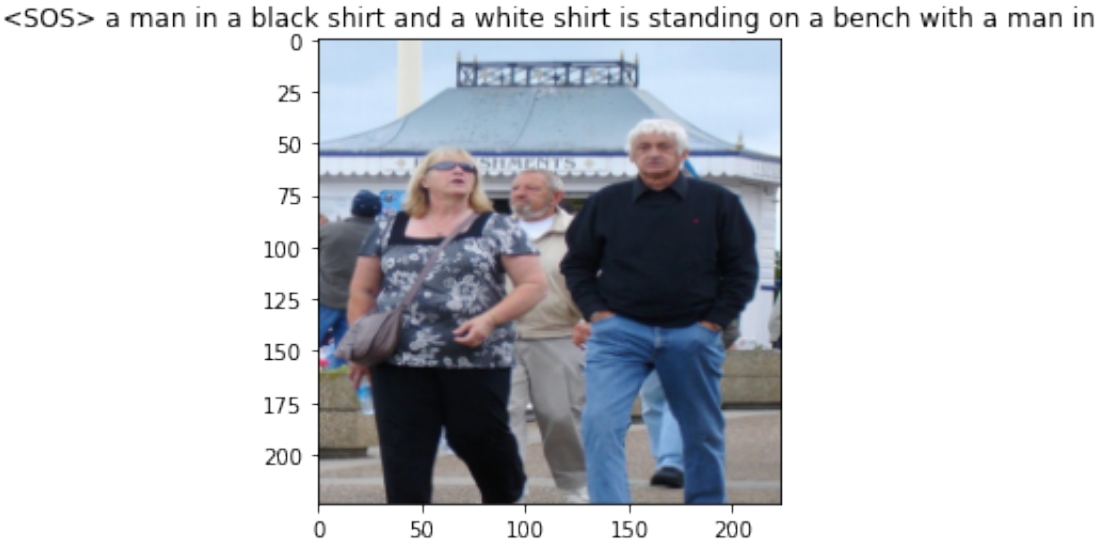

```
Epoch: 2 loss: 2.62387
features shape - torch.Size([1, 400])
<SOS> a young girl in a blue shirt is jumping into a pool . <EOS>
```



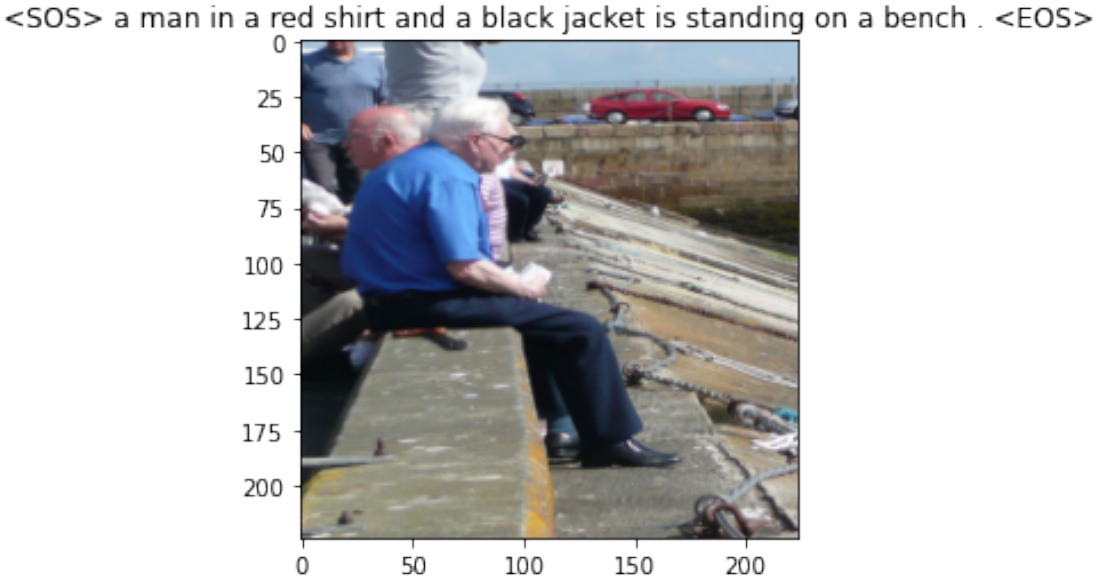

```
Epoch: 2 loss: 2.26696
features shape - torch.Size([1, 400])
<SOS> a young girl in a pink shirt is playing with a toy . <EOS>
```



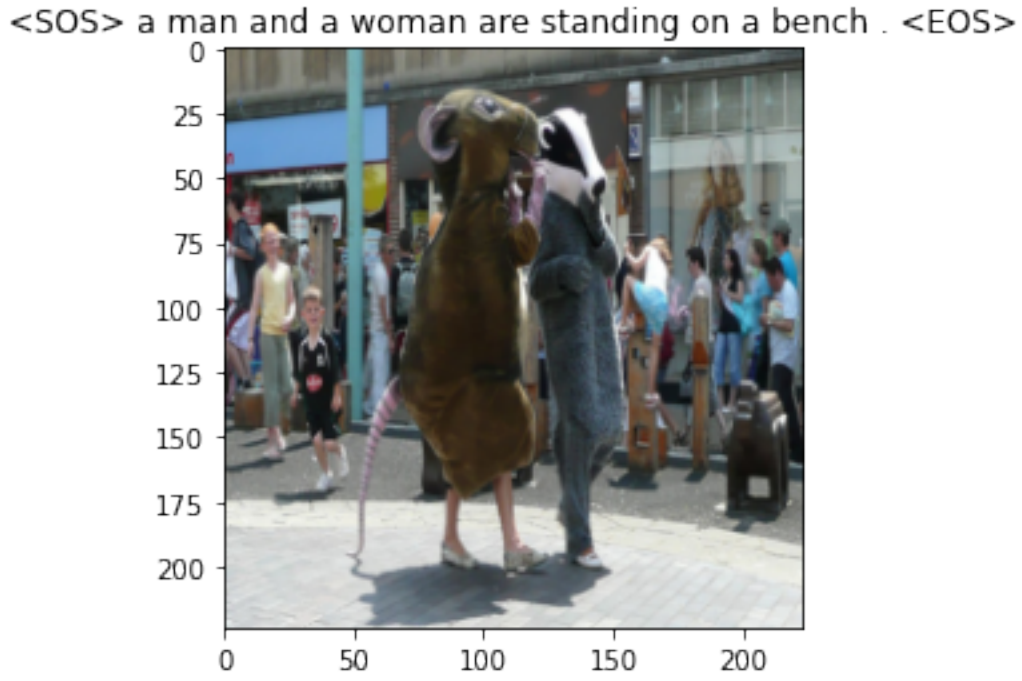
```
Epoch: 3 loss: 2.56823
features shape - torch.Size([1, 400])
<SOS> a man in a black shirt and a white shirt is standing on a bench with a man
in
```



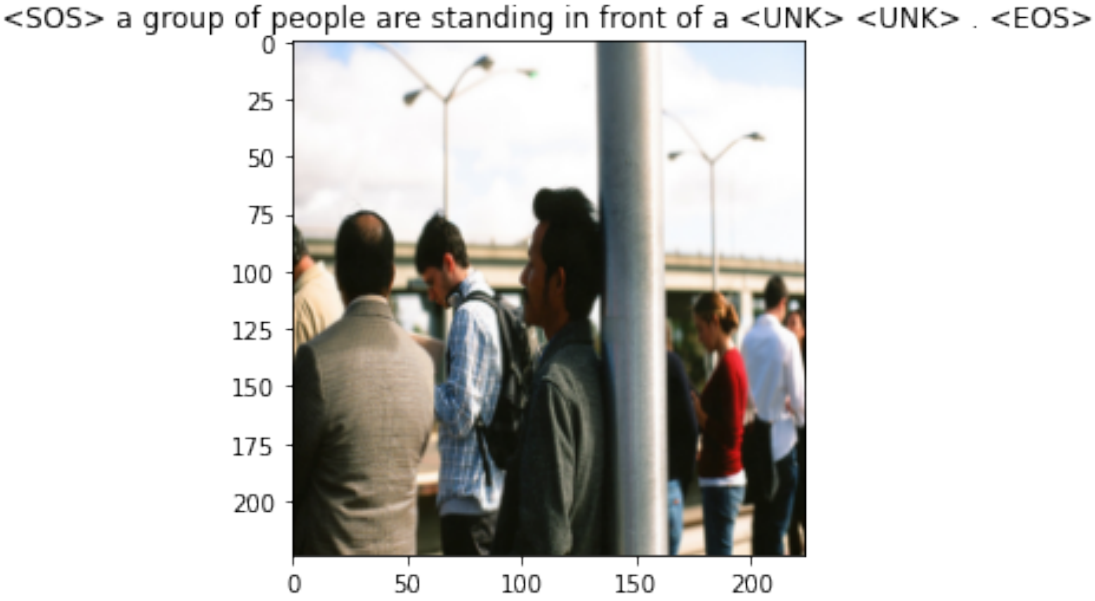

```
Epoch: 3 loss: 2.50525
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt and a black jacket is standing on a bench . <EOS>
```



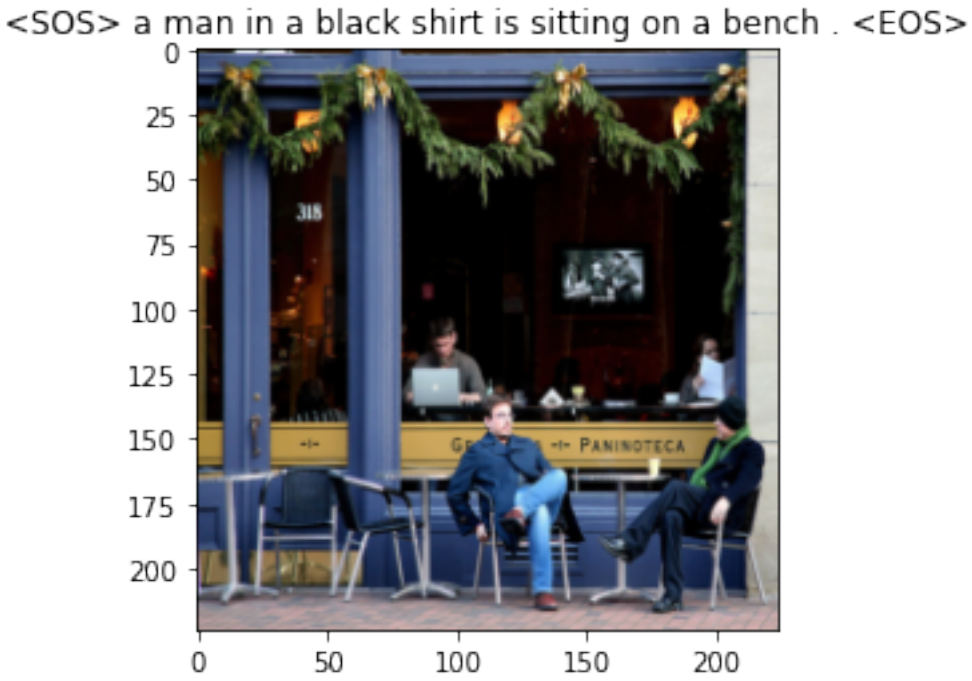
```
Epoch: 3 loss: 2.44488
features shape - torch.Size([1, 400])
<SOS> a man and a woman are standing on a bench . <EOS>
```




```
Epoch: 3 loss: 3.01704
features shape - torch.Size([1, 400])
<SOS> a group of people are standing in front of a <UNK> <UNK> . <EOS>
```

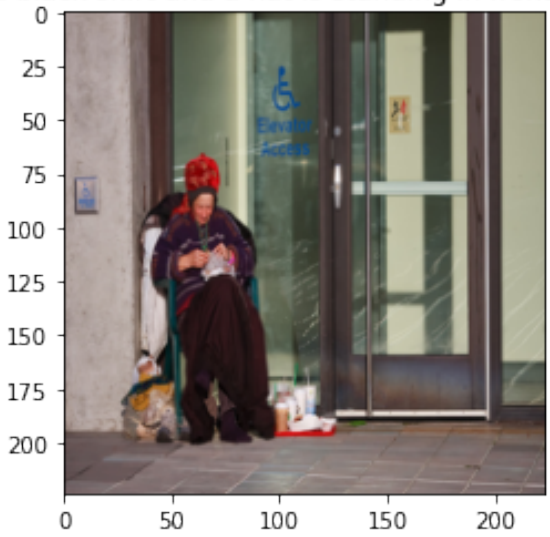


```
Epoch: 3 loss: 2.01231
features shape - torch.Size([1, 400])
<SOS> a man in a black shirt is sitting on a bench . <EOS>
```



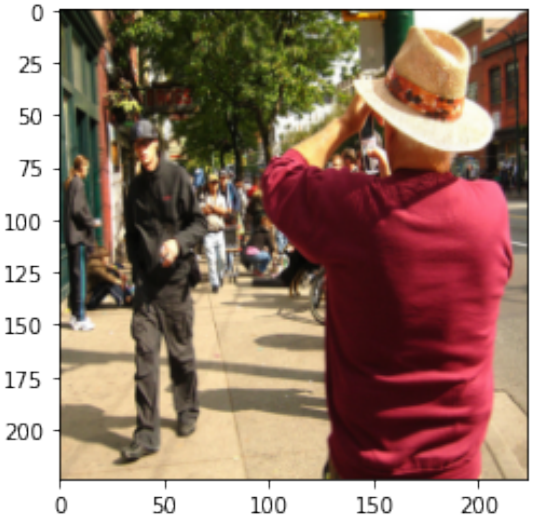

```
Epoch: 4 loss: 2.43919
features shape - torch.Size([1, 400])
<SOS> a man in a black shirt and a hat is standing in front of a crowd of people
.
```

<SOS> a man in a black shirt and a hat is standing in front of a crowd of people .

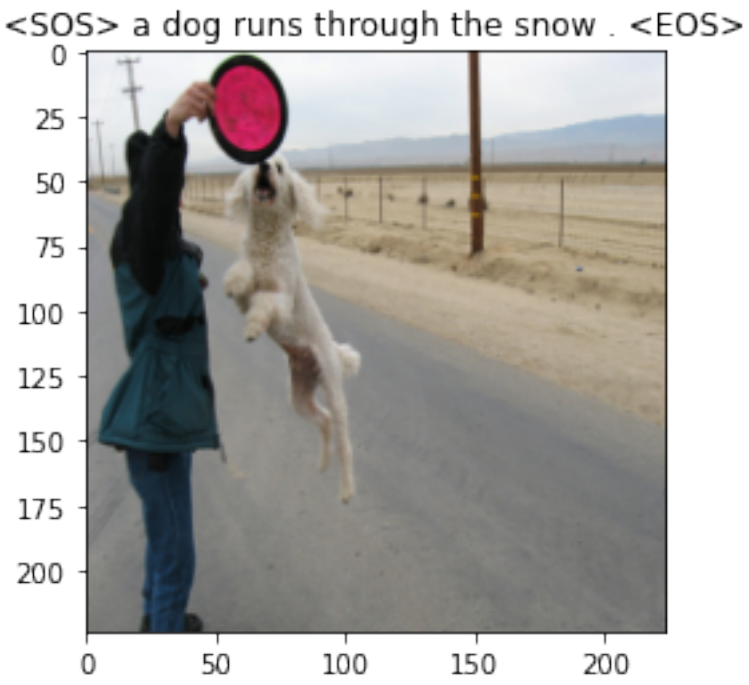



```
Epoch: 4 loss: 1.90217
features shape - torch.Size([1, 400])
<SOS> a man in a black shirt and a woman in a black shirt and a woman in a black
```

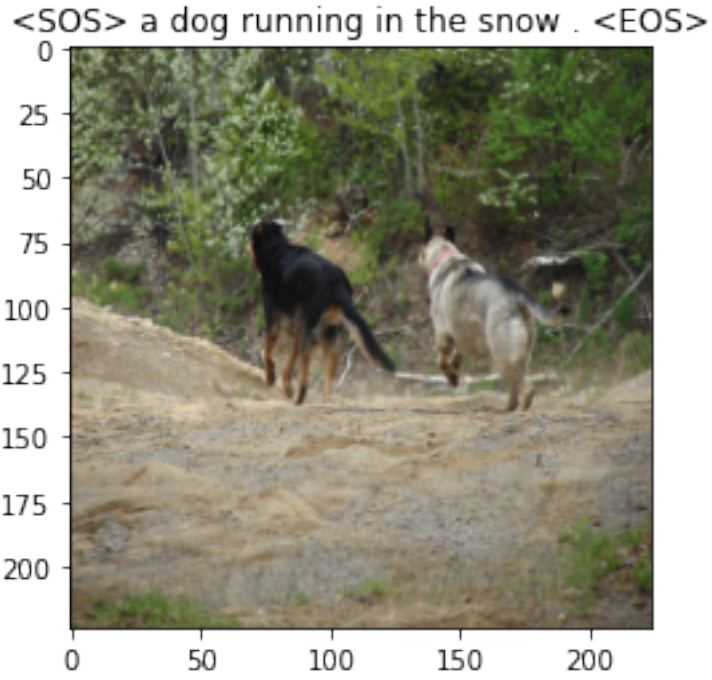
<SOS> a man in a black shirt and a woman in a black shirt and a woman in a black



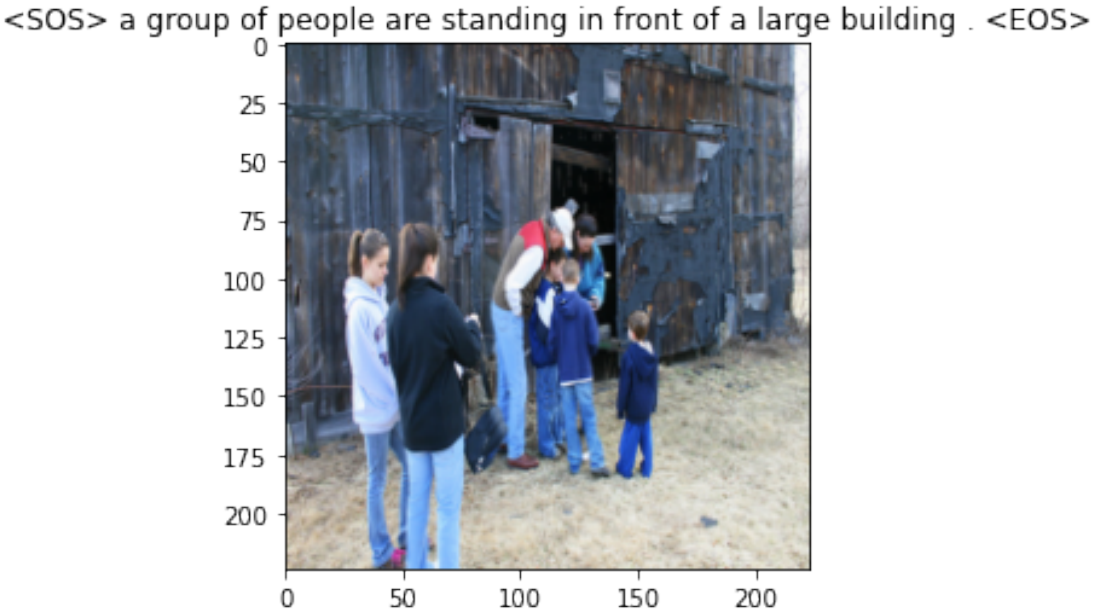

```
Epoch: 4 loss: 2.03861
features shape - torch.Size([1, 400])
<SOS> a dog runs through the snow . <EOS>
```



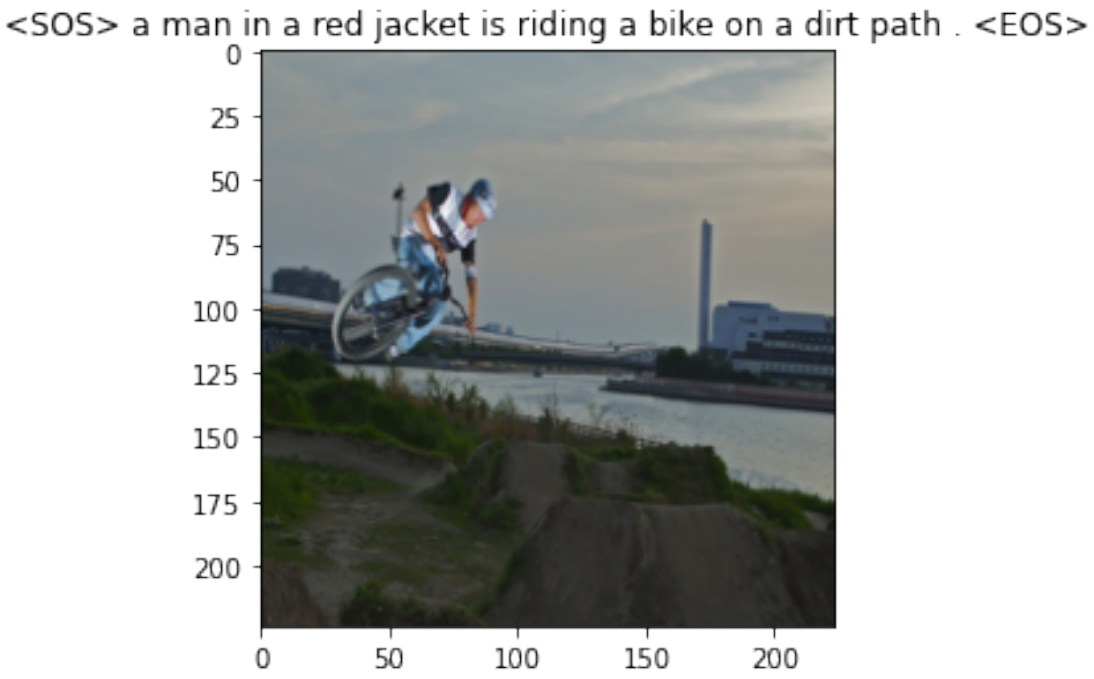

```
Epoch: 4 loss: 2.17918
features shape - torch.Size([1, 400])
<SOS> a dog running in the snow . <EOS>
```



```
Epoch: 4 loss: 1.69424
features shape - torch.Size([1, 400])
<SOS> a group of people are standing in front of a large building . <EOS>
```

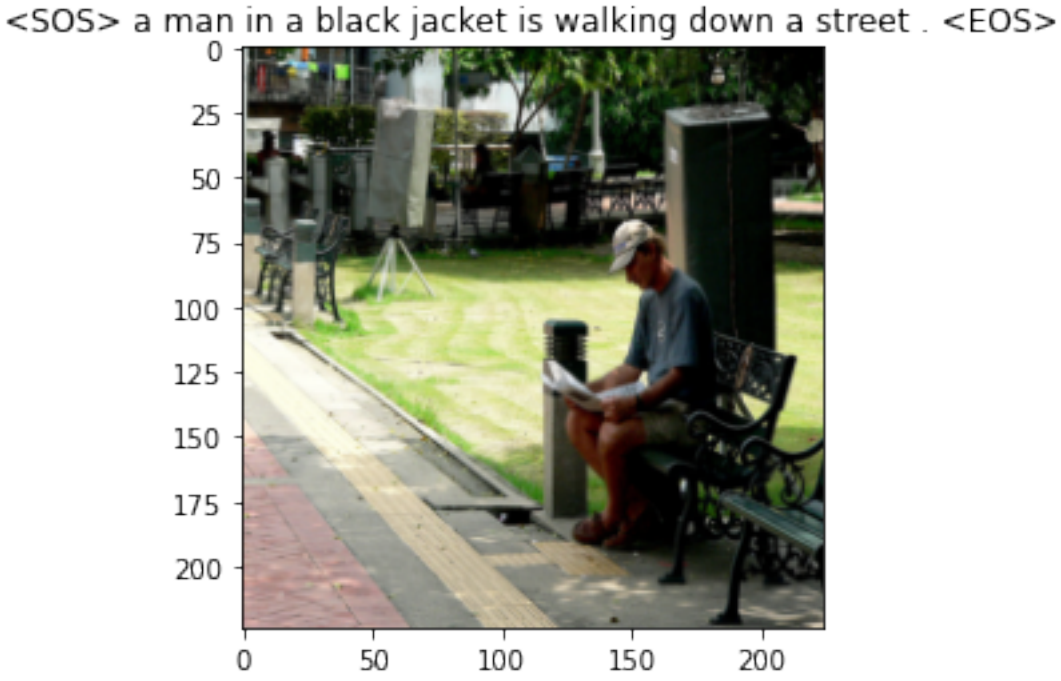



```
Epoch: 5 loss: 2.85185
features shape - torch.Size([1, 400])
<SOS> a man in a red jacket is riding a bike on a dirt path . <EOS>
```

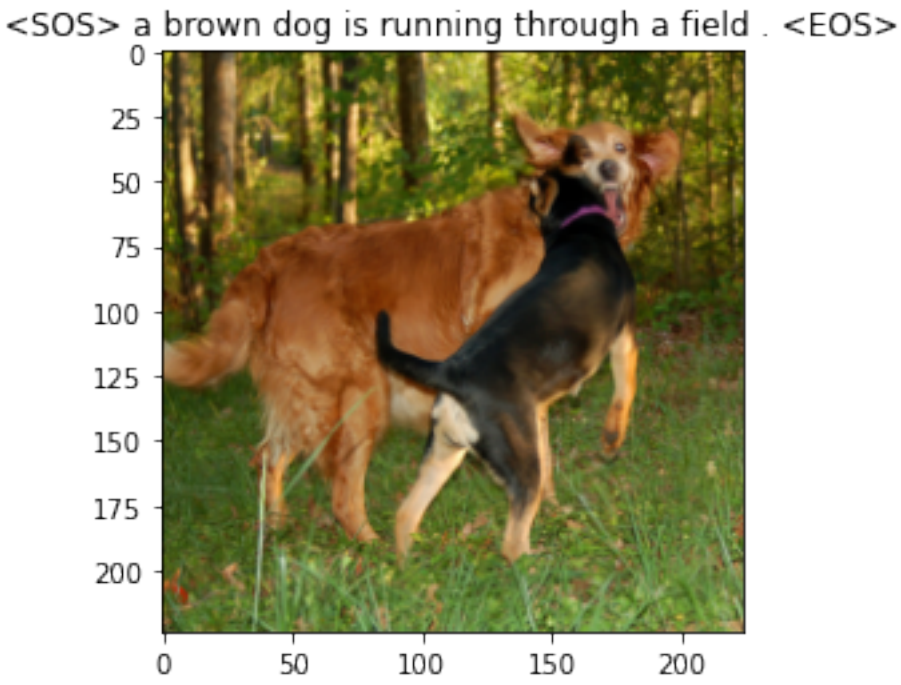



```
Epoch: 5 loss: 2.48593
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt is riding a bike on a track . <EOS>
```

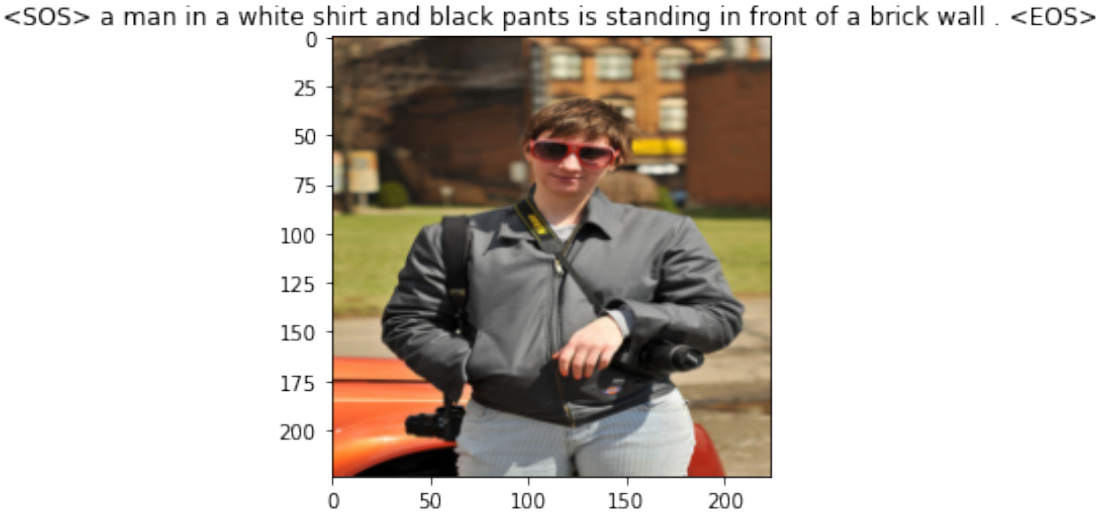
```
Epoch: 5 loss: 2.16536
features shape - torch.Size([1, 400])
<SOS> a man in a black jacket is walking down a street . <EOS>
```




```
Epoch: 5 loss: 1.94676
features shape - torch.Size([1, 400])
<SOS> a brown dog is running through a field . <EOS>
```

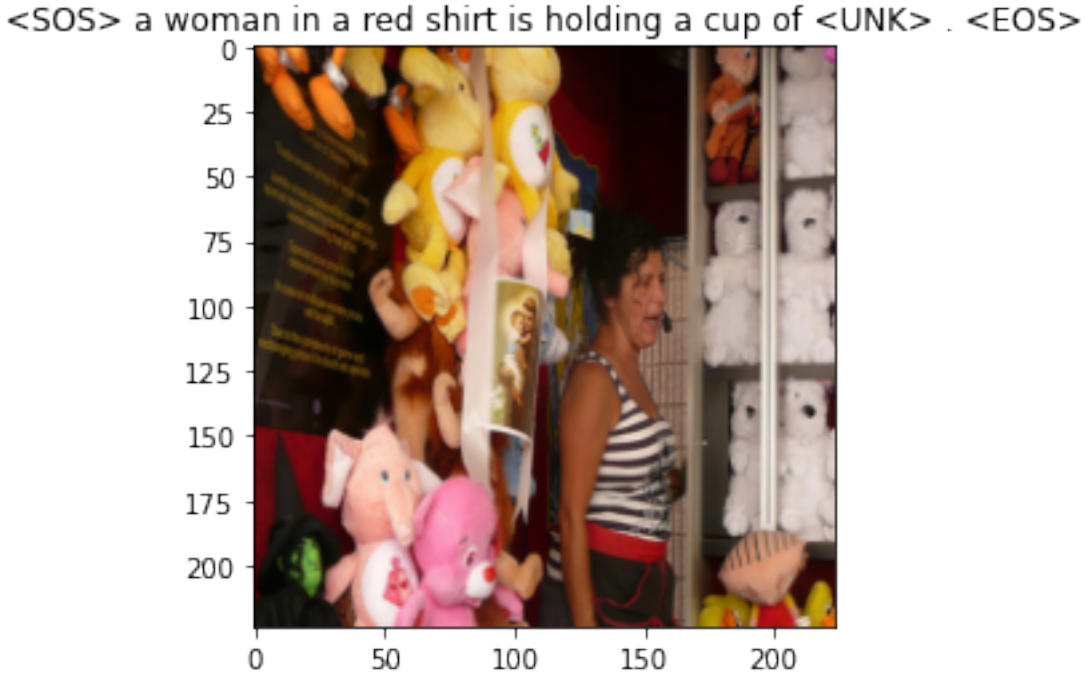



```
Epoch: 5 loss: 1.84172
features shape - torch.Size([1, 400])
<SOS> a man in a white shirt and black pants is standing in front of a brick
wall . <EOS>
```

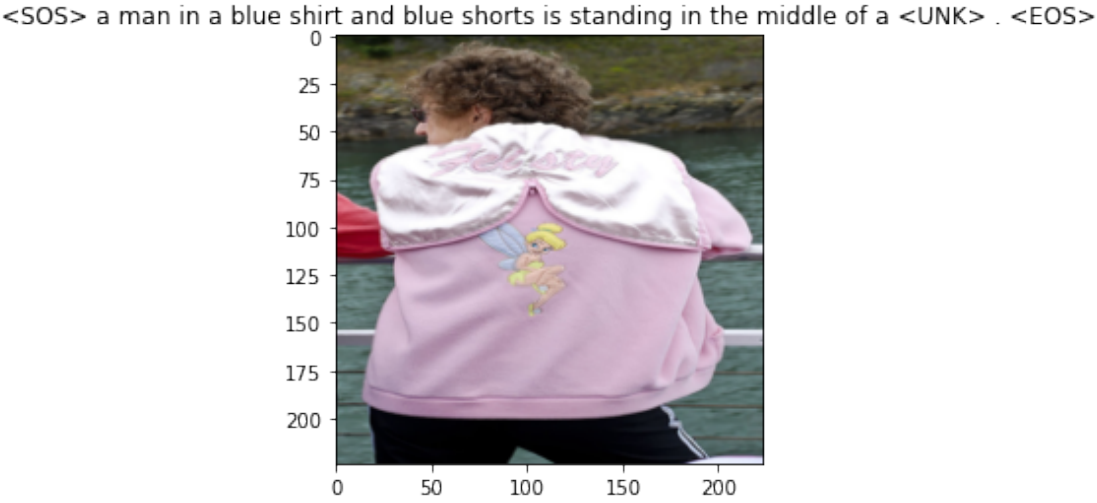



```
Epoch: 6 loss: 2.51889
features shape - torch.Size([1, 400])
```

<SOS> a woman in a red shirt is holding a cup of <UNK> . <EOS>

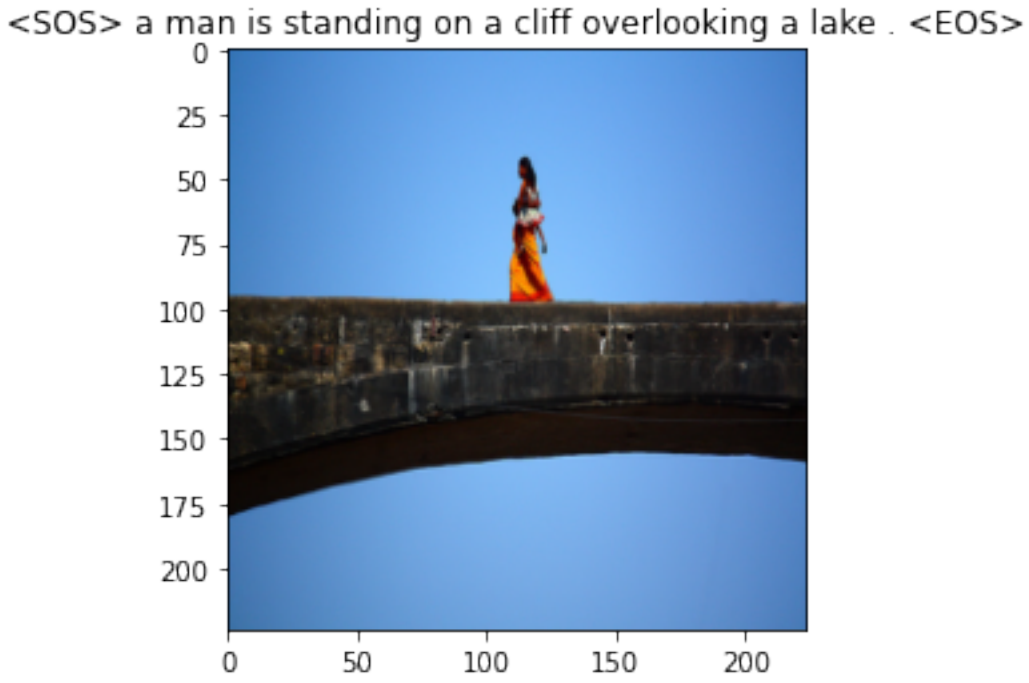


Epoch: 6 loss: 2.26482
features shape - torch.Size([1, 400])
<SOS> a man in a blue shirt and blue shorts is standing in the middle of a <UNK> . <EOS>

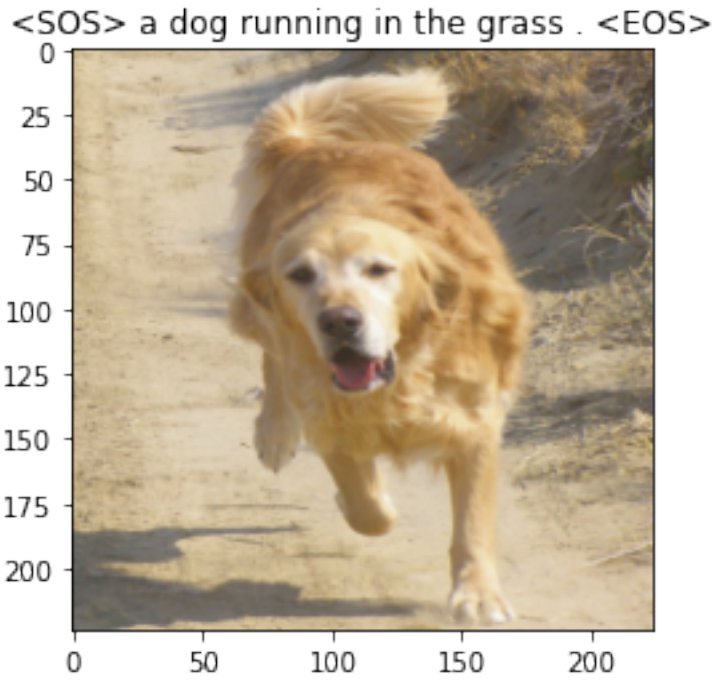


Epoch: 6 loss: 2.31131

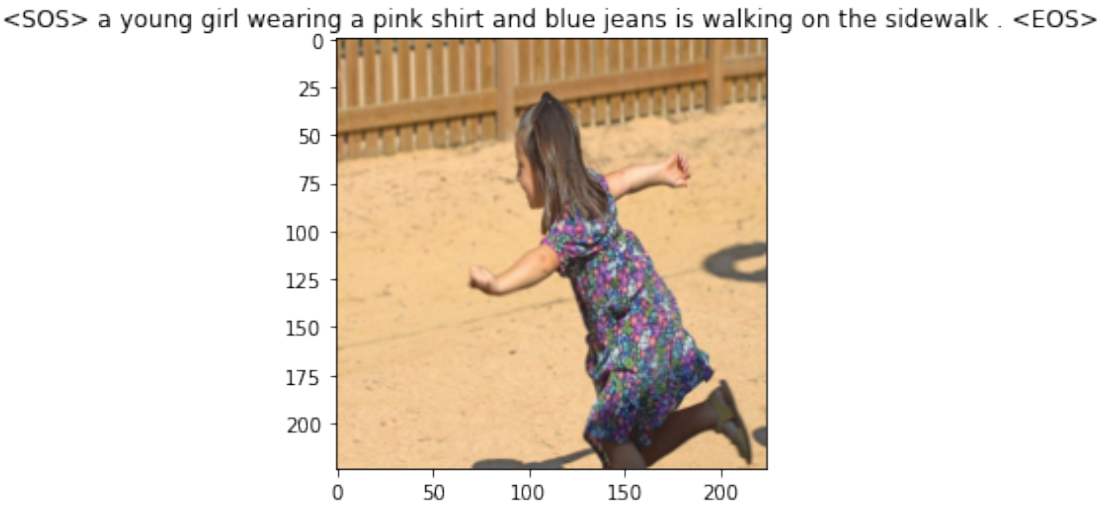
```
features shape - torch.Size([1, 400])
<SOS> a man is standing on a cliff overlooking a lake . <EOS>
```



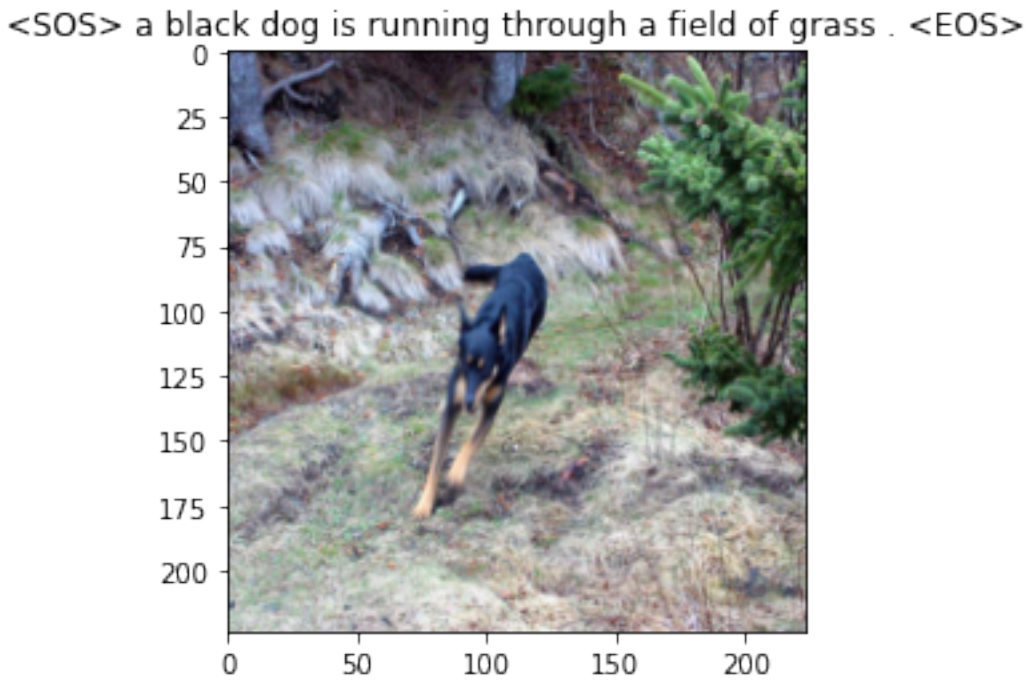

```
Epoch: 6 loss: 1.90300
features shape - torch.Size([1, 400])
<SOS> a dog running in the grass . <EOS>
```



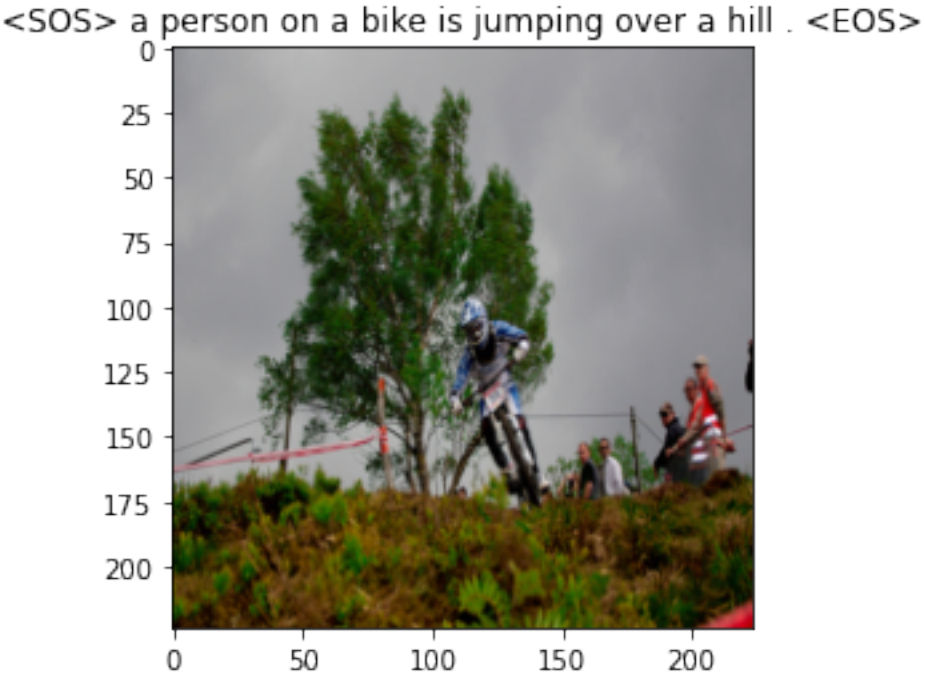

```
Epoch: 6 loss: 2.26018
features shape - torch.Size([1, 400])
<SOS> a young girl wearing a pink shirt and blue jeans is walking on the
sidewalk . <EOS>
```



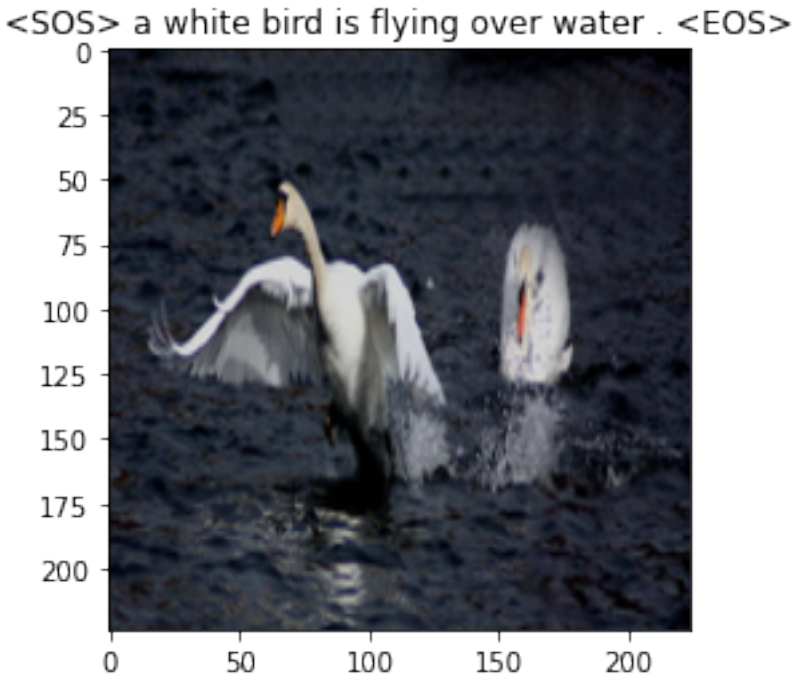

```
Epoch: 7 loss: 1.61253
features shape - torch.Size([1, 400])
<SOS> a black dog is running through a field of grass . <EOS>
```




```
Epoch: 7 loss: 1.82638
features shape - torch.Size([1, 400])
<SOS> a person on a bike is jumping over a hill . <EOS>
```

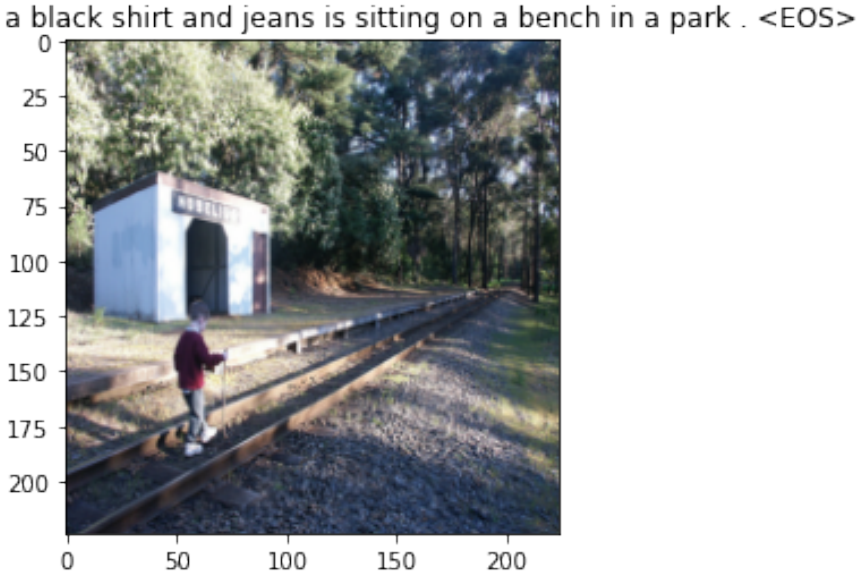


```
Epoch: 7 loss: 1.97932
features shape - torch.Size([1, 400])
<SOS> a white bird is flying over water . <EOS>
```



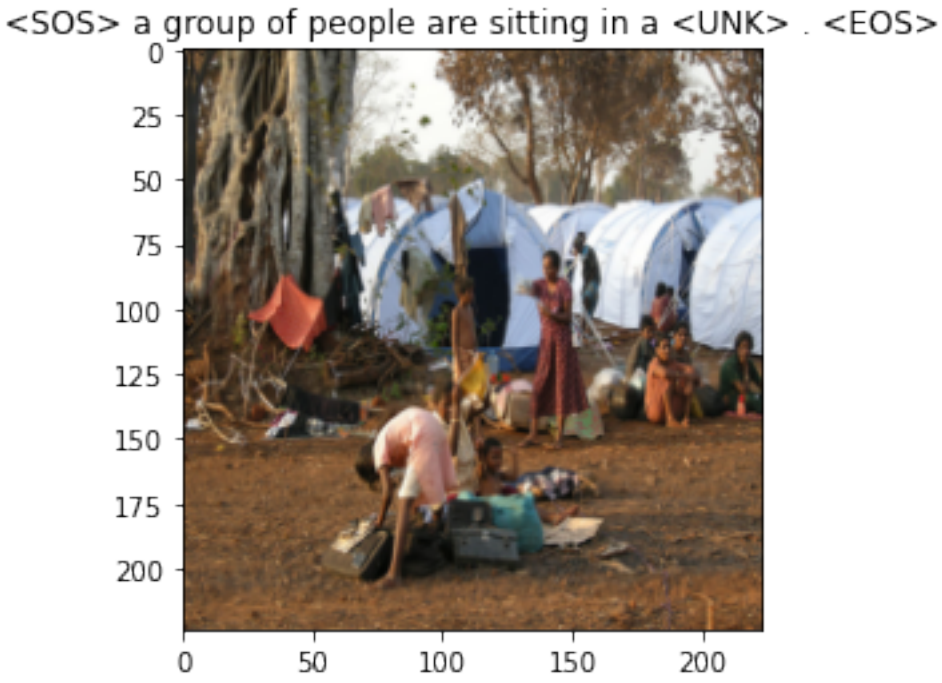

```
Epoch: 7 loss: 2.29445
features shape - torch.Size([1, 400])
<SOS> a black and white dog is running on the grass . <EOS>
```


```
Epoch: 7 loss: 1.71076
features shape - torch.Size([1, 400])
<SOS> a man in a black shirt and jeans is sitting on a bench in a park . <EOS>
```

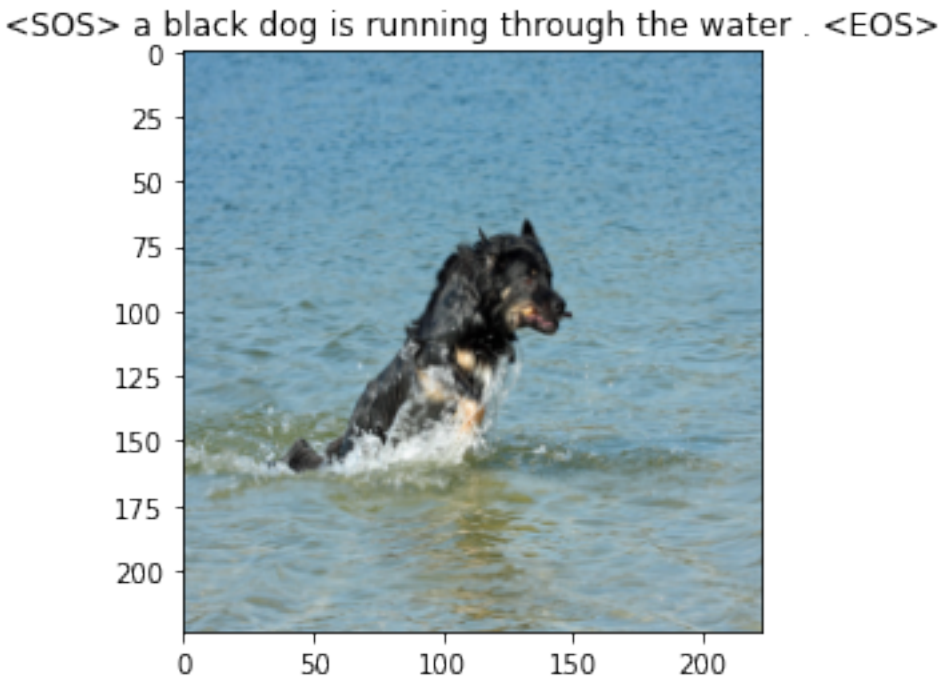



```
Epoch: 8 loss: 2.04997
```

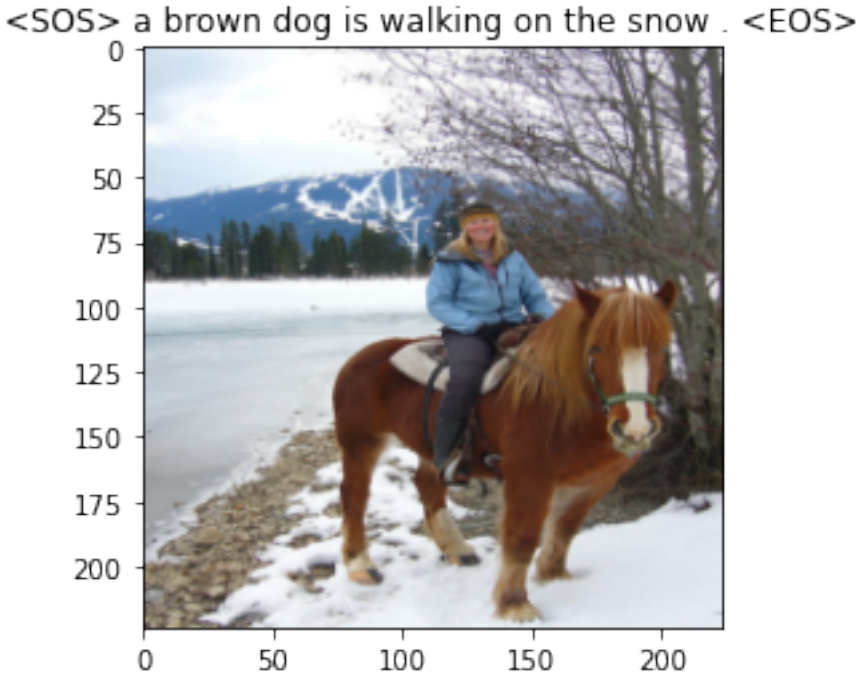
```
features shape - torch.Size([1, 400])
<SOS> a group of people are sitting in a <UNK> . <EOS>
```



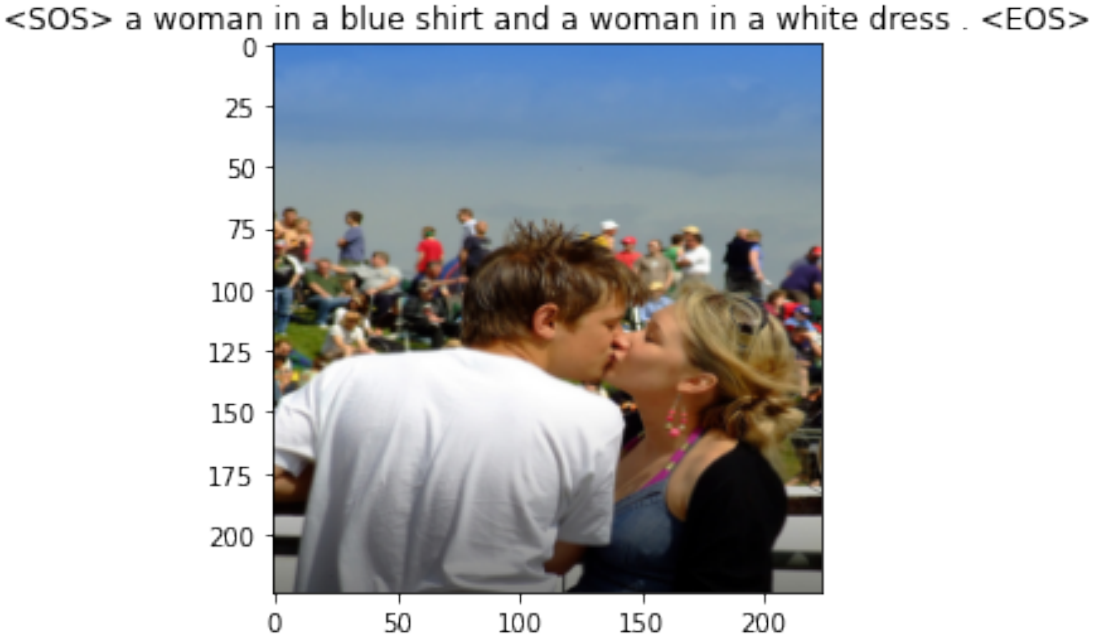

```
Epoch: 8 loss: 2.47491
features shape - torch.Size([1, 400])
<SOS> a black dog is running through the water . <EOS>
```



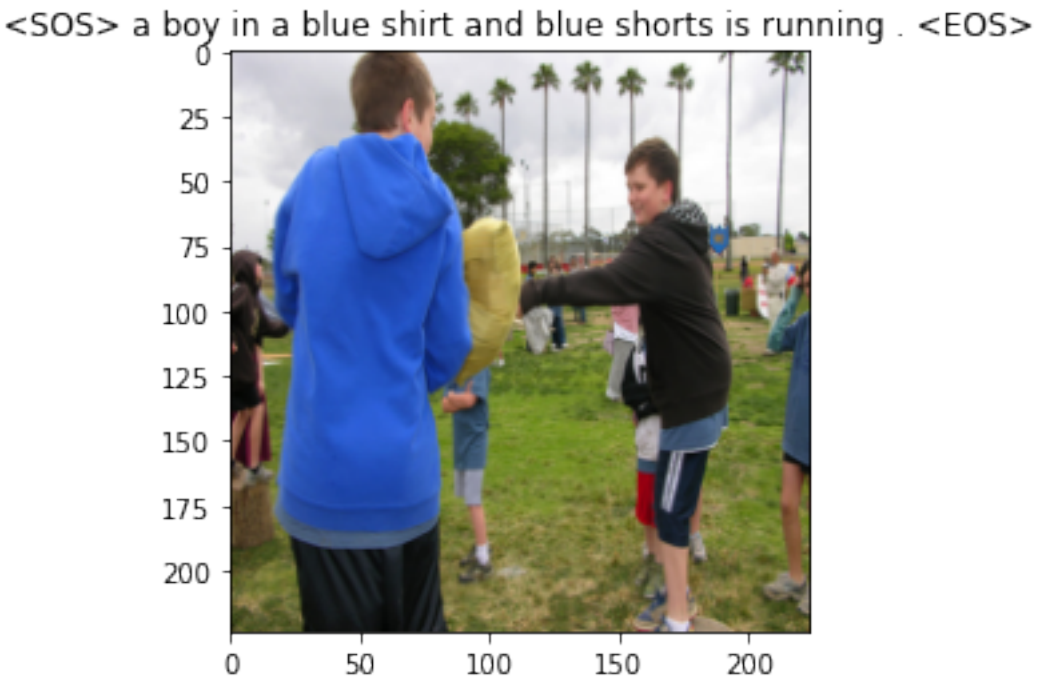

```
Epoch: 8 loss: 1.87362
features shape - torch.Size([1, 400])
<SOS> a brown dog is walking on the snow . <EOS>
```



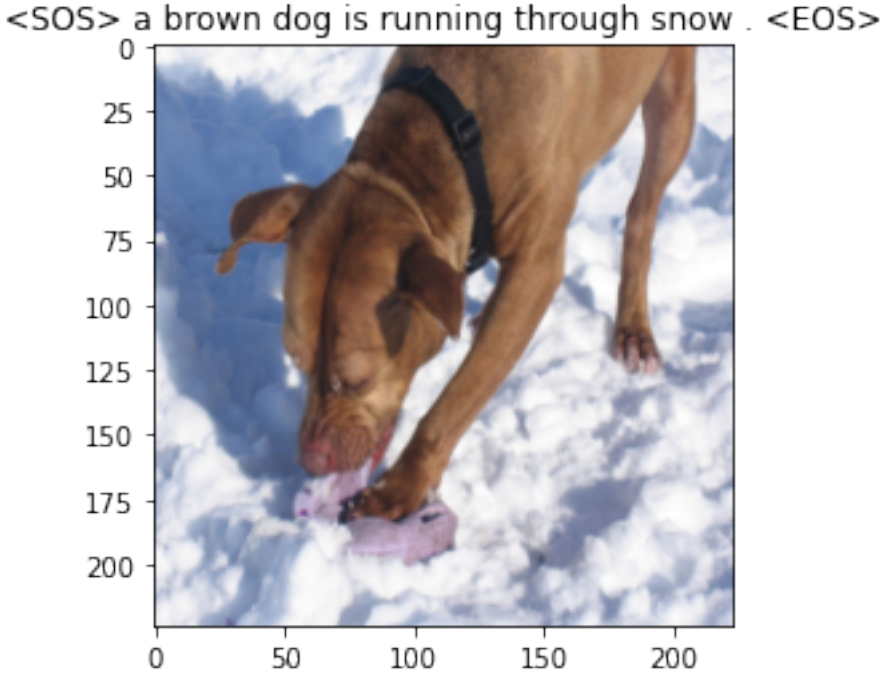
```
Epoch: 8 loss: 2.07622
features shape - torch.Size([1, 400])
<SOS> a woman in a blue shirt and a woman in a white dress . <EOS>
```



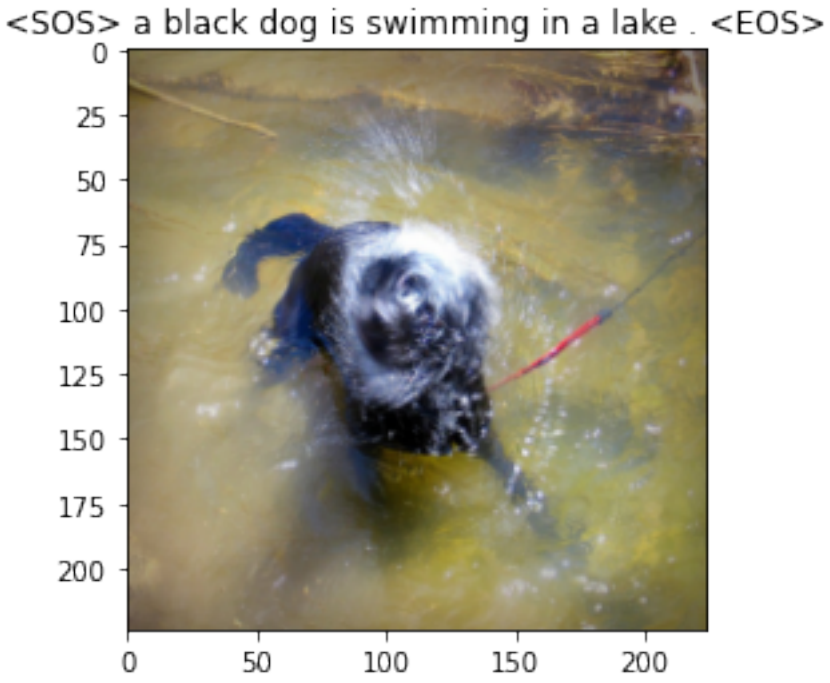

```
Epoch: 8 loss: 1.78076
features shape - torch.Size([1, 400])
<SOS> a boy in a blue shirt and blue shorts is running . <EOS>
```



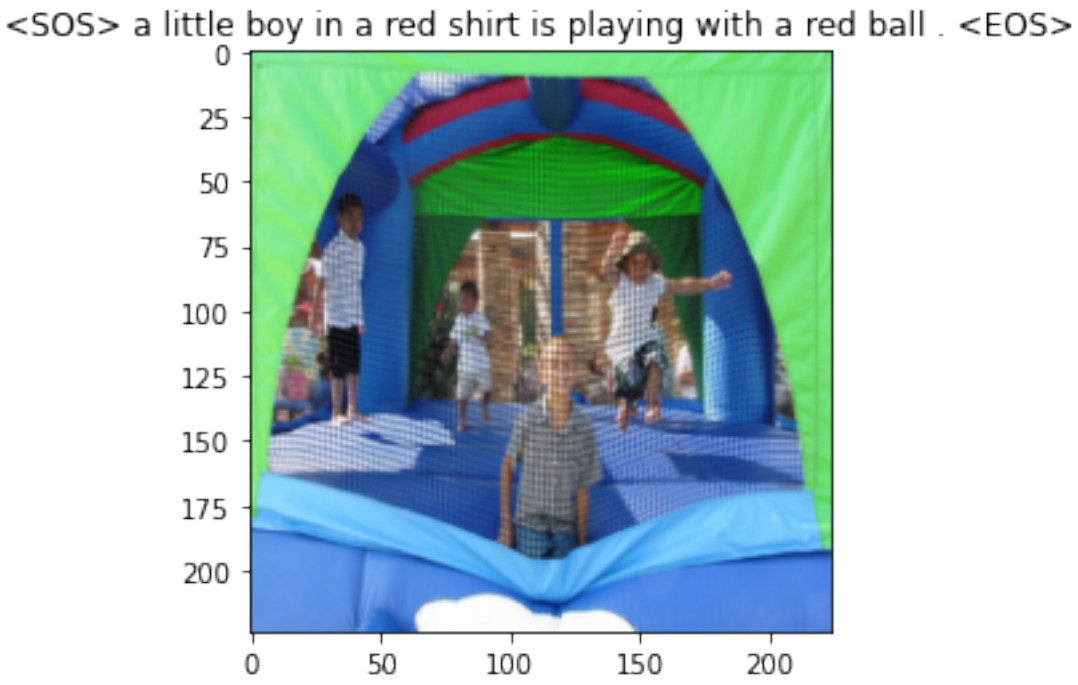

```
Epoch: 9 loss: 1.85543
features shape - torch.Size([1, 400])
<SOS> a brown dog is running through snow . <EOS>
```



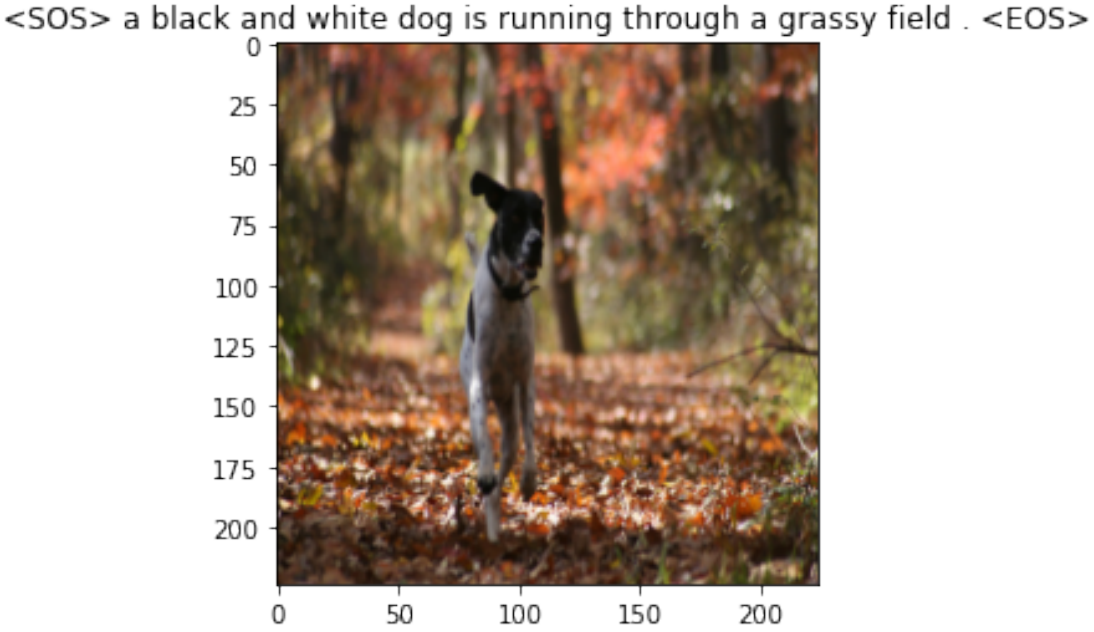
```
Epoch: 9 loss: 1.83989
features shape - torch.Size([1, 400])
<SOS> a black dog is swimming in a lake . <EOS>
```



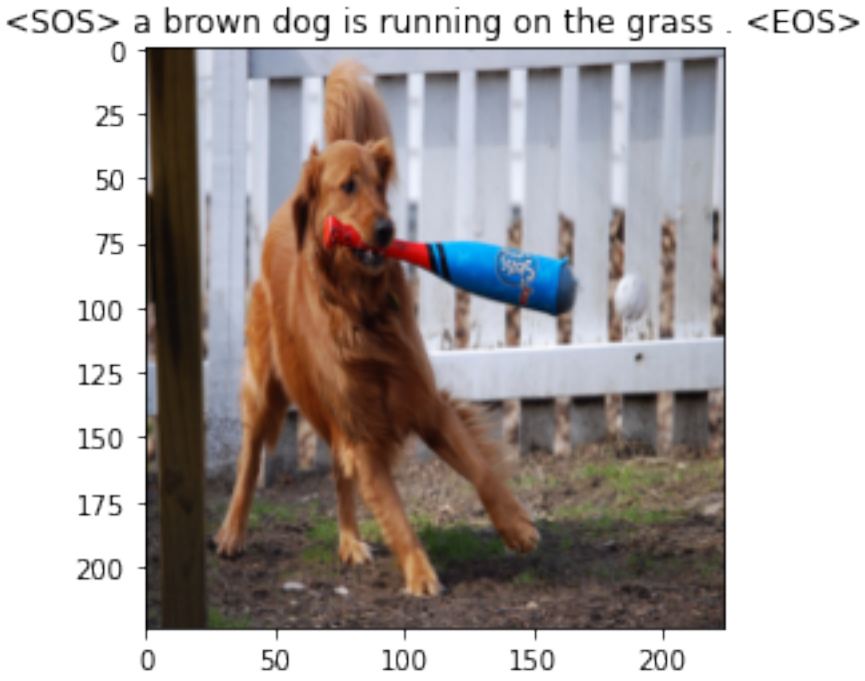

```
Epoch: 9 loss: 1.59633
features shape - torch.Size([1, 400])
<SOS> a little boy in a red shirt is playing with a red ball . <EOS>
```



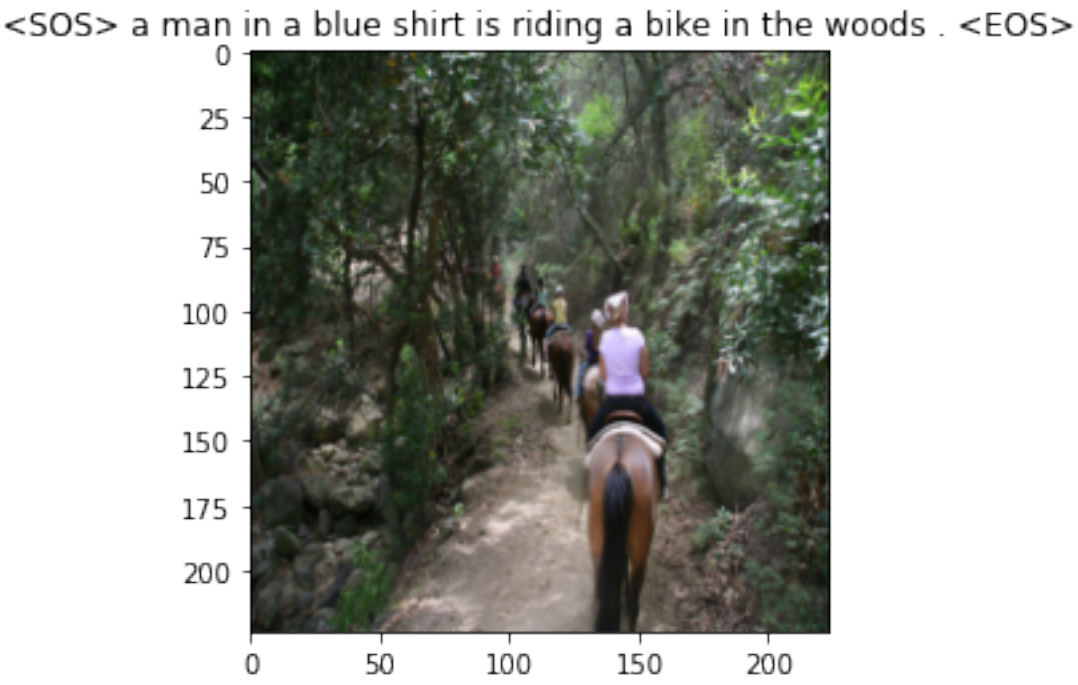

```
Epoch: 9 loss: 2.01517
features shape - torch.Size([1, 400])
<SOS> a black and white dog is running through a grassy field . <EOS>
```



```
Epoch: 9 loss: 1.81744
features shape - torch.Size([1, 400])
<SOS> a brown dog is running on the grass . <EOS>
```

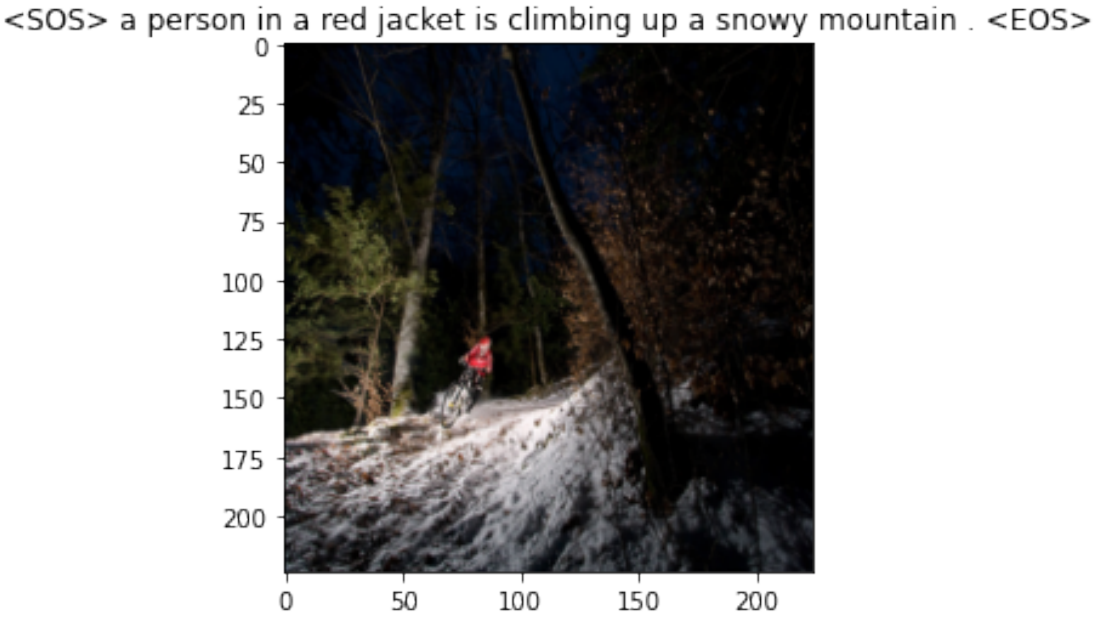



```
Epoch: 10 loss: 1.45585
features shape - torch.Size([1, 400])
<SOS> a man in a blue shirt is riding a bike in the woods . <EOS>
```

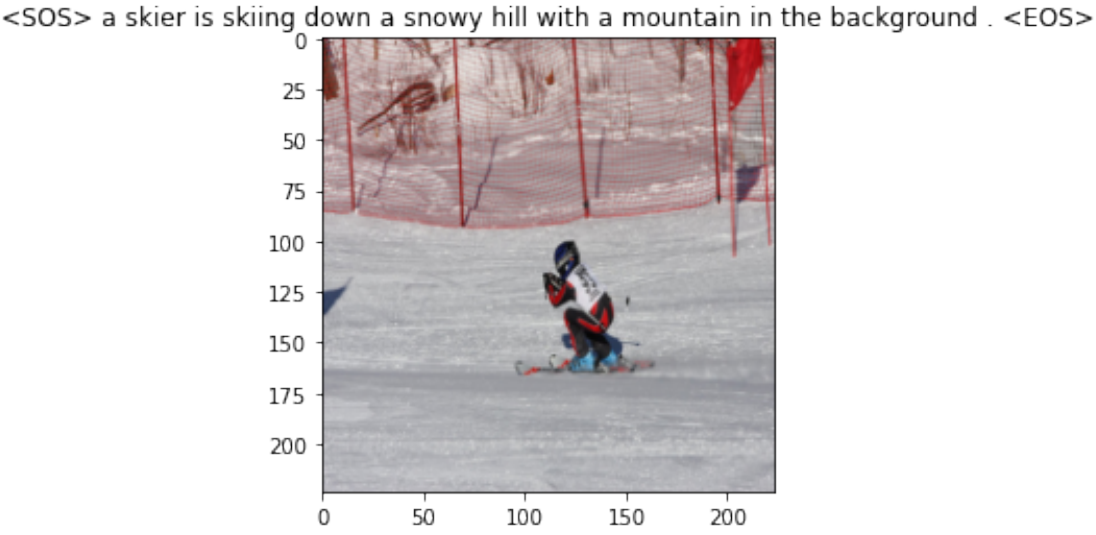



```
Epoch: 10 loss: 1.89351
features shape - torch.Size([1, 400])
<SOS> a man in a red jacket is standing on a mountain . <EOS>
```

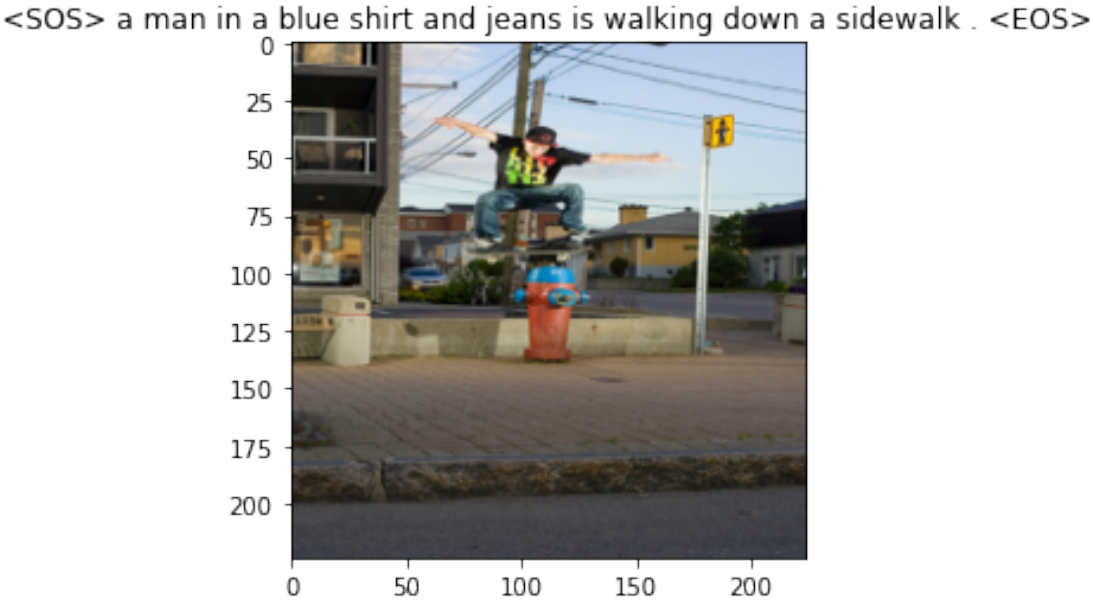
```
Epoch: 10 loss: 1.78173
features shape - torch.Size([1, 400])
<SOS> a person in a red jacket is climbing up a snowy mountain . <EOS>
```




```
Epoch: 10 loss: 1.62482
features shape - torch.Size([1, 400])
<SOS> a skier is skiing down a snowy hill with a mountain in the background .
<EOS>
```

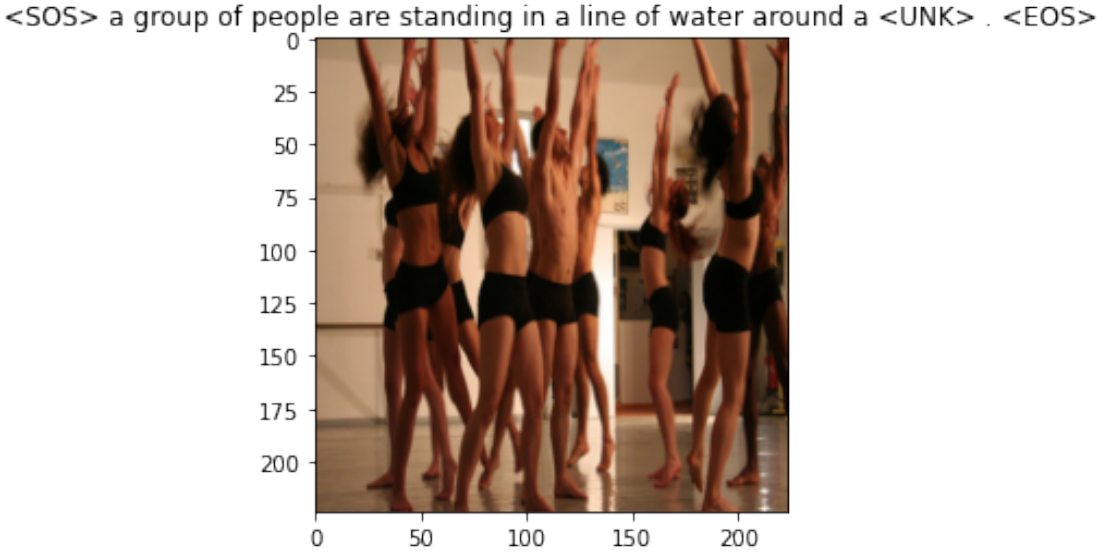


```
Epoch: 10 loss: 1.49703
features shape - torch.Size([1, 400])
<SOS> a man in a blue shirt and jeans is walking down a sidewalk . <EOS>
```

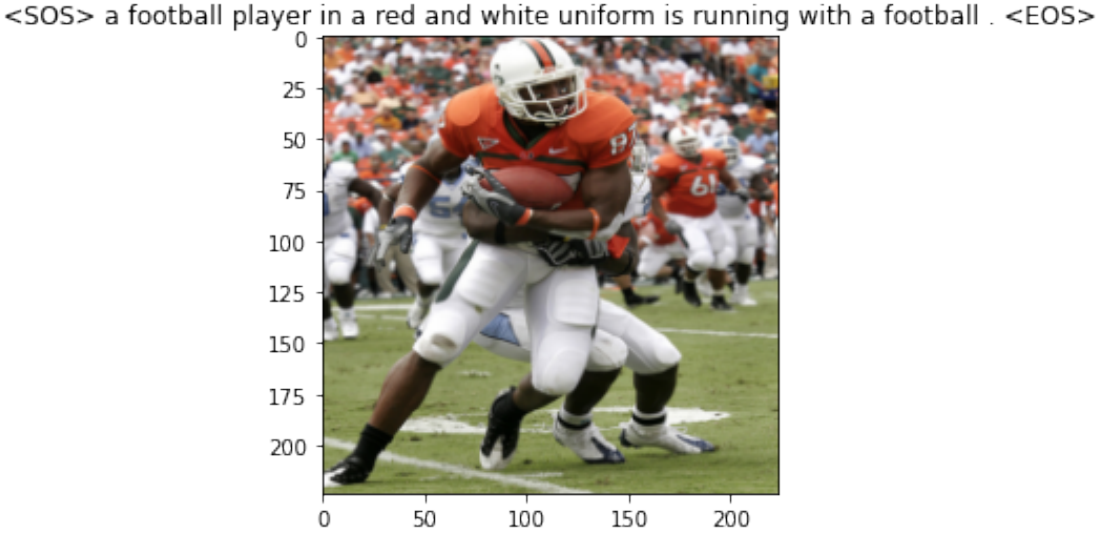



```
Epoch: 11 loss: 1.99301
features shape - torch.Size([1, 400])
<SOS> a group of people are standing together in front of a building . <EOS>
```

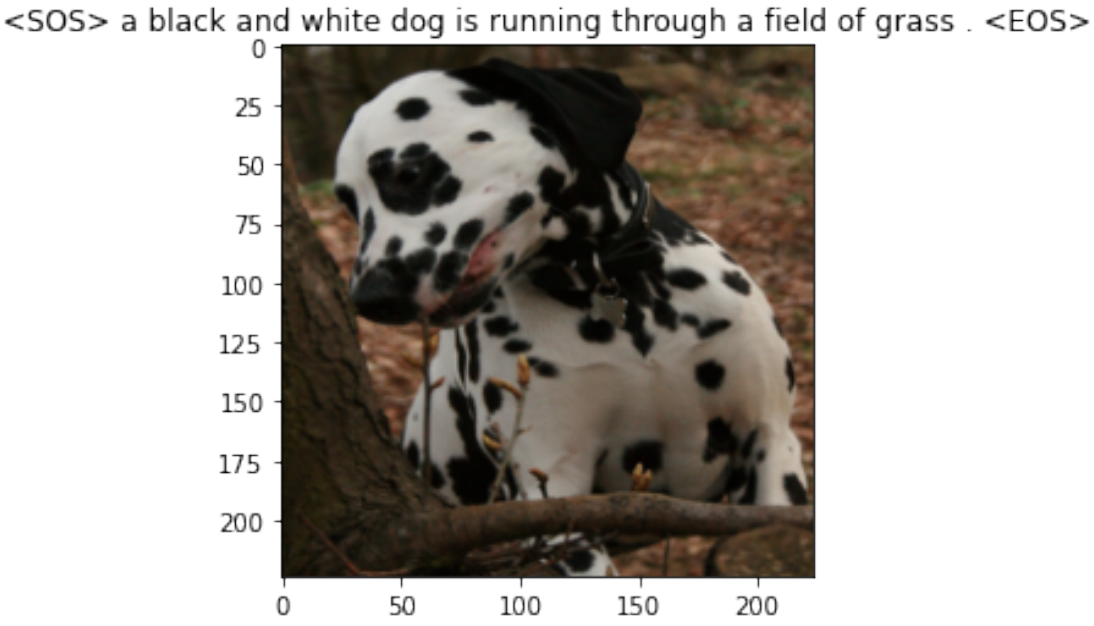
```
Epoch: 11 loss: 2.22866
features shape - torch.Size([1, 400])
<SOS> a group of people are standing in a line of water around a <UNK> . <EOS>
```



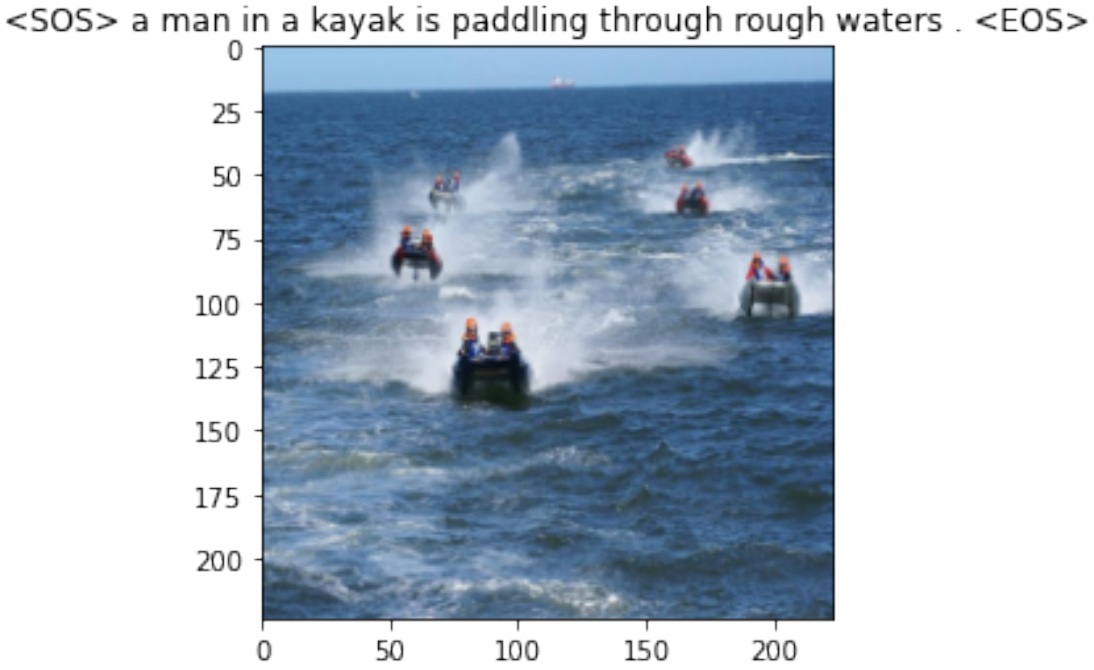

```
Epoch: 11 loss: 1.52018
features shape - torch.Size([1, 400])
<SOS> a football player in a red and white uniform is running with a football .
<EOS>
```



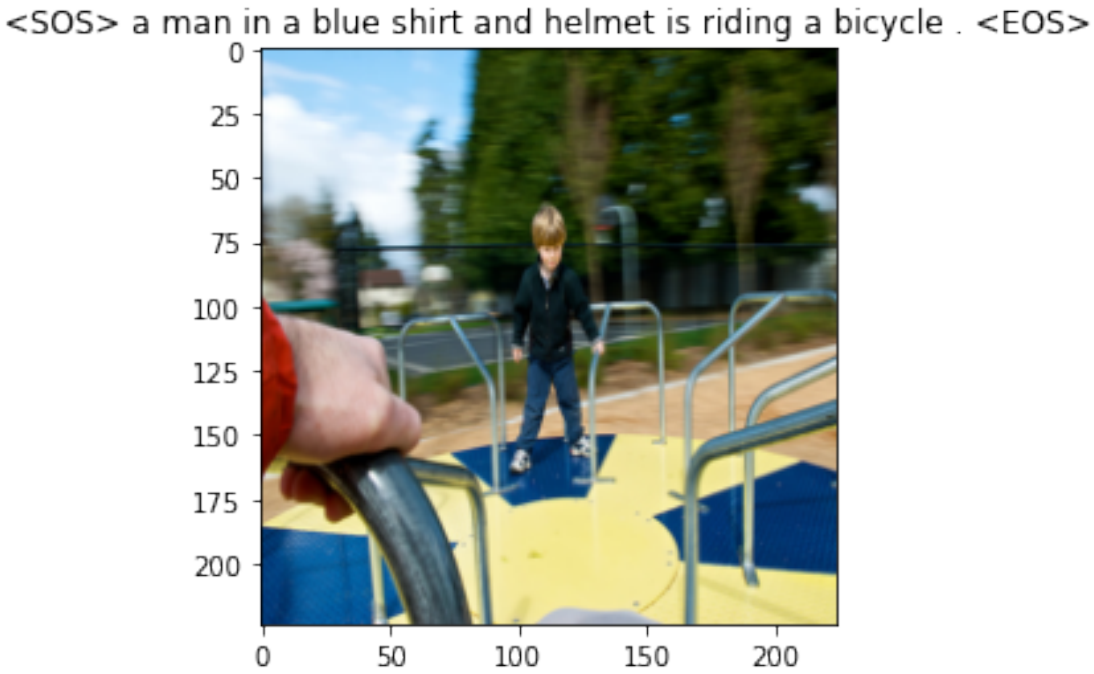
```
Epoch: 11 loss: 1.64358
features shape - torch.Size([1, 400])
<SOS> a black and white dog is running through a field of grass . <EOS>
```



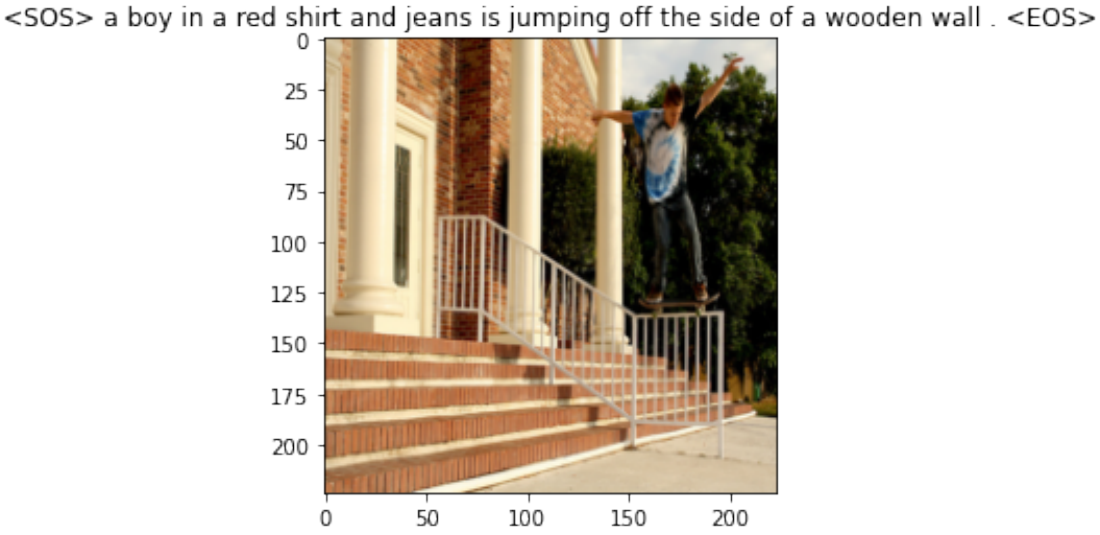

```
Epoch: 11 loss: 1.81231
features shape - torch.Size([1, 400])
<SOS> a man in a kayak is paddling through rough waters . <EOS>
```



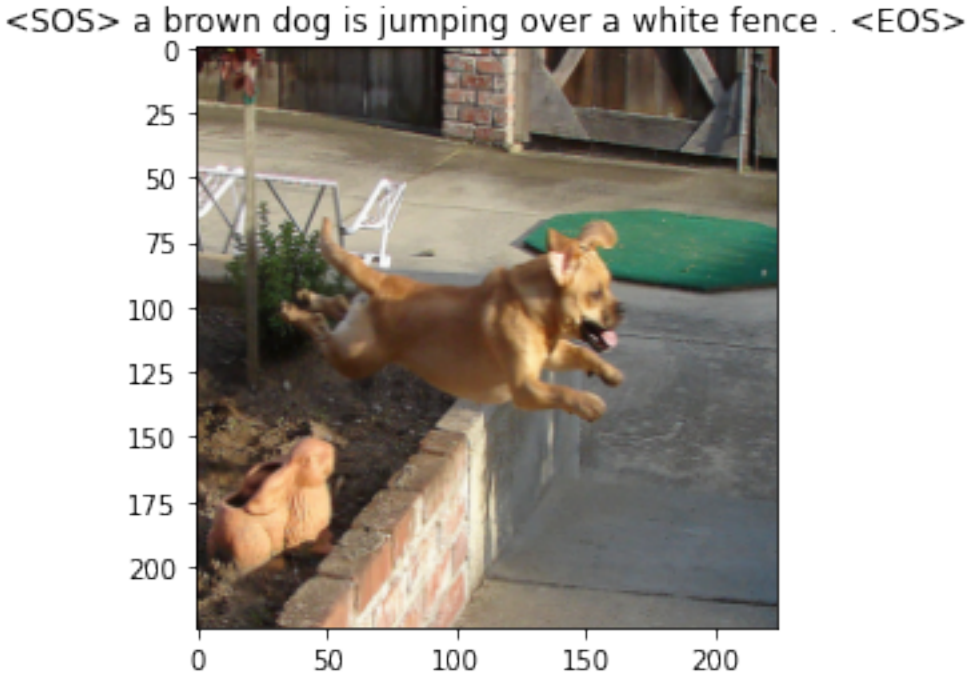
```
Epoch: 12 loss: 1.43334
features shape - torch.Size([1, 400])
<SOS> a man in a blue shirt and helmet is riding a bicycle . <EOS>
```



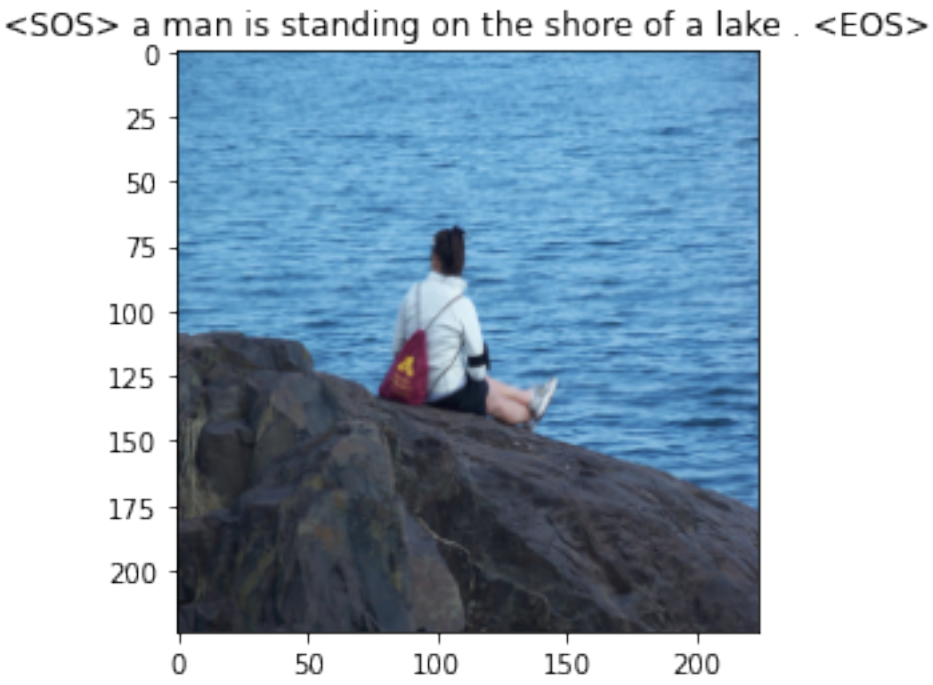

```
Epoch: 12 loss: 1.62114
features shape - torch.Size([1, 400])
<SOS> a boy in a red shirt and jeans is jumping off the side of a wooden wall .
<EOS>
```



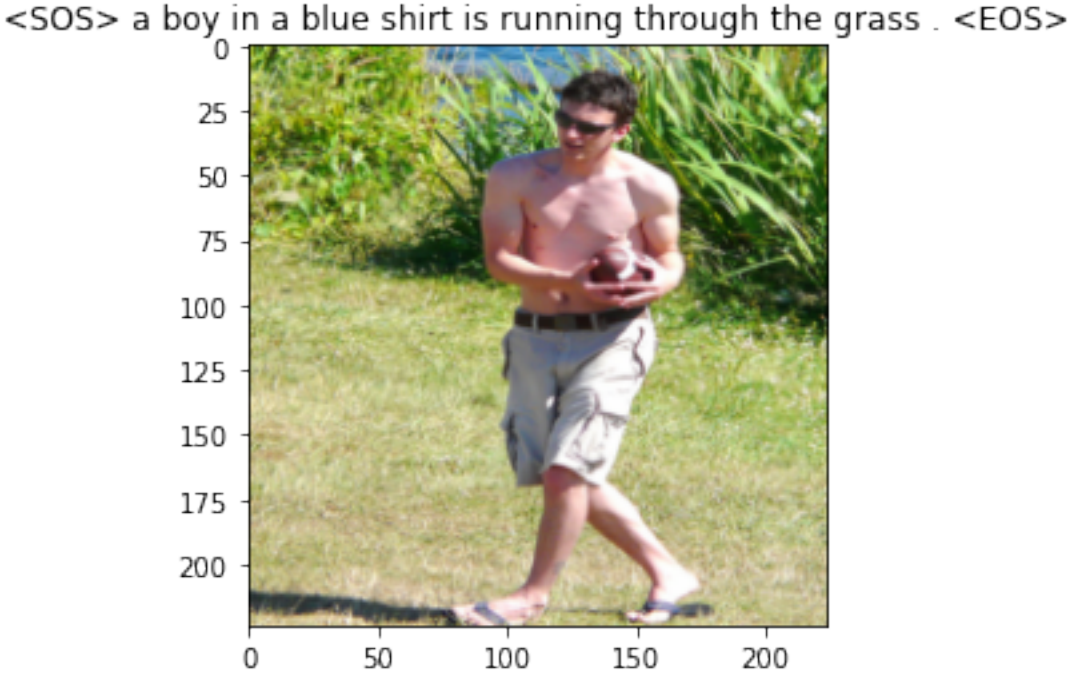

```
Epoch: 12 loss: 1.45096
features shape - torch.Size([1, 400])
<SOS> a brown dog is jumping over a white fence . <EOS>
```



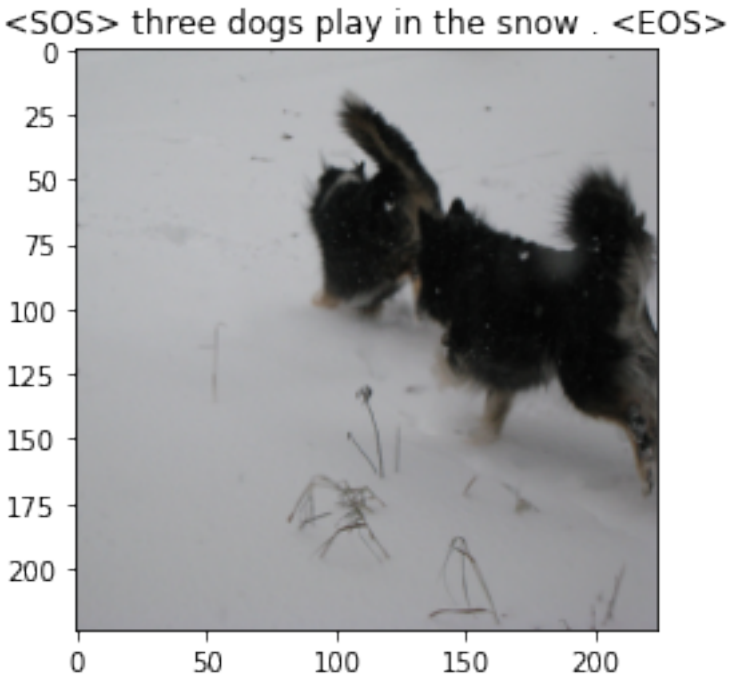

```
Epoch: 12 loss: 1.56160
features shape - torch.Size([1, 400])
<SOS> a man is standing on the shore of a lake . <EOS>
```



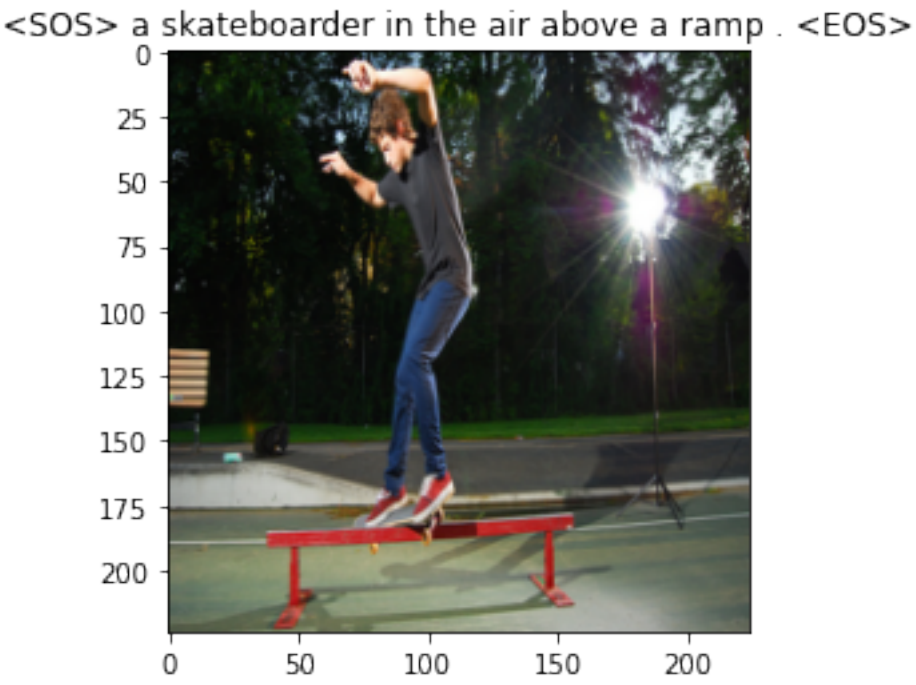

```
Epoch: 12 loss: 2.23930
features shape - torch.Size([1, 400])
<SOS> a boy in a blue shirt is running through the grass . <EOS>
```



```
Epoch: 13 loss: 1.69041
features shape - torch.Size([1, 400])
<SOS> three dogs play in the snow . <EOS>
```

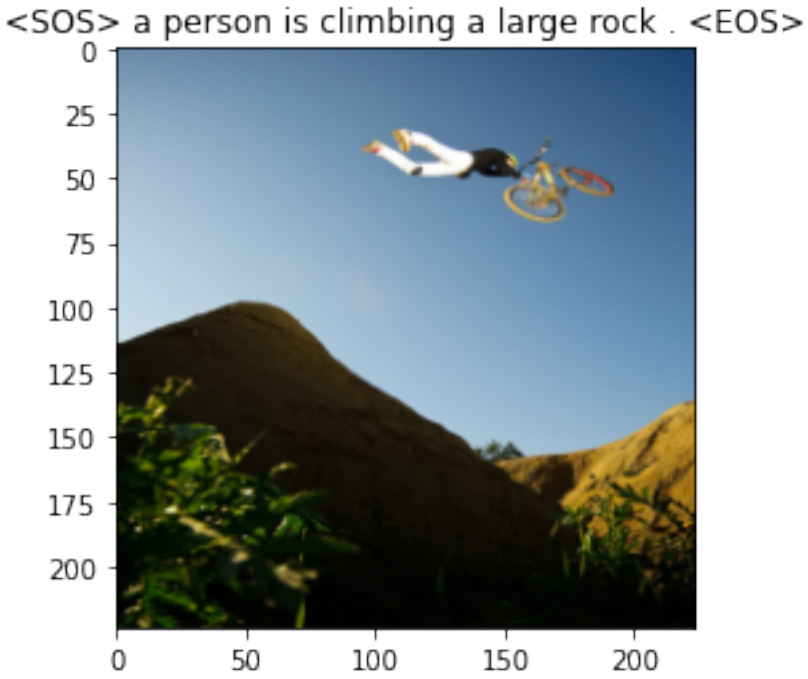



```
Epoch: 13 loss: 1.46840
features shape - torch.Size([1, 400])
<SOS> a skateboarder in the air above a ramp . <EOS>
```

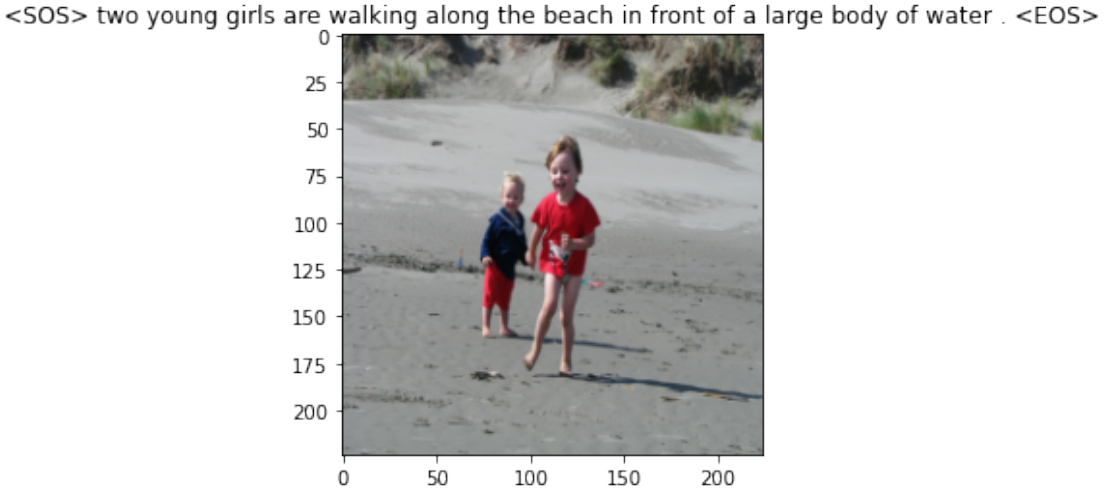



```
Epoch: 13 loss: 1.48965
features shape - torch.Size([1, 400])
<SOS> a boy wearing a blue shirt is running through the grass . <EOS>
```

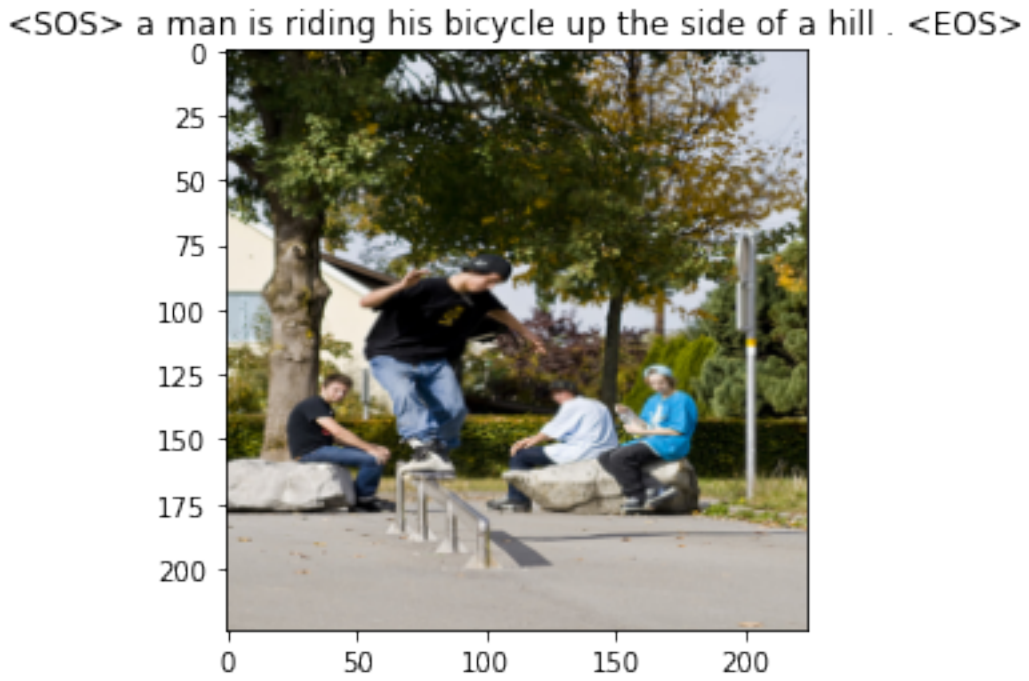
```
Epoch: 13 loss: 1.78002
features shape - torch.Size([1, 400])
<SOS> a person is climbing a large rock . <EOS>
```



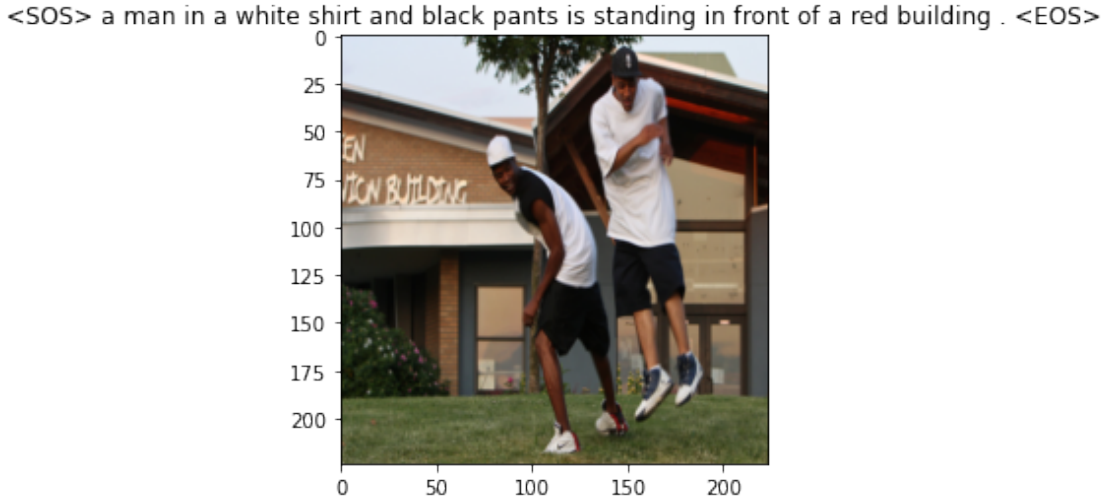

```
Epoch: 13 loss: 2.03370
features shape - torch.Size([1, 400])
<SOS> two young girls are walking along the beach in front of a large body of
water . <EOS>
```



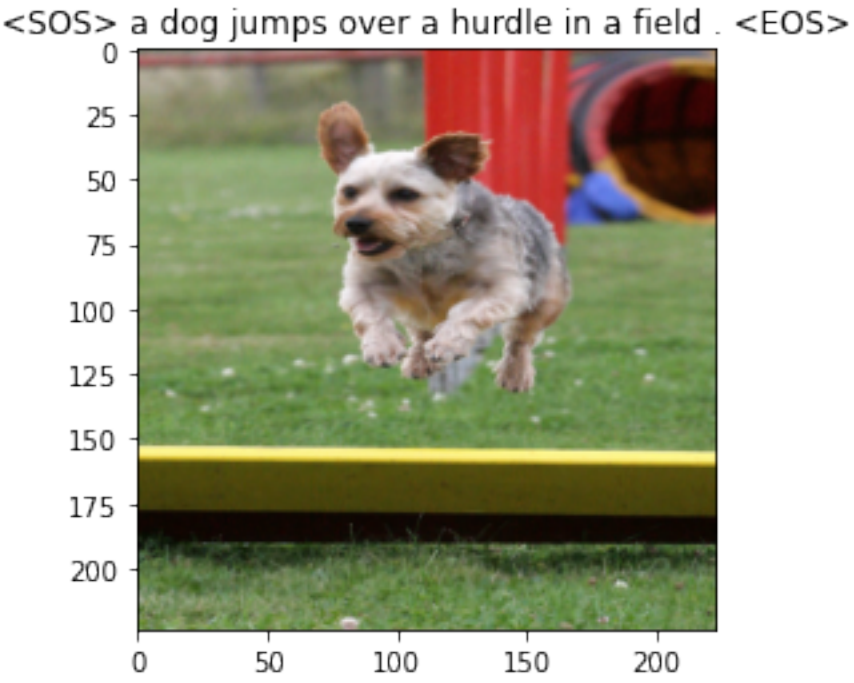
```
Epoch: 14 loss: 1.49844
features shape - torch.Size([1, 400])
<SOS> a man is riding his bicycle up the side of a hill . <EOS>
```



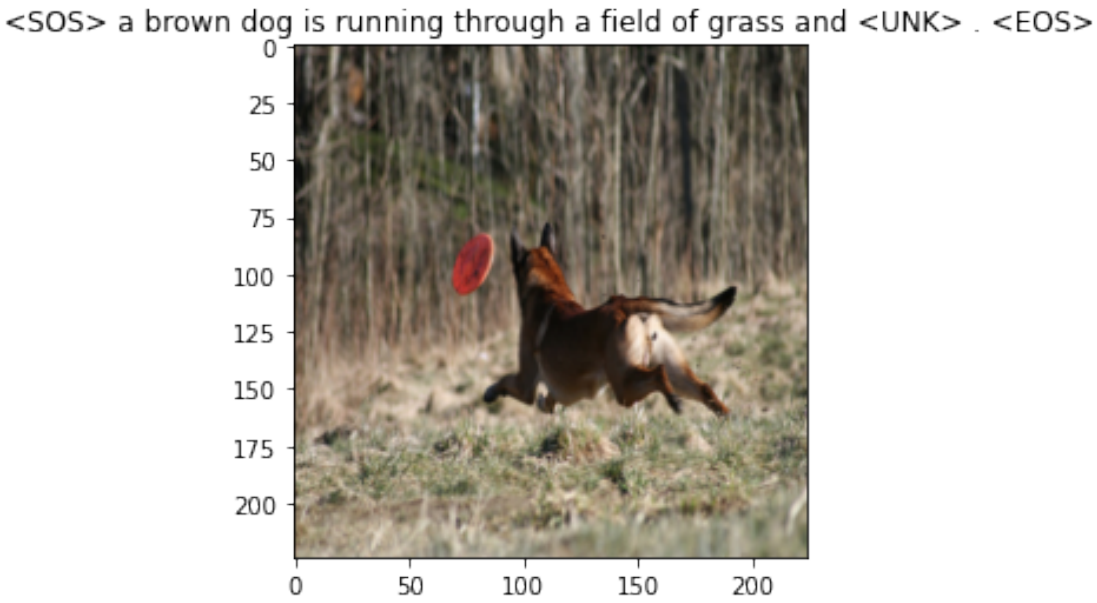

```
Epoch: 14 loss: 1.26312
features shape - torch.Size([1, 400])
<SOS> a man in a white shirt and black pants is standing in front of a red
building . <EOS>
```



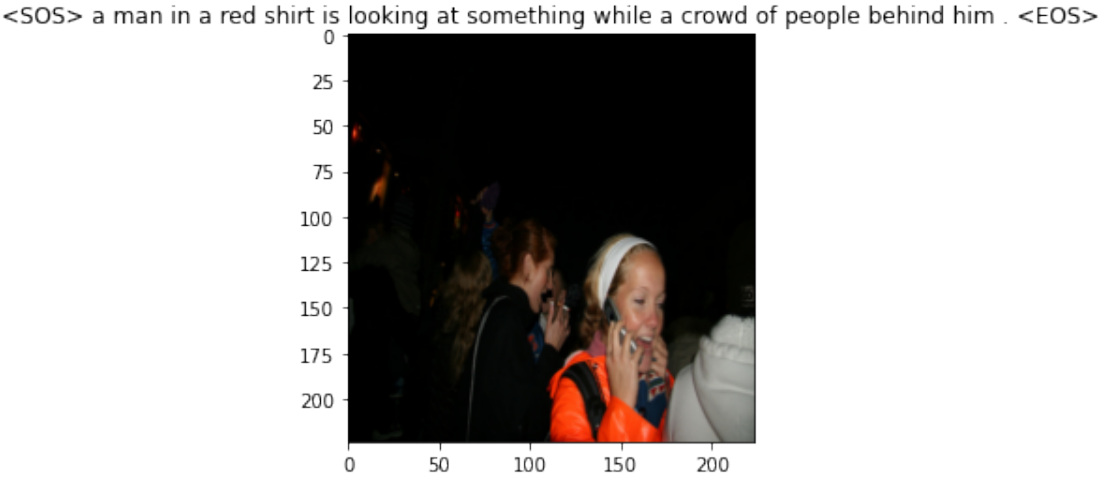
```
Epoch: 14 loss: 1.31252
features shape - torch.Size([1, 400])
<SOS> a dog jumps over a hurdle in a field . <EOS>
```



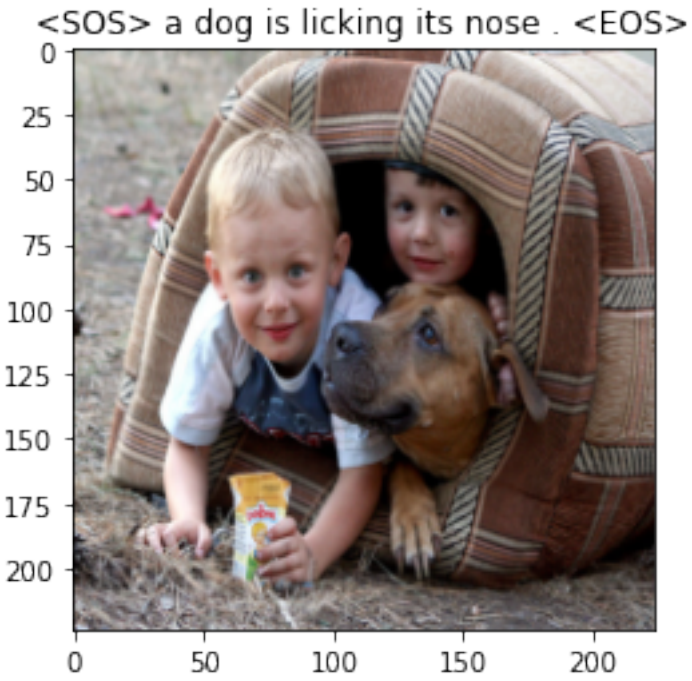

```
Epoch: 14 loss: 1.91136
features shape - torch.Size([1, 400])
<SOS> a brown dog is running through a field of grass and <UNK> . <EOS>
```



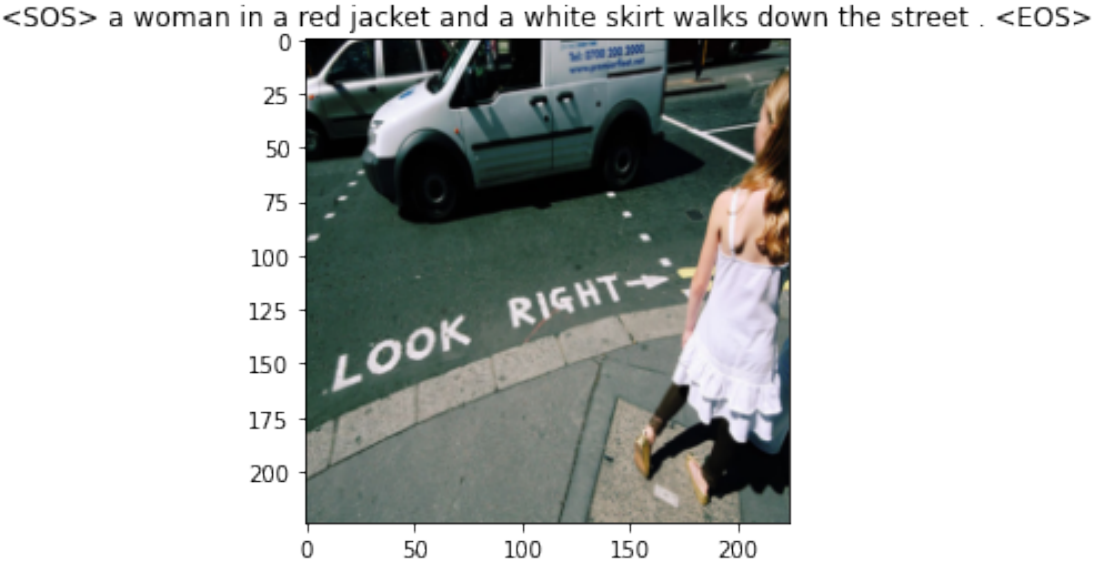

```
Epoch: 14 loss: 1.48404
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt is looking at something while a crowd of people
behind him . <EOS>
```




```
Epoch: 15 loss: 1.51153
features shape - torch.Size([1, 400])
<SOS> a dog is licking its nose . <EOS>
```

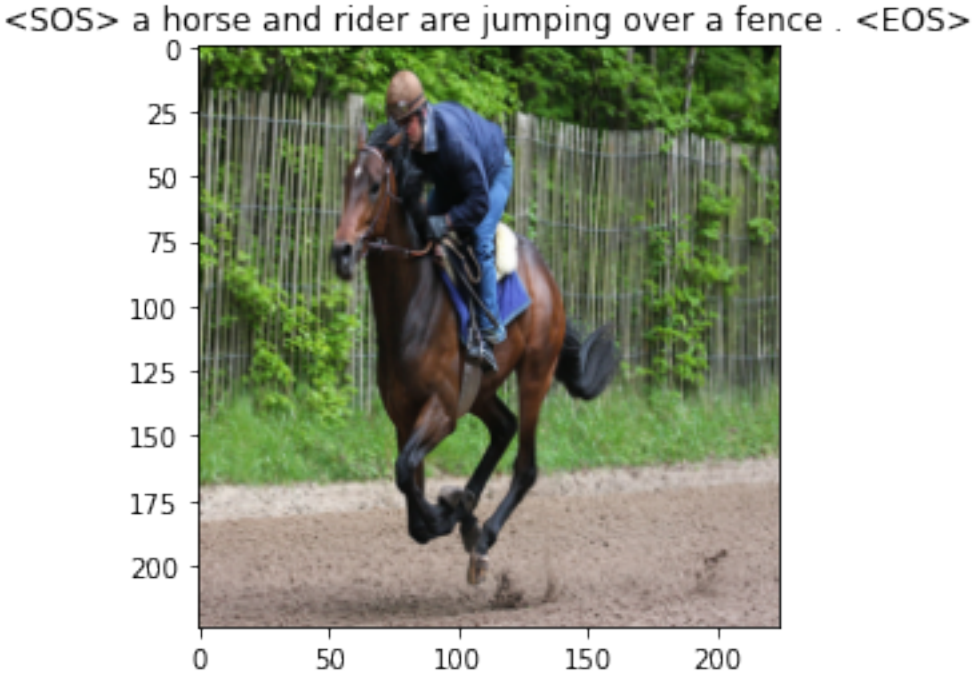



```
Epoch: 15 loss: 1.33652
features shape - torch.Size([1, 400])
<SOS> a woman in a red jacket and a white skirt walks down the street . <EOS>
```

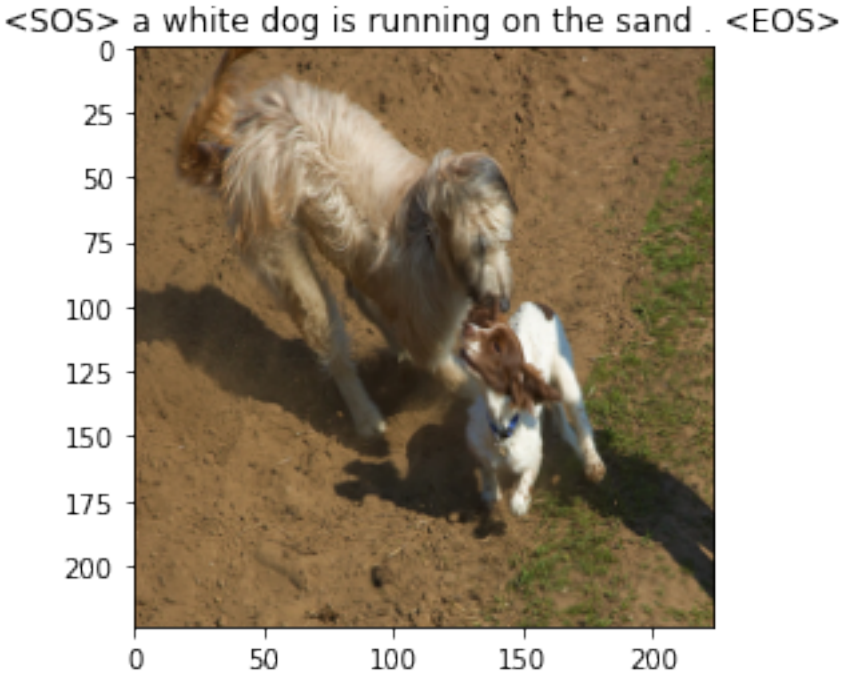



```
Epoch: 15 loss: 1.49241
features shape - torch.Size([1, 400])
```

<SOS> a horse and rider are jumping over a fence . <EOS>

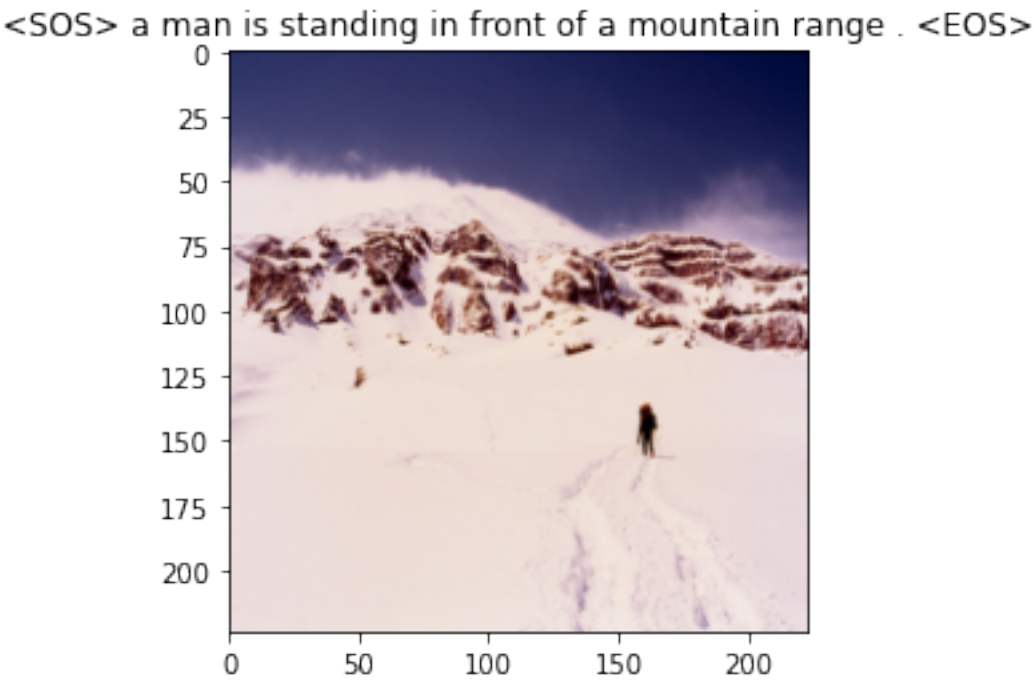


Epoch: 15 loss: 1.47096
features shape - torch.Size([1, 400])
<SOS> a white dog is running on the sand . <EOS>

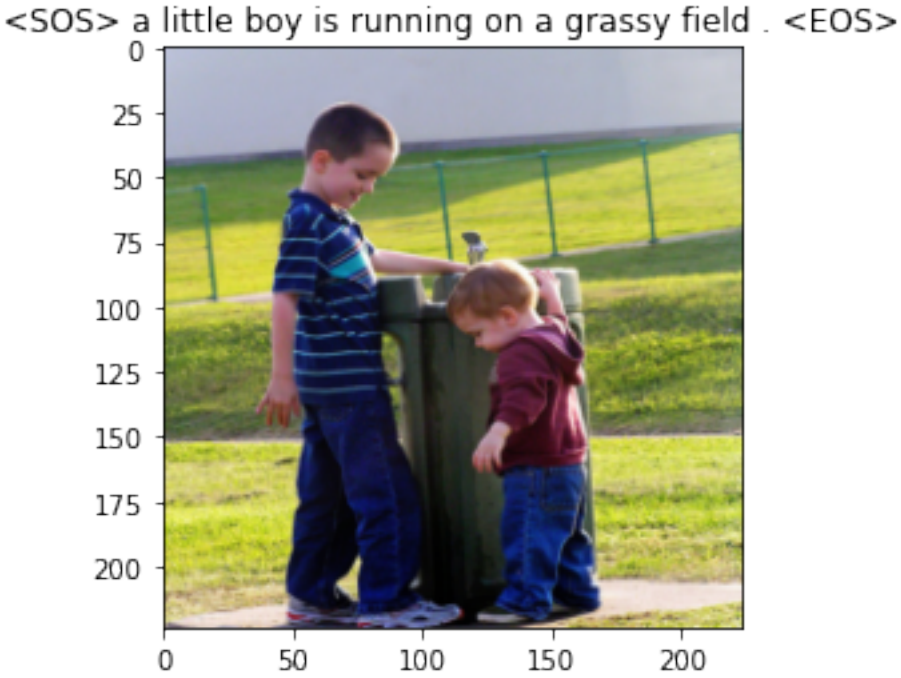


```
Epoch: 15 loss: 1.44012
features shape - torch.Size([1, 400])
<SOS> a boy in a blue shirt is running . <EOS>
```

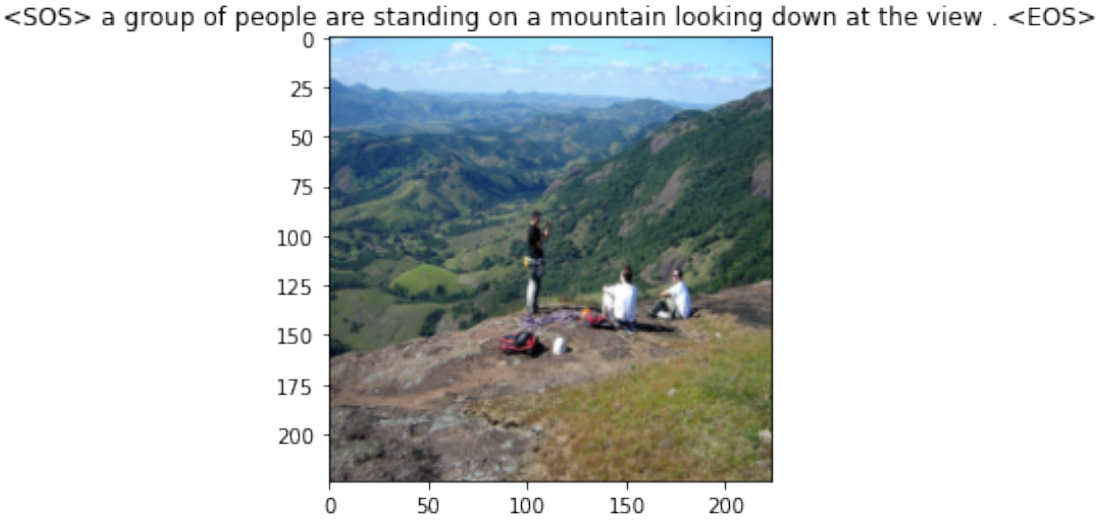

```
Epoch: 16 loss: 1.40911
features shape - torch.Size([1, 400])
<SOS> a man is standing in front of a mountain range . <EOS>
```



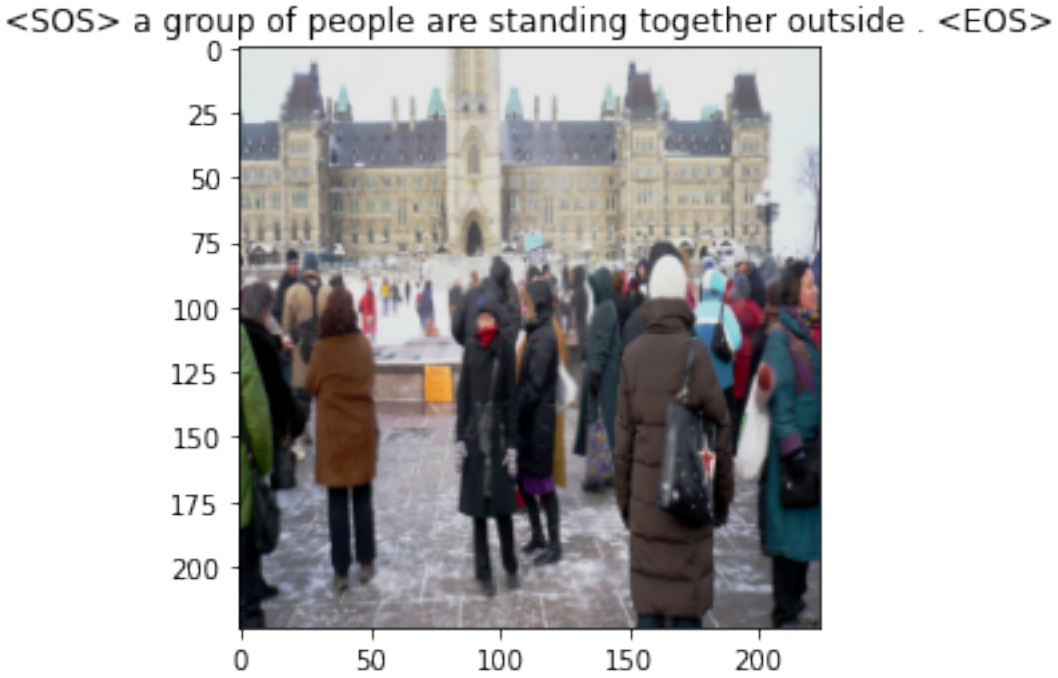

```
Epoch: 16 loss: 1.46050
features shape - torch.Size([1, 400])
<SOS> a little boy is running on a grassy field . <EOS>
```



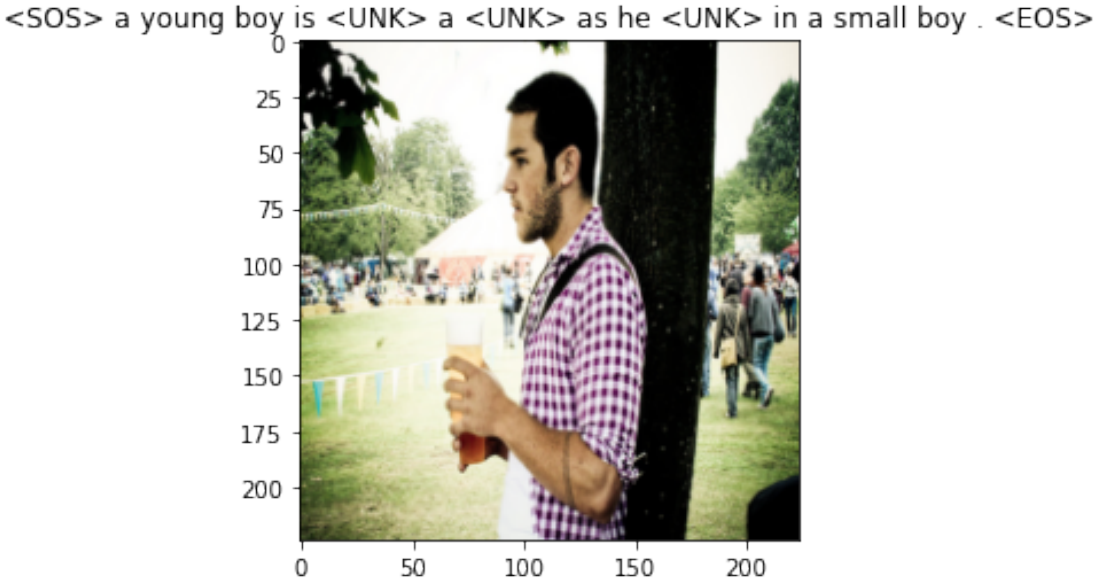
```
Epoch: 16 loss: 1.37911
features shape - torch.Size([1, 400])
<SOS> a group of people are standing on a mountain looking down at the view .
<EOS>
```



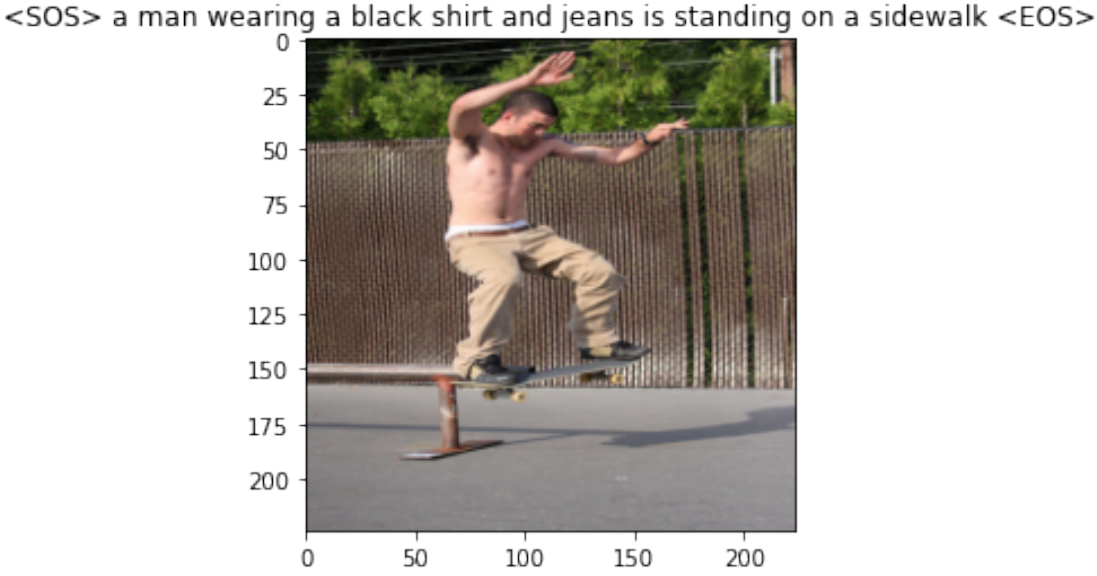

```
Epoch: 16 loss: 0.88052
features shape - torch.Size([1, 400])
<SOS> a group of people are standing together outside . <EOS>
```



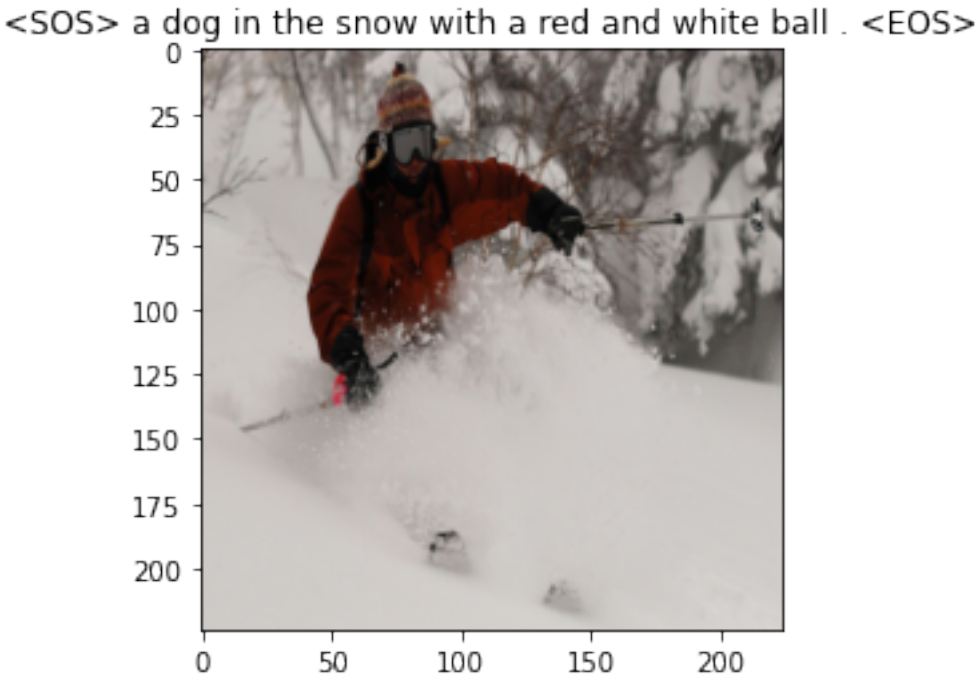
```
Epoch: 16 loss: 1.58364
features shape - torch.Size([1, 400])
<SOS> a young boy is <UNK> a <UNK> as he <UNK> in a small boy . <EOS>
```



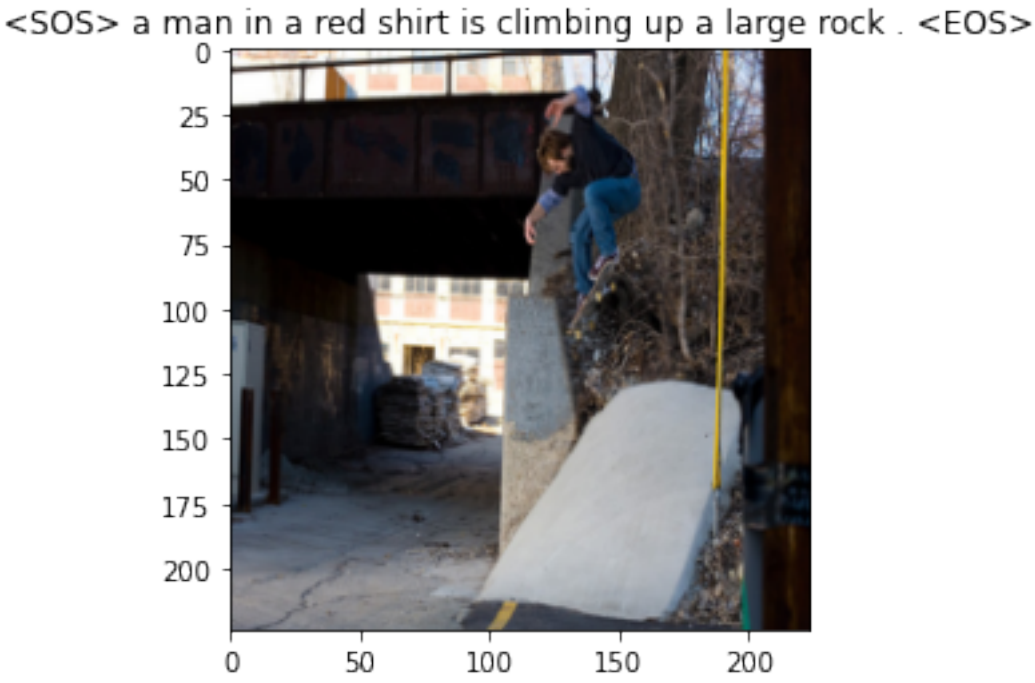

```
Epoch: 17 loss: 1.43897
features shape - torch.Size([1, 400])
<SOS> a man wearing a black shirt and jeans is standing on a sidewalk <EOS>
```



```
Epoch: 17 loss: 1.20997
features shape - torch.Size([1, 400])
<SOS> a dog in the snow with a red and white ball . <EOS>
```



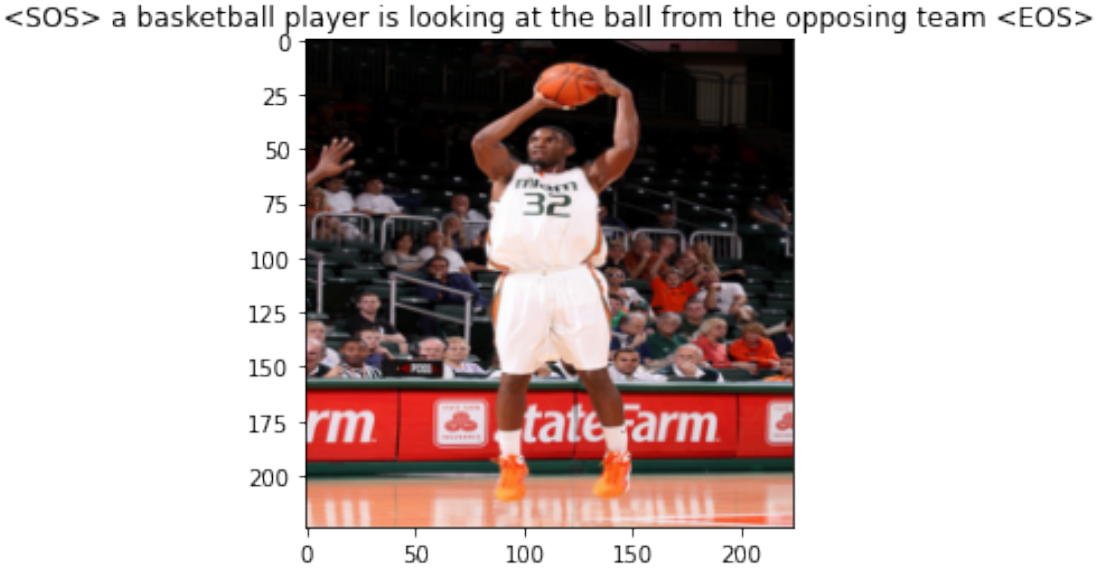

```
Epoch: 17 loss: 1.32309
features shape - torch.Size([1, 400])
<SOS> a man in a red shirt is climbing up a large rock . <EOS>
```



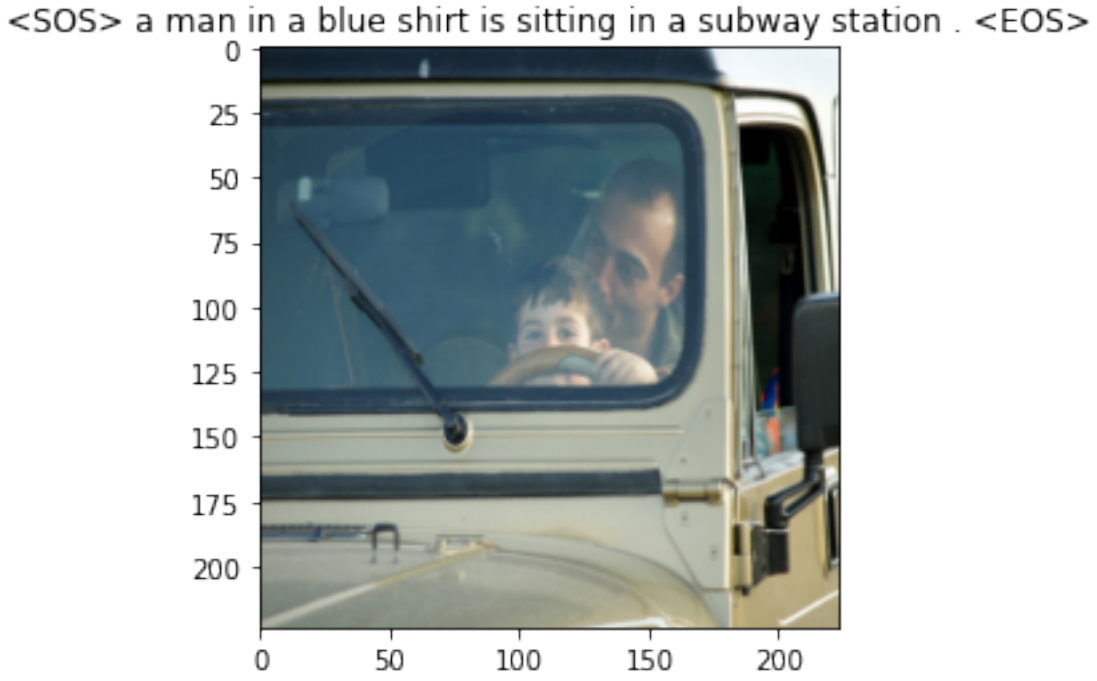

```
Epoch: 17 loss: 1.26028
features shape - torch.Size([1, 400])
<SOS> a large group of people are sitting and watching something . <EOS>
```


```
Epoch: 17 loss: 1.02822
```

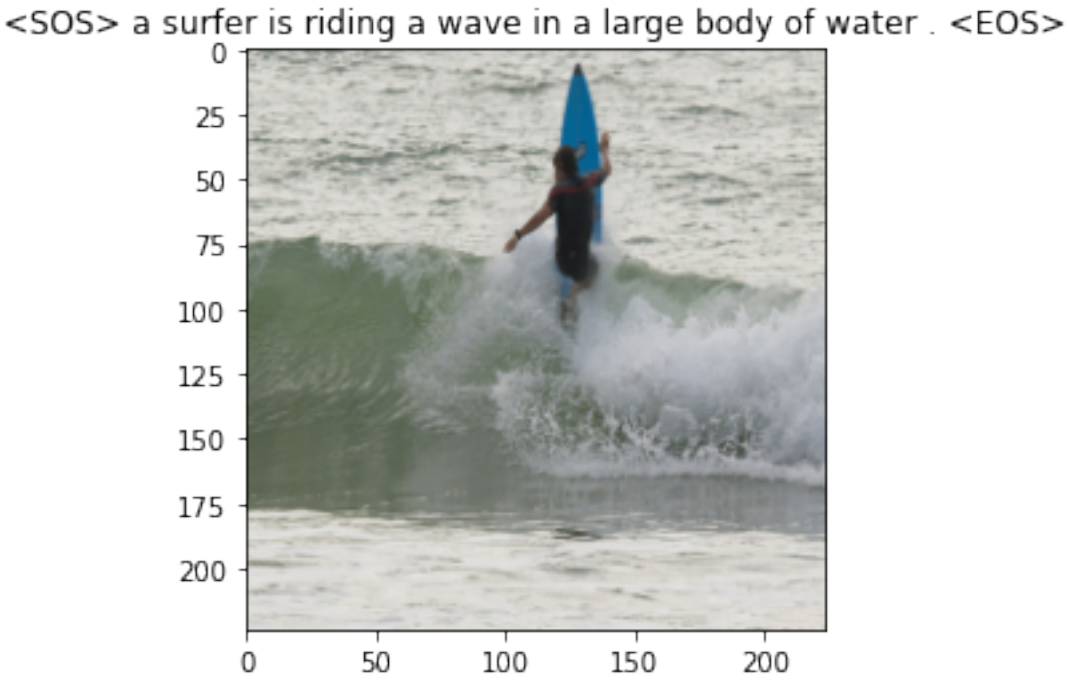
```
features shape - torch.Size([1, 400])
<SOS> a basketball player is looking at the ball from the opposing team <EOS>
```




```
Epoch: 18 loss: 1.18566
features shape - torch.Size([1, 400])
<SOS> a man in a blue shirt is sitting in a subway station . <EOS>
```

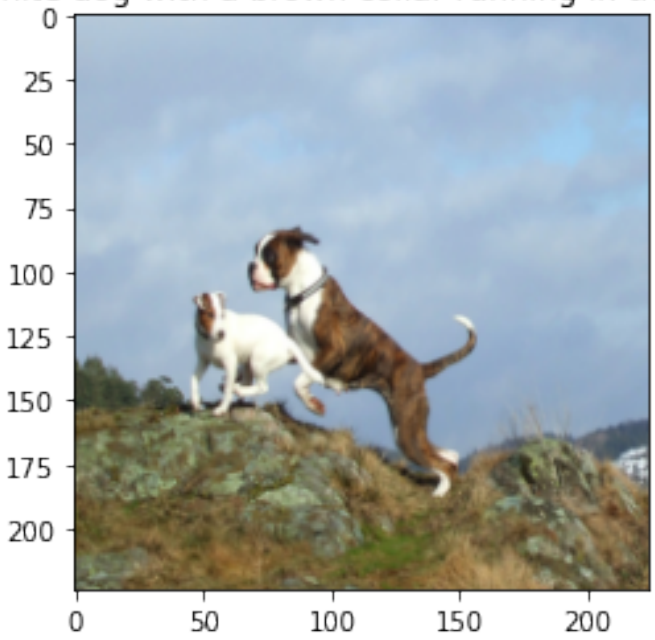


```
Epoch: 18 loss: 0.99385
features shape - torch.Size([1, 400])
<SOS> a surfer is riding a wave in a large body of water . <EOS>
```



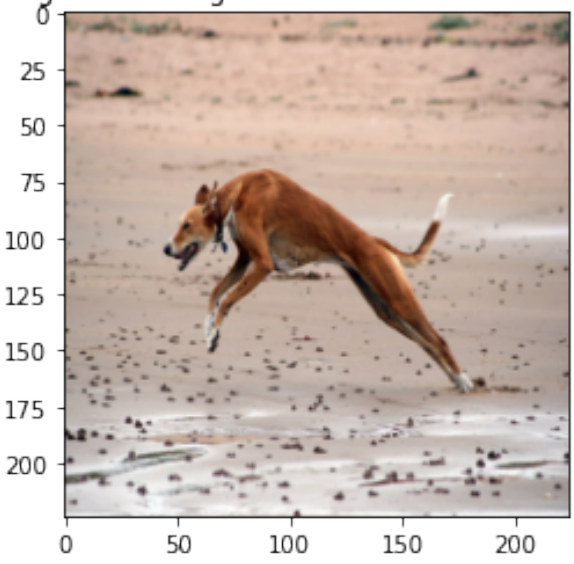

```
Epoch: 18 loss: 1.60899
features shape - torch.Size([1, 400])
<SOS> a white dog with a brown collar running in the grass <EOS>
```

<SOS> a white dog with a brown collar running in the grass <EOS>



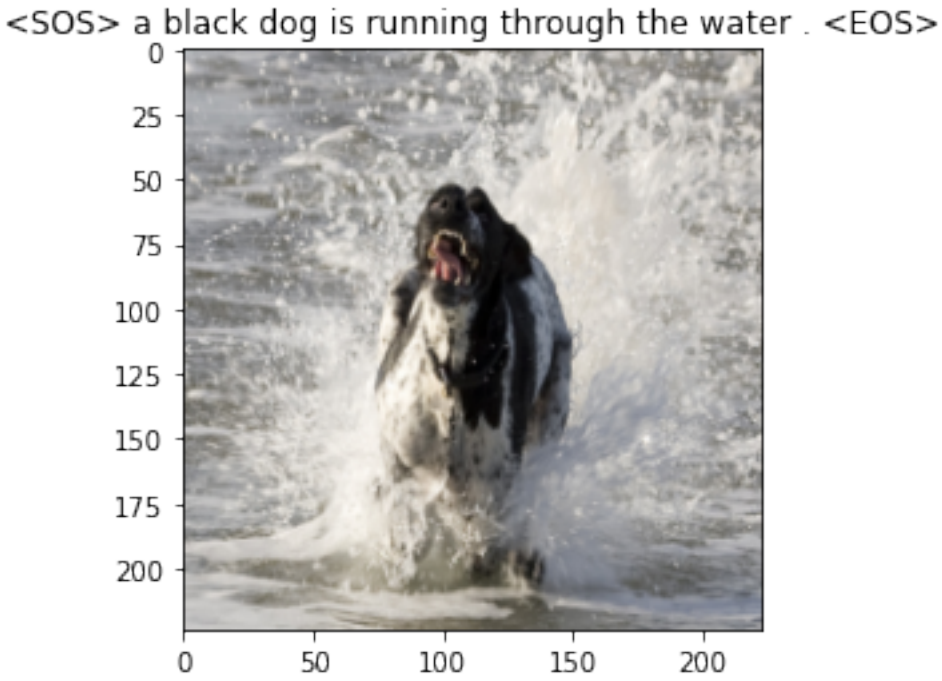

```
Epoch: 18 loss: 1.67471
features shape - torch.Size([1, 400])
<SOS> a brown dog runs through the water with a stick in its mouth . <EOS>
```

<SOS> a brown dog runs through the water with a stick in its mouth . <EOS>

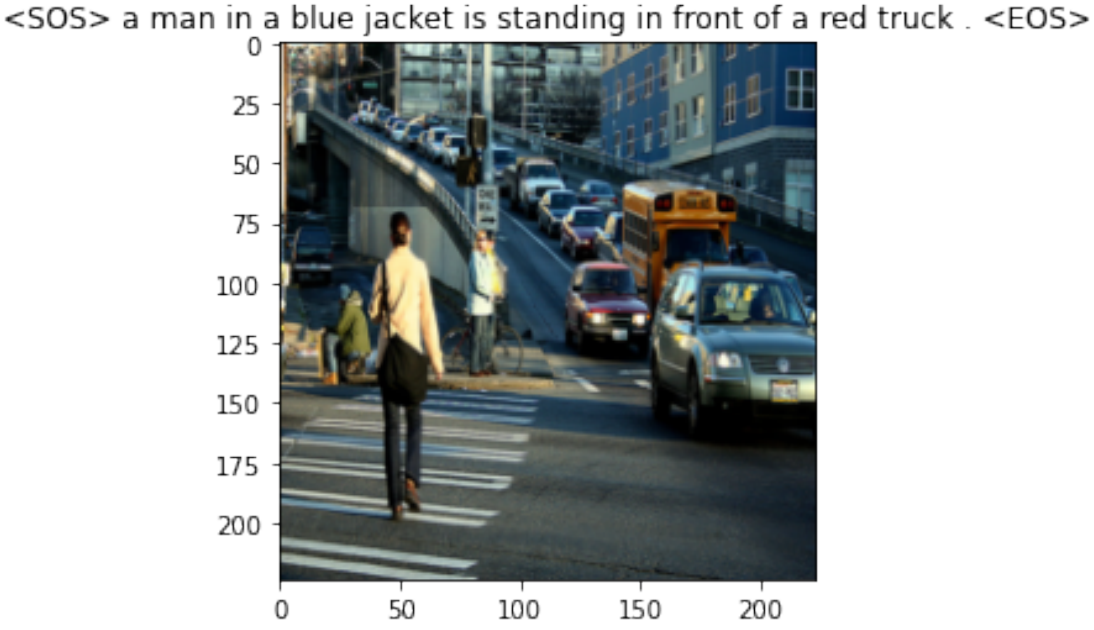



```
Epoch: 18 loss: 1.23940
```

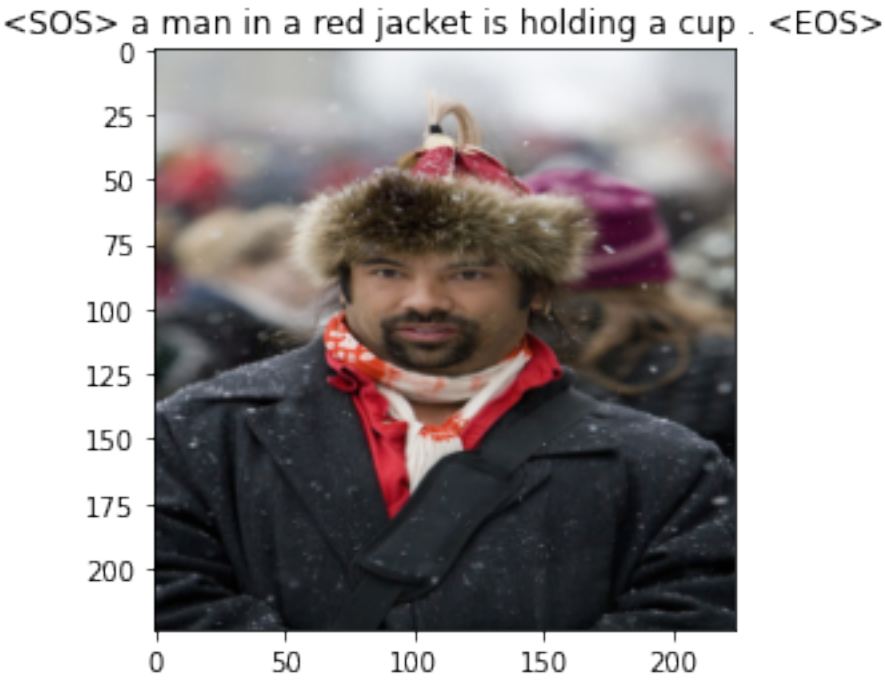
```
features shape - torch.Size([1, 400])
<SOS> a black dog is running through the water . <EOS>
```




```
Epoch: 19 loss: 1.16485
features shape - torch.Size([1, 400])
<SOS> a man in a blue jacket is standing in front of a red truck . <EOS>
```

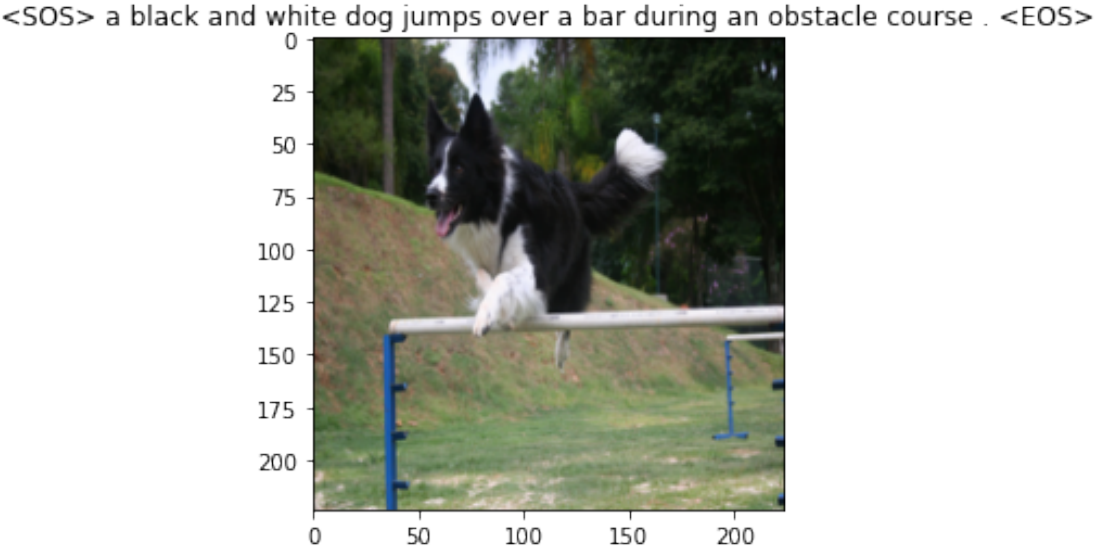


```
Epoch: 19 loss: 1.22149
features shape - torch.Size([1, 400])
<SOS> a man in a red jacket is holding a cup . <EOS>
```



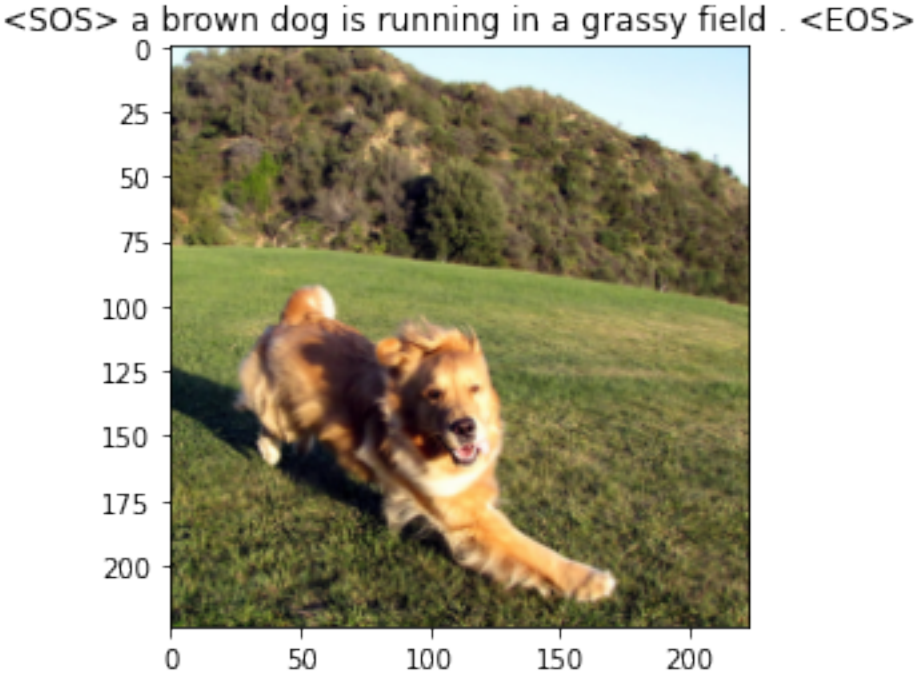

```
Epoch: 19 loss: 1.32016
features shape - torch.Size([1, 400])
<SOS> a little boy is playing in a blow up toy . <EOS>
```


```
Epoch: 19 loss: 1.61783
features shape - torch.Size([1, 400])
<SOS> a black and white dog jumps over a bar during an obstacle course . <EOS>
```

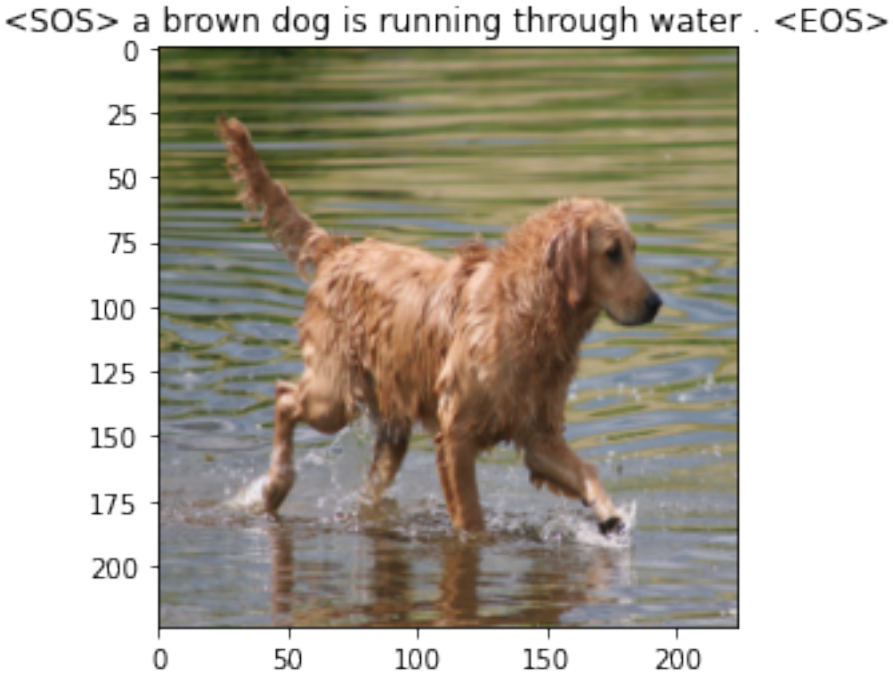



```
Epoch: 19 loss: 1.24004
features shape - torch.Size([1, 400])
```

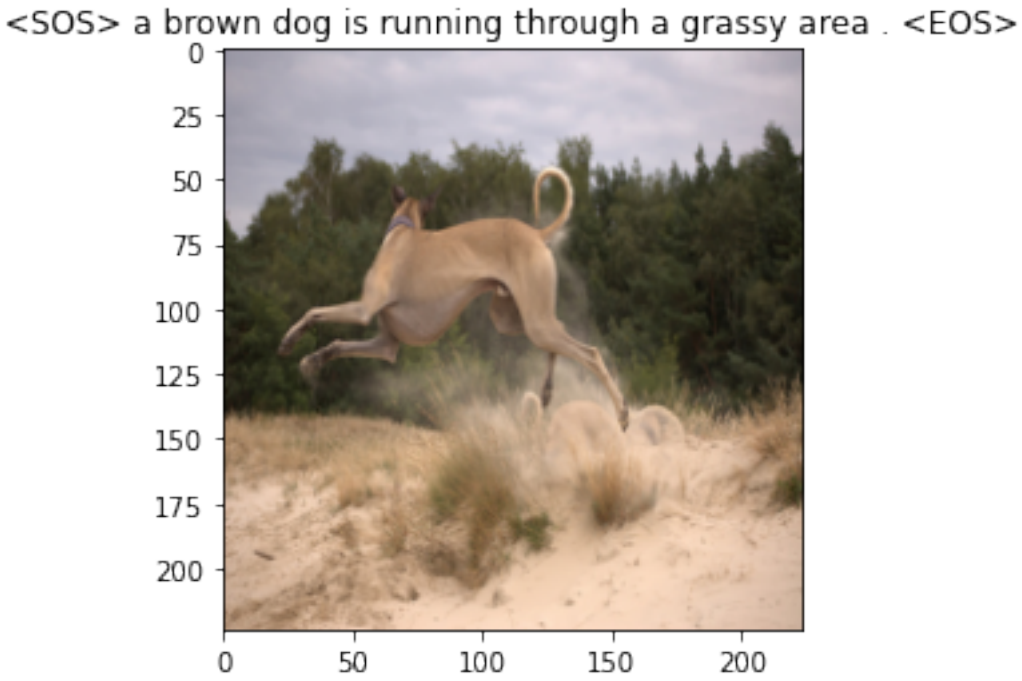
```
<SOS> a brown dog is running in a grassy field . <EOS>
```



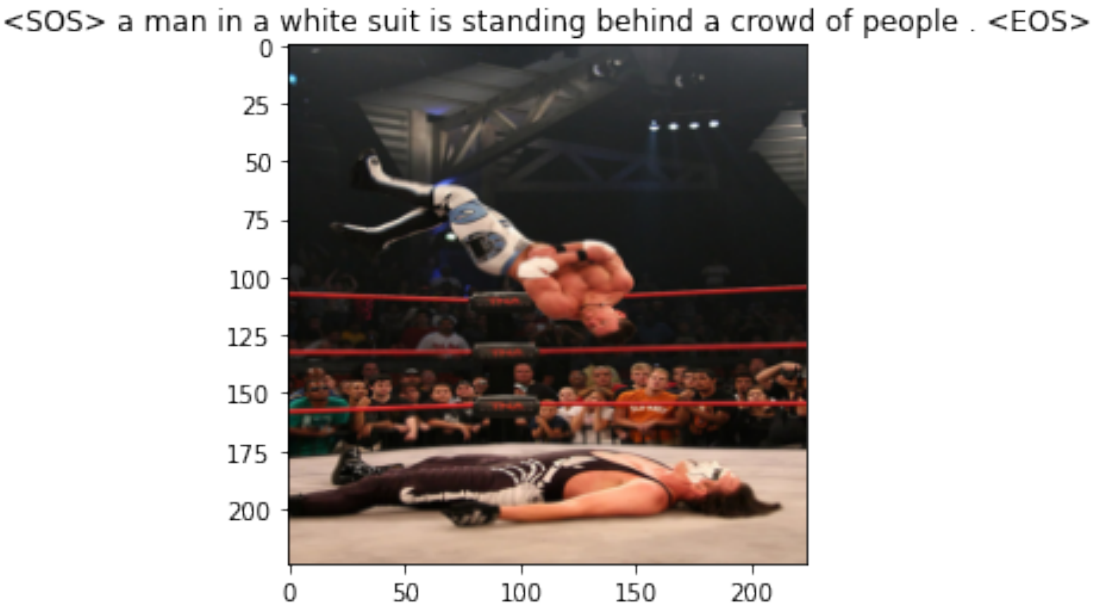

```
Epoch: 20 loss: 0.99562
features shape - torch.Size([1, 400])
<SOS> a brown dog is running through water . <EOS>
```



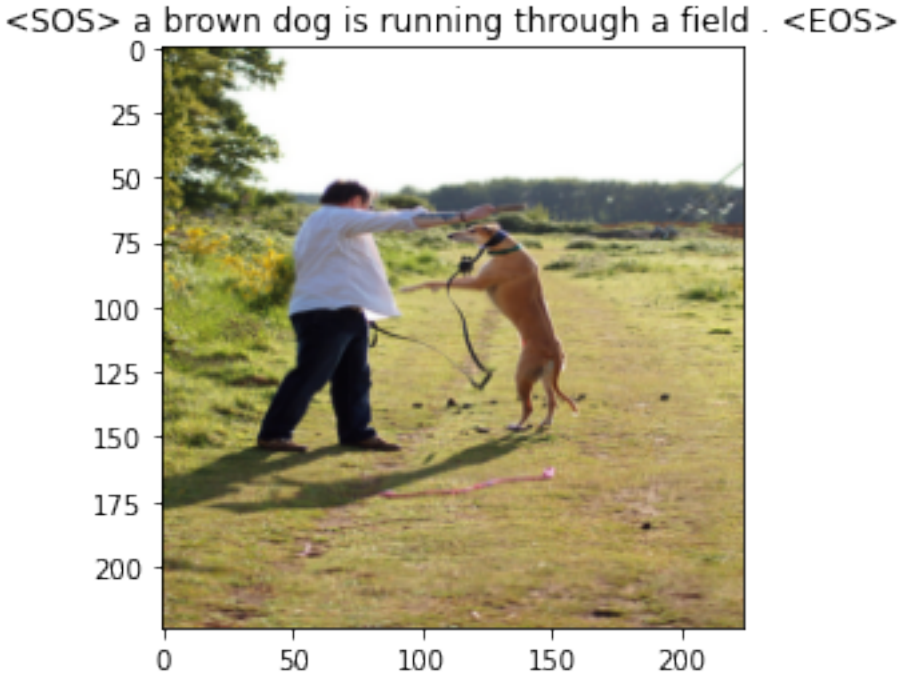
```
Epoch: 20 loss: 1.48864
features shape - torch.Size([1, 400])
<SOS> a brown dog is running through a grassy area . <EOS>
```




```
Epoch: 20 loss: 1.43418
features shape - torch.Size([1, 400])
<SOS> a man in a white suit is standing behind a crowd of people . <EOS>
```




```
Epoch: 20 loss: 1.21785
features shape - torch.Size([1, 400])
<SOS> a brown dog is running through a field . <EOS>
```




```
Epoch: 20 loss: 1.25592
```

```
features shape - torch.Size([1, 400])
<SOS> a man in a red jacket is standing in front of a colorful truck . <EOS>
```