UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI MATEMATICA E INFORMATICA

ARTIFICIAL INTELLIGENCE AND COMPUTER SCIENCE

DEEP LEARNING

Image Captioning Generator Model

Submitted to:

Prof. Gianluigi Greco

Dott. Carlo Adornetto

Submitted by:

Jana Mamaree
Mat. 241157

Sami Sulaiman
Mat. 251560

ACADEMIC YEAR 2024/2025

1. Introduction

Image captioning is a task in computer vision that combines the ability of machines to understand images with the ability to generate natural language descriptions. By automating the process of generating text for images, image captioning has a wide range of applications, including helping visually impaired individuals understand their environment, providing context to social media images, and improving search engines by associating relevant captions with images.

The primary objective of this project is to build an image captioning model that can generate captions for images using a hybrid approach that combines Convolutional Neural Networks (CNNs) for feature extraction and Long Short-Term Memory (LSTM) networks for sequence generation. CNN extracts feature from images, while the LSTM generates captions based on those features. The VGG16 architecture was chosen for feature extraction due to its robust performance in image classification tasks. The captions are generated by feeding the image features into an LSTM model trained on a dataset of image-caption pairs.

This project investigates the effectiveness of pre-trained VGG16 for image feature extraction and explores the use of LSTM for generating coherent and contextually appropriate captions. Evaluating the model's performance will be done using the BLEU score, a common metric for machine-generated text in the context of image captioning.

2. Task - Image Captioning

a. Dataset Description

The Flickr8k dataset is used for image captioning tasks. It consists of 8,000 images sourced from the Flickr photo-sharing platform, each paired with 5 unique captions describing the content of the image.

The dataset contains images from various categories, including animals, nature, people, and everyday activities. These images are diverse and contain complex scenes, making the task of generating accurate captions challenging.

Number of Images: 8,000 images with 5 captions each, totaling 40,000 unique captions.

Source: Flickr8k Dataset

The captions in the dataset were preprocessed by removing non-essential characters, converting all text to lowercase, and tokenizing the text into sequences of words. This

preprocessed text data was then used to train the LSTM model. The images were resized to a consistent size and passed through the VGG16 model to extract features.

b. Methods

i. Preprocessing

To effectively train the image captioning model, the following preprocessing steps were implemented:

- Loading and Resizing Images: The images from the Flickr8k dataset were loaded into memory and resized to 224x224 pixels. This size was chosen because it matches the input size required by the VGG16 model, which is widely used for feature extraction tasks. Resizing is essential to ensure that all images have consistent dimensions, allowing the neural network to process them efficiently.

- Normalization: To standardize the image inputs for the VGG16 model, the "preprocess_input" function from Keras was applied. This function normalizes the pixel values of each image by scaling them to the range expected by VGG16. Specifically, it adjusts pixel values by subtracting the mean RGB value and scales them accordingly. This normalization is crucial because the VGG16 model was pretrained on ImageNet, and such preprocessing ensures that the input data is compatible with the model's expectations.

- Caption Preprocessing: Each image in the Flickr8k dataset is paired with five captions, which were tokenized into words using a word tokenizer. The tokenizer was fitted on the captions and used to convert text into sequences of integer indices, representing the words in the vocabulary.
  A maximum sequence length was chosen is 34 words to standardize the length of the captions. Captions longer than this length were truncated, and shorter captions were padded with zeros to match the fixed length.

- Feature Extraction: The VGG16 model, pre-trained on the ImageNet dataset, was used to extract features from the images. The last fully connected layers of VGG16 were removed, and the output of the penultimate layer (a 4096-dimensional feature vector) was used as the feature representation of each image. This feature vector

encapsulates the important visual information that the LSTM will use to generate captions.

- Data Generator: A custom data generator was implemented to efficiently handle the data in batches during training. The generator yields pairs of image features and corresponding caption sequences, ensuring that the model is trained on varying inputs in each batch.

ii. Architecture and training

Model: "functional_2"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160

Figure 1 VGG16 architecture model

block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102,764,544
fc2 (Dense)	(None, 4096)	16,781,312

Total params: 134,260,544 (512.16 MB)

Trainable params: 134,260,544 (512.16 MB)

Non-trainable params: 0 (0.00 B)

Figure 2 VGG16 architecture model

- Feature Extraction (VGG16): The VGG16 network is a deep convolutional network known for its effectiveness in image recognition tasks. For this project, the VGG16 model was used as a fixed feature extractor by removing the fully connected layers and using the output from the last convolutional layer.
  The VGG16 model takes an image as input and outputs a 4096-dimensional feature vector, which is then passed to the LSTM network for caption generation.

- Sequence Generation (LSTM): The LSTM model is used to generate captions based on the image features. It consists of the following layers:
  1. Embedding Layer: This layer converts the integer-encoded words into dense vectors of fixed size, which are then fed into the LSTM layers.
  2. LSTM Layers: The LSTM processes the embedded word sequences and generates the next word in the caption. The LSTM uses the previously predicted words (or the <startseq> token for the first word) along with the image features to generate a sequence of words.
  3. Dense Layer: After the LSTM layers, the output is passed through a dense layer with a Softmax activation function to predict the next word in the sequence. The

dense layer produces a probability distribution over the vocabulary for each word in the caption.

- Training: The model was trained using categorical cross-entropy loss, which is appropriate for multi-class classification tasks like this one, where the task is to predict the next word in the sequence. The Adam optimizer was used for training the model, which helps in updating the model weights efficiently.
  Batch size of 32 and 50 epochs were used, with the model being trained on the training set and validated on the validation set.
  The model was evaluated based on BLEU scores, which measure the similarity between the predicted captions and the ground truth captions.

```
Epoch 45/50
240/240 ━━━━━━━━━━━━━━━━━━━━ 105s 437ms/step - accuracy: 0.5221 - loss: 1.7598
Training Accuracy after Epoch 45: 0.5224964022636414
Training Loss after Epoch 45: 1.7604838609695435
Epoch 46/50
240/240 ━━━━━━━━━━━━━━━━━━━━ 106s 443ms/step - accuracy: 0.5251 - loss: 1.7475
Training Accuracy after Epoch 46: 0.5251244306564331
Training Loss after Epoch 46: 1.7493412494659424
Epoch 47/50
240/240 ━━━━━━━━━━━━━━━━━━━━ 106s 441ms/step - accuracy: 0.5267 - loss: 1.7331
Training Accuracy after Epoch 47: 0.5277853608131409
Training Loss after Epoch 47: 1.735107660293579
Epoch 48/50
240/240 ━━━━━━━━━━━━━━━━━━━━ 106s 440ms/step - accuracy: 0.5288 - loss: 1.7263
Training Accuracy after Epoch 48: 0.5293059349060059
Training Loss after Epoch 48: 1.7284491062164307
Epoch 49/50
240/240 ━━━━━━━━━━━━━━━━━━━━ 105s 440ms/step - accuracy: 0.5312 - loss: 1.7180
Training Accuracy after Epoch 49: 0.5314169526100159
Training Loss after Epoch 49: 1.7188665866851807
Epoch 50/50
240/240 ━━━━━━━━━━━━━━━━━━━━ 107s 445ms/step - accuracy: 0.5329 - loss: 1.7080
Training Accuracy after Epoch 50: 0.5334696769714355
Training Loss after Epoch 50: 1.7091437578201294
```

- Evaluation:
  The model's performance was evaluated using two main metrics: accuracy during training and the BLEU score for caption generation.
  1. Training Evaluation: After completing 50 epochs of training, the model achieved a training accuracy of 0.533, indicating a moderate ability to classify the images.

The training loss was 1.709, which represents the model's progress in minimizing error through training.

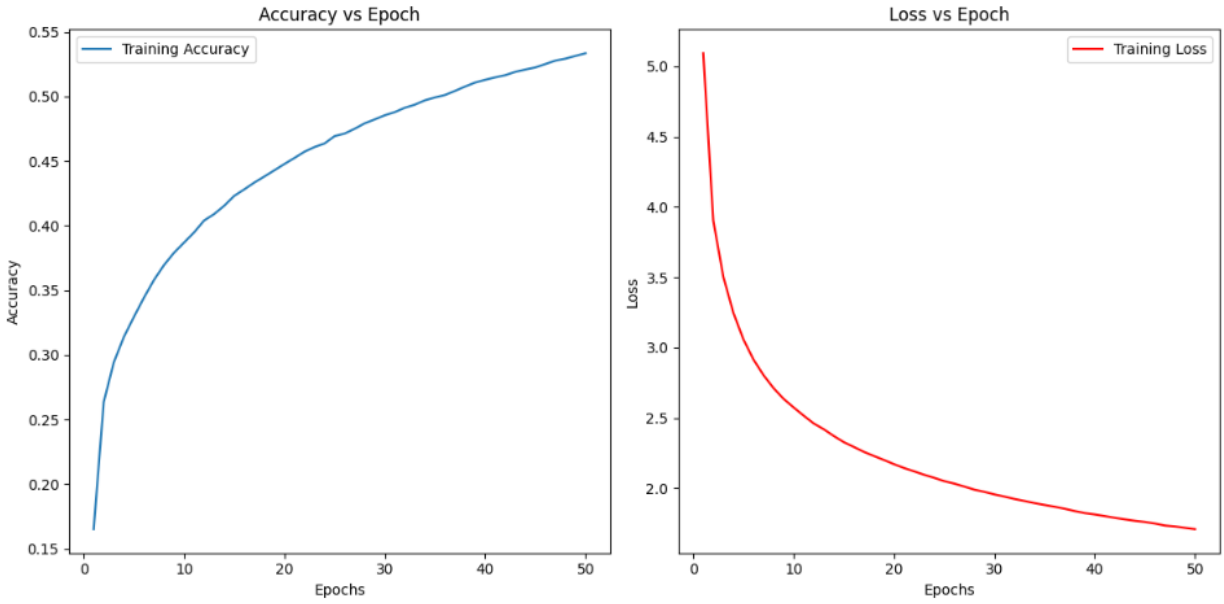
The training process, which took 107 seconds per step, shows the model is learning to map images to captions effectively, but there is room for improvement in both accuracy and loss.

2. BLEU Score Evaluation: After the model was trained, it was evaluated on the test set using the BLEU-1 and BLEU-2 scores, which measure the overlap between the generated and ground-truth captions at different n-gram levels:

BLEU-1: 0.515363

BLEU-2: 0.286102

The BLEU-1 score of 0.515 demonstrates that the model effectively captures individual words in the reference captions, while the BLEU-2 score of 0.286 indicates moderate success in generating bi-gram sequences. These scores suggest that the model produces captions with reasonable word overlap but may still struggle with coherence and grammatical structure.

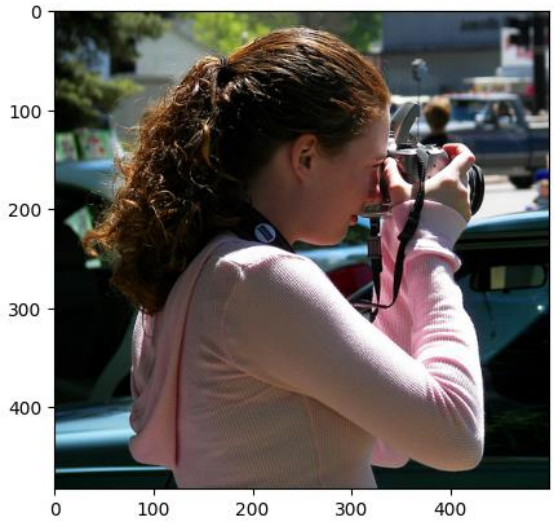


iii. Results

The BLEU-1 score of 0.515 and BLEU-2 score of 0.286 highlight the model's ability to capture key elements of the images but also reveal challenges in generating fluent and contextually accurate descriptions. These outcomes are consistent with the early stages of training a model for complex tasks like image captioning, which often require further tuning and enhancement.

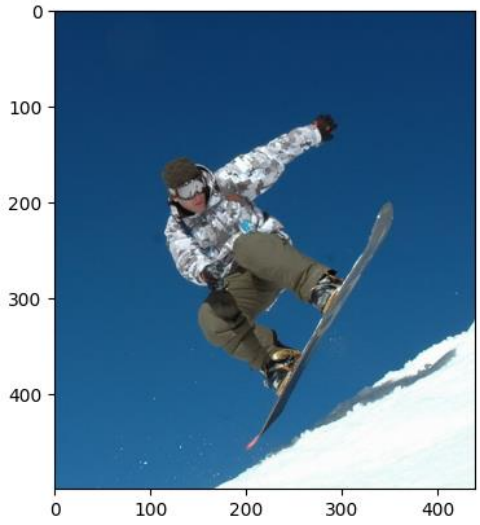
Here’s an example of generating caption for some images that exist in the dataset:

```
the predicted caption:
startseq woman taking picture endseq
```




```
the predicted caption:
startseq frightened man climbing up the rock endseq
```


```
the predicted caption:
startseq man in hoodie is doing trick on white skateboard endseq
```

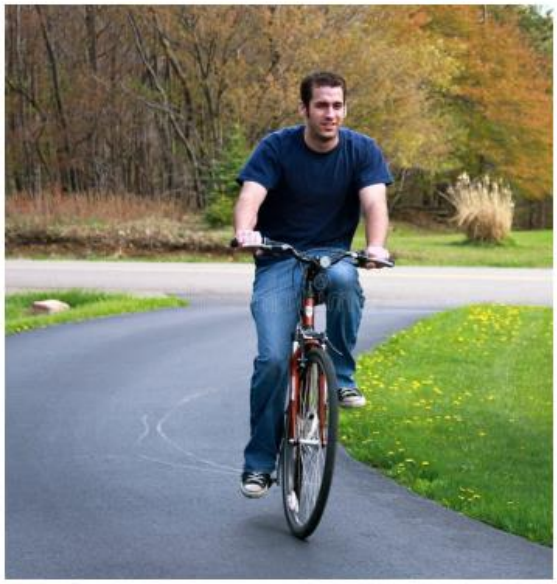



```
the predicted caption:
startseq two women are posing for picture endseq
```

Here’s an example of generating caption for an image that doesn’t exist in the dataset (URL):

'startseq white dog running on grass endseq'

startseq man riding bike in the street endseq



3. Conclusion

This project successfully demonstrated the integration of VGG16 for image feature extraction and LSTM for generating captions in the task of image captioning. The model was able to produce meaningful captions for images, as demonstrated by the BLEU scores achieved during evaluation. The combination of VGG16 and LSTM provided an effective way to handle both visual and textual information, allowing the model to generate coherent captions that describe the content of the images.

While the model performed well, there is still room for improvement. Incorporating more advanced techniques such as attention mechanisms could enhance the model's ability to focus on relevant parts of the image while generating captions. Additionally, experimenting with different datasets or fine-tuning the VGG16 model for specific image captioning tasks could lead to even better results.

In future work, the model could be expanded to handle a larger vocabulary, more complex image scenes, and multiple languages. Additionally, fine-tuning the LSTM and VGG16 models using more computationally intensive techniques such as transfer learning could further improve the model's captioning accuracy.