Alexandria University
Faculty of Engineering
Computer and System Engineering
Department

CSE: Pattern Recognition
Assigned: Tuesday, April 24 2024
Due: Wednesday, May 8, 2024

Assignment #3
Image Captioning Model Implementation

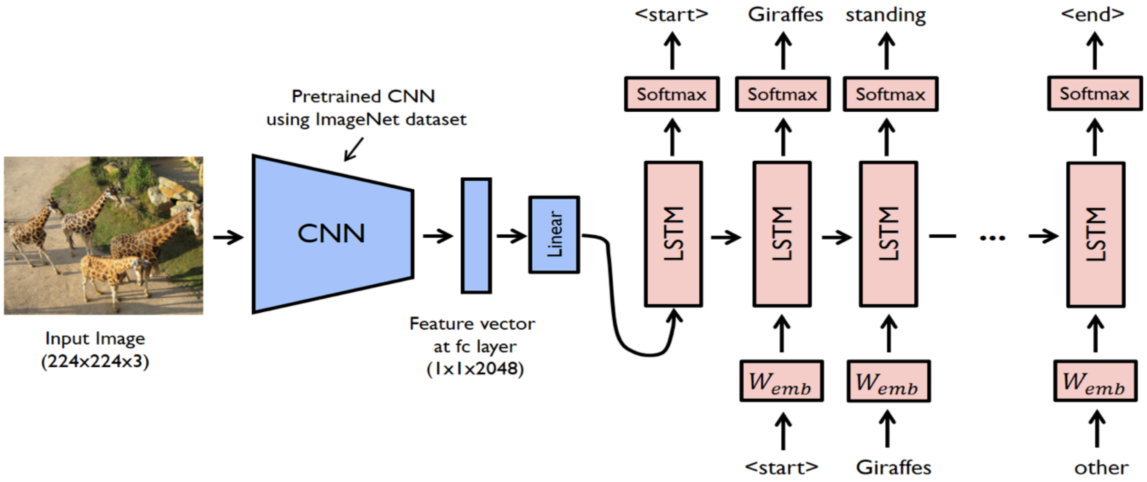

Image captioning:

Image captioning is the process of generating textual descriptions for images automatically. It combines computer vision and natural language processing techniques to understand the content of an image and generate a coherent and relevant description.

Objective:

In this assignment, you will learn to apply Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to image captioning for Flickr8k dataset. By completing this assignment, you will:

1. Gain practical experience in implementing CNN and RNN architectures.
2. Understand the process of feature extraction from images using CNN.
3. Learn how to generate captions for images using RNN particularly LSTM.
4. Analyze model performance using validation and test datasets.
5. Interpret model predictions and identify areas for improvement.

Dr. Marwan Torki

Eng. Ismail El Yamany

Assignment Steps:

Step 1: Dataset Preparation

For this assignment, we will be using the Flickr8k dataset, which consists of 8000 images. Each image in the dataset is associated with five different captions, providing diverse descriptions for the same image. Load the dataset from Hugging Face.
The dataset is divided into three subsets: Training Set Contains 6000 images, Validation Set: Consists of 1000 images. And Test Set: Includes 1000 images.

Before training our image captioning model, it is essential to preprocess the dataset. Here are the key steps involved in dataset preparation:

1) **Load Images:**

   - Load, resize and normalize the images to a suitable format that can be efficiently processed by the CNN model.
   - You should choose a standard size and normalization suitable for your CNN model.

2) **Preprocessing Captions for RNN:**

   From the five captions associated with each image, you can select any one caption for training the model. The RNN component of our model will generate captions based on the features extracted by the CNN. Before feeding the captions into the RNN, preprocess them as follows:

   - Vocabulary: Build a vocabulary by mapping each unique word in the captions to a unique integer index. This mapping facilitates word embedding and decoding during training and inference.
   - Tokenization: Tokenize the captions into words to convert them into a sequence of tokens.
   - Padding: Ensure that all sequences of tokens have the same length by padding shorter sequences with a special token or trimming longer sequences. This ensures uniformity in input size.

3) **Preparing Output Labels:**

   - For training the RNN, we need to prepare output labels for each input sequence. The output labels should represent the next word in the sequence at each sub-iteration. These labels will be used during training to predict the next word given the previous words in the sequence.

Step 2: Convolution Neural Network (ResNet)

- Utilize a **pre-trained ResNet** model on the **ImageNet dataset** and keep its layers frozen.
- Extract image features just before the last layer of classification.
- Add a fully connected layer after the ResNet to train.

Step 3: Recurrent Neural Network (LSTM)

- Utilize an LSTM network for caption generation.
- Utilize cross entropy loss in as your loss function in training your network
- Apply teacher forcing during training, where the ground-truth words are fed as inputs to the LSTM at each time step during training, instead of using the model's own predictions.

Step 4: Training Procedure

- Load the dataset into training, validation, and test sets.
- Train the model using appropriate optimization techniques.
- Monitor **Loss, Accuracy and BLEU score** on validation and report on test sets.

Step 5: Caption Generation

- Showcase examples from the test set where the model performs well.
- Highlight instances where the model performs poorly and analyze the reasons behind it.
- Download images from the internet for testing, Input these images into the trained model and Demonstrate how to generate captions for the input images.

Step 6: Hyperparamter tuning

- Experiment with various model hyperparameters through Hyperparameter Tuning.

Bonus:

- The three groups achieving the highest test BLEU score will be rewarded with a bonus.
- Enhance your model by incorporating *Attention Mechanism*.

Submission Guidelines:

- Work in teams of 3.
- Submit the implemented code along with detailed comments for clarity in a single jupyter notebook.
- Submit your saved jupyter notebook with results in PDF format as your Report.
- Present examples of generated captions with corresponding images.
- Provide analysis and discussion on model performance and potential areas for improvement as comments in the notebook.

Dr. Marwan Torki

Eng. Ismail El Yamany

Resources:

https://huggingface.co/datasets/jxie/flickr8k
https://www.kaggle.com/datasets/adityajn105/flickr8k/data
https://rupamgoyal12.medium.com/image-caption-generator-using-resnet50-and-lstm-model-a5b11f60cd23