Practical Learning in Computer Vision

Final Project - Image Captioning

Name:	**Alon Barak**
Lecturer:	**Gil Ben-Artzi**
TA:	**Evgeny Neiterman**
Date of Per:	2023.06.28
Date of Sub:	2023.07.03

1 Introduction

Image captioning is a prominent task in the field of computer vision and natural language processing, aiming to bridge the gap between visual perception and linguistic understanding. With the rapid advancements in deep learning, specifically in the domain of generative artificial intelligence, the development of image captioning systems has witnessed significant progress. This project focuses on employing an AutoEncoder architecture, comprising a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN), to tackle the challenging task of generating descriptive and coherent captions for images. By combining the power of CNNs in extracting meaningful visual features and the sequential nature of RNNs in generating text, this architecture demonstrates the potential to automatically generate accurate and contextually relevant captions, thus enhancing the comprehension and accessibility of visual content.

2 Data-Set

For this project, the Flickr8k dataset was employed as the primary resource for training and evaluating the image captioning system. The Flickr8k dataset is a widely used benchmark in the field of image captioning, consisting of approximately 8,000 high-quality images collected from the photo-sharing website Flickr. Each image in the dataset is associated with five manually annotated captions, providing diverse and descriptive textual descriptions for various visual scenes. The dataset encompasses a wide range of objects, activities, and contexts, enabling the model to learn the correlation between visual features and their corresponding linguistic expressions. By leveraging the rich and diverse annotations present in the Flickr8k dataset, the image captioning system trained on this data can acquire a broad understanding of different visual concepts and their respective textual representations. The utilization of this comprehensive dataset facilitates the development of a robust and versatile model capable of generating meaningful captions for a wide array of images.

3 Pre-processing

To prepare the image data for the Encoder, a series of pre-processing operations were applied using pre-implemented functions from the PyTorch library (torchvision.transforms). Firstly, the images were resized to a standardized shape of (3, 365, 365). This resizing operation ensures uniformity in the dimensions of the images, allowing the model to process them consistently.

Next, a random crop was performed on the resized images to obtain a smaller size of (3, 300, 300). This random cropping operation helps in introducing variability in the training data by extracting different regions of interest from the images. By doing so, the model learns to capture and understand diverse visual features, enhancing its ability to generate accurate and contextually relevant captions.

Lastly, the image tensors were normalized using the values ((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)). Normalization is a crucial step that scales the pixel values of the image to a standardized range. In this case, the normalization operation brings the pixel values to the range of [-1, 1]. This normalization range helps in stabilizing the training process by ensuring that the input data falls within a reasonable numerical range for the model to process effectively. Additionally, normalizing the image tensors with a mean of 0.5 and a standard deviation of 0.5 assists in reducing the impact of lighting variations and normalizing color intensities across different images, promoting better generalization and robustness of the model.

By performing these pre-processing operations, including resizing, random cropping and normalization, the input images are effectively prepared for the Encoder architecture. These operations contribute to the model's ability to extract meaningful visual features and capture the essential information necessary for generating accurate and coherent captions.

4 Network Architecture

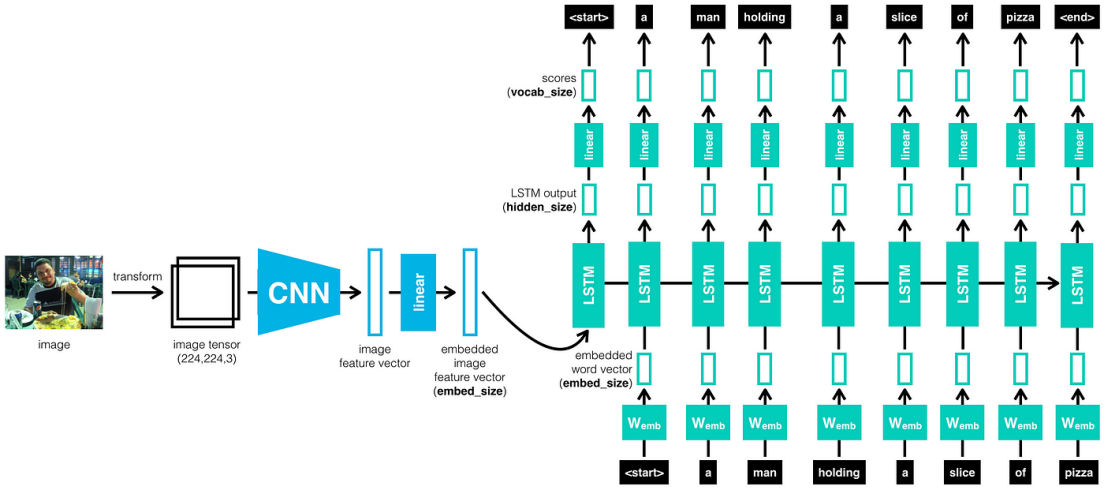
As shown in the figure below (except for input shape) the project utilizes a combination of an Encoder and a Decoder to accomplish the image captioning task. The Encoder is implemented using a Convolutional Neural Network (CNN), specifically a pre-trained Inception V3 model. However, in this project, the last fully connected (FC) layer of the Inception V3 model is re-initialized to facilitate fine-tuning. By re-initializing only the last FC layer, the Encoder can adapt and learn from the specific image captioning task while leveraging the pre-trained weights of the rest of the network. This approach helps in extracting rich visual features from the input image.

On the other hand, the Decoder is constructed using a Recurrent Neural Network (RNN). It comprises several key components. Firstly, an Embedding layer is employed to convert the ground-truth captions into embedded vectors. The embedding layer helps in representing words as dense numerical vectors, capturing their semantic relationships. This allows the model to understand and generate meaningful captions based on the embedded representation of the input text.

Following the embedding layer, a Long Short-Term Memory (LSTM) block with a hidden layer of size 512 is incorporated into the Decoder. The LSTM block is responsible for sequentially processing the embedded vectors and capturing the contextual information. LSTM networks excel at modeling sequential data and are widely used for natural language processing tasks due to their ability to handle long-term dependencies.

Additionally, an additional fully connected (FC) layer is included in the Decoder. This FC layer takes the LSTM outputs and generates a tensor with a shape equal to the vocabulary size. The vocabulary size represents the number of unique words in the dataset. The output tensor from the FC layer serves as the basis for predicting the most probable word in the generated caption at each time step.

The final model combines both the Encoder and the Decoder components. Given an input image of shape (3, 300, 300), the Encoder processes the image and produces an embedding representation. This embedding representation is then passed to the Decoder, which sequentially generates words using the LSTM block and the FC layer, ultimately outputting an embedded vector representation of the predicted sentence. This integrated architecture enables the model to leverage the extracted visual features and generate coherent and contextually relevant captions for the given image.

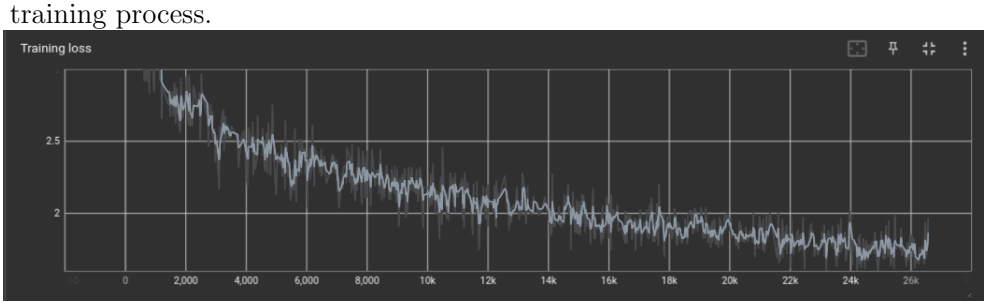


5 Training

During the training process, the AutoEncoder architecture, comprising a CNN Encoder and an RNN Decoder, was utilized for image captioning. The model parameters were optimized using the Adam optimizer and the Cross Entropy (CE) loss function. Specifically, the Decoder parameters and the

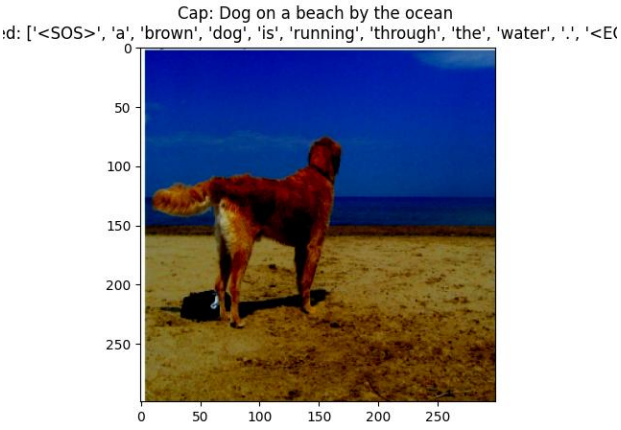
last fully connected (FC) layer of the Encoder were fine-tuned, while the pre-trained weights of the remaining layers in the Inception V3 model were frozen to leverage their learned visual representations.

To facilitate effective learning, the Decoder employed the Teacher Forcing technique. This involved providing the Decoder with the ground truth captions concatenated with the visual features extracted by the Encoder. By exposing the model to the correct captions during training, it learned to associate the visual features with their corresponding textual expressions, enabling it to generate accurate and contextually relevant captions. The model trained for 15 epochs with a batch size of 32 images per iteration due to hardware constraints, optimally it should train for more than 100 epochs with a larger batch size to generate proper results. Below is the Loss graph of the Network during the training process.

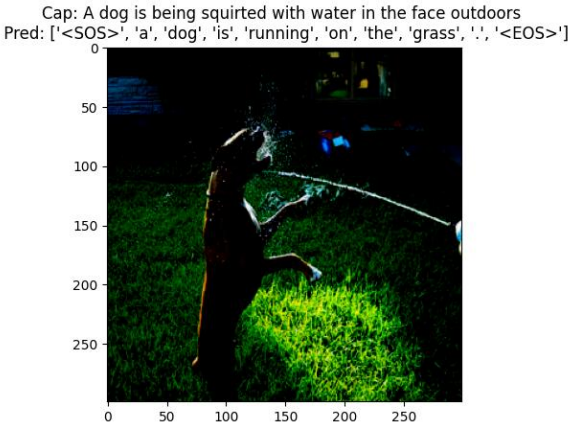


6 Inference & Results

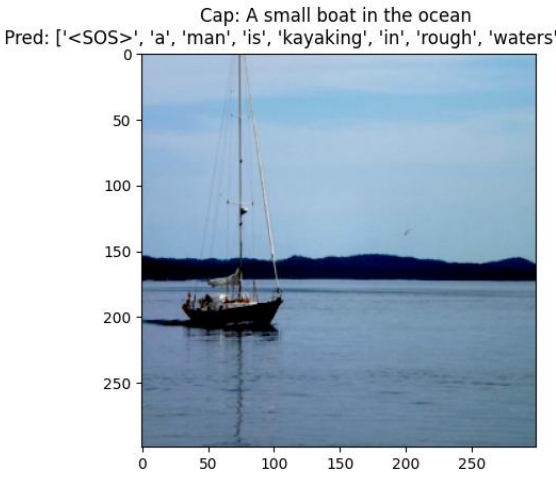
Unlike during training, where the Decoder received ground truth captions, in inference, the model does not have access to the ground truth captions. Instead, the LSTM within the Decoder operates in a sequential manner, predicting each word based on the previously generated words. This process is known as Auto-Regression. The Auto-Regressive nature of the LSTM allows the model to generate captions in a self-guided manner, without relying on ground truth captions. By leveraging its learned knowledge, the model can produce captions that capture the salient features of the input image and convey relevant descriptions. Below, the outputs of the model on test samples.



(a) first



(b) second



(c) third

Figure 1: Results