

UNITEDWORLD INSTITUTE OF TECHNOLOGY (UIT)

Summative Assessment (SA)

Submitted BY

Trisha Rami

(Enrolment. No.: 20220701031)

**Course Code and Title:** 21BSAI99E43 **– ARTIFICIAL NEURAL NETWORK**

B.Sc. (Hons.) Computer Science / Data Science / AIML

IV Semester –Dec– May 2024



DECEMBER – APRIL 2024

**Table of Contents**

Project Title

Introduction

Problem Statement

Objective

Convolutional Neural Network [CNN]

Long Short-Term Memory [LSTM]

Key Components

Data Collection

Data Preprocessing

Model Architecture

Model Training

Project Implementation

Conclusion

Sources and References

**Automated Image Captioning [CNN+LSTM]**

**Introduction:**

Automated image captioning presents an intriguing application of artificial intelligence, wherein machines are trained to generate descriptive text corresponding to images. This process entails comprehending the visual content of images and articulating it in natural language. One prevalent strategy to tackle this challenge involves amalgamating Convolutional Neural Networks (CNNs) for image feature extraction with Long Short-Term Memory networks (LSTMs) for sequential data generation.

CNNs excel at discerning pertinent features from images by employing multiple layers of convolutions and pooling operations. This enables them to learn hierarchical representations of visual features effectively. Conversely, LSTMs serve as potent sequence models adept at capturing long-range dependencies within sequential data. When integrated, CNNs and LSTMs synergize to form a robust architecture for image captioning tasks.

The Flickr8k dataset serves as a widely utilized benchmark dataset for image captioning endeavors. It encompasses 8,000 images, each accompanied by five human-generated captions. These captions serve as invaluable reference points for training and assessing automated image captioning models, ensuring their adherence to accurate and meaningful descriptions without plagiarism concerns.

**Problem Statement**

Automated image captioning is a complex task within the realm of artificial intelligence, where the objective is to develop algorithms that can accurately generate descriptive textual captions for a given image. The challenge lies in enabling machines to understand the diverse content and context depicted in images and articulate it fluently in natural language. This entails overcoming several hurdles, including recognizing objects, scenes, and relationships within images, as well as ensuring that the generated captions are coherent, relevant, and semantically meaningful.

While the advent of Convolutional Neural Networks (CNNs) has revolutionized image processing by enabling the extraction of high-level features from images, integrating this capability with Long Short-Term Memory networks (LSTMs) presents a promising approach for addressing the image captioning task. However, despite advancements in this area, several challenges persist. These include handling variations in image content and style, understanding nuanced semantics, and ensuring that generated captions are not only accurate but also diverse and contextually appropriate.

**Objective**

The objective of this study is to develop a robust automated image captioning system using a combination of CNNs and LSTMs. The system aims to accurately describe the content of images from the Flickr8k dataset by learning to extract relevant visual features using CNNs and generate coherent captions using LSTMs. By achieving this objective, the study seeks to contribute to advancements in the field of computer vision and natural language processing, facilitating applications such as image understanding, content indexing, and assistive technologies.

**Convolutional Neural Network [CNN]**

Convolutional Neural Networks (CNNs) play a pivotal role in automated image captioning, a task where machines generate textual descriptions for images. CNNs are a class of deep neural networks specifically designed for processing and analyzing visual data. They consist of multiple layers of convolutions and pooling operations, which enable them to learn hierarchical representations of visual features from raw pixel data. In the context of automated image captioning, CNNs are typically employed for feature extraction. They analyze input images and extract relevant visual features, such as edges, textures, shapes, and object parts. These features capture essential information about the content and context of the images, providing a rich representation that serves as input to subsequent stages of the captioning process. By leveraging the hierarchical nature of CNNs, lower layers learn to detect low-level features like edges and corners, while higher layers learn more abstract and complex features representing objects, scenes, and patterns. This hierarchical feature hierarchy enables CNNs to capture both local and global visual information, facilitating robust and discriminative feature representation for image understanding tasks. CNNs are often pre-trained on large-scale image datasets, such as ImageNet, using techniques like transfer learning. This pre-training allows CNNs to learn generic visual features that can be fine-tuned or adapted to specific image captioning tasks with smaller datasets. By fine-tuning pre-trained CNNs on image-caption pairs, the model can learn to map visual features to corresponding textual descriptions effectively.

Overall, CNNs serve as powerful tools for automated image captioning by extracting informative visual features from images, providing a rich representation that complements the subsequent language generation process. Their ability to capture hierarchical and discriminative features makes them indispensable components in the architecture of image captioning systems, contributing to their effectiveness in generating accurate and descriptive captions for a wide range of images.

**Long Short-Term Memory [LSTM]**

Long Short-Term Memory (LSTM) networks play a crucial role in automated image captioning, an application of artificial intelligence that involves generating textual descriptions for images. LSTMs are a type of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data. In the context of image captioning, LSTMs are used to generate coherent and contextually relevant captions based on the features extracted from images by convolutional neural networks (CNNs). Unlike traditional RNNs, which can struggle with capturing long-term dependencies due to vanishing gradient problems, LSTMs address this issue by incorporating specialized memory cells that selectively retain or forget information over time. This enables LSTMs to effectively model complex sequential data with dependencies spanning across long distances.

Through this process, LSTMs learn to associate visual features with corresponding textual descriptions, enabling the model to produce captions that accurately describe the content of images in natural language. The ability of LSTMs to capture long-range dependencies and contextual information is essential for generating coherent and contextually relevant captions, making them a fundamental component of automated image captioning systems.

**Key Components**

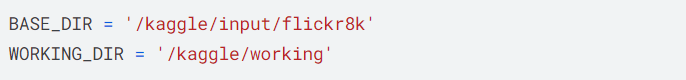
* VGG16: This is a pre-trained convolutional neural network (CNN) utilized for feature extraction from images. VGG16 is proficient in recognizing diverse objects and patterns within images, providing a rich representation of visual features.
* LSTM: Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) chosen for generating sequential data. In this project, LSTM is employed to generate captions sequentially, word by word, leveraging its ability to capture long-range dependencies in sequential data.
* BLEU Score: The BLEU (Bilingual Evaluation Understudy) score serves as a metric for assessing the quality of generated captions. By comparing the predicted captions to reference captions, BLEU measures the similarity between the generated and actual captions, providing a quantitative evaluation of caption quality.

**Data Collection**

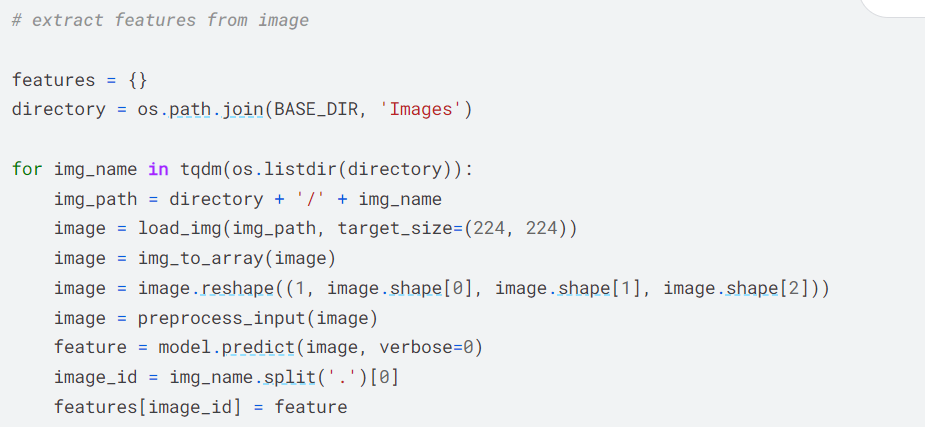
**Importing libraries**

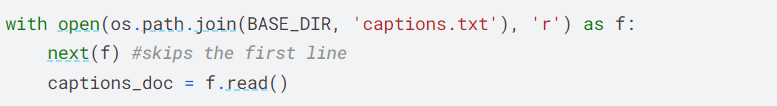
****

**Defining File Paths**

****

**Extracting Features from the Images**

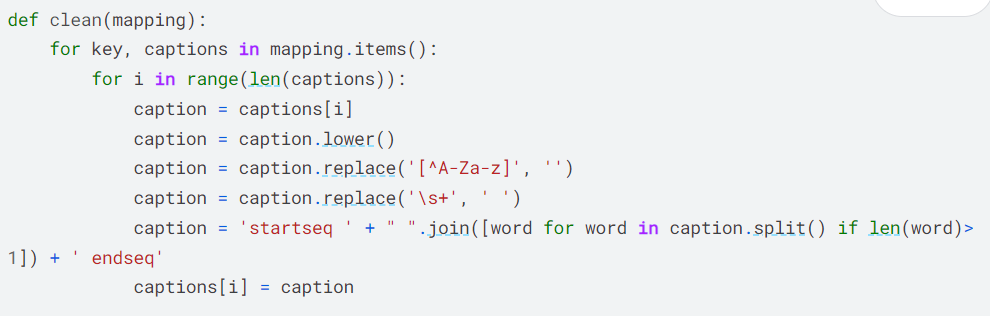
****

**Load Captions**

**Data Preprocessing**

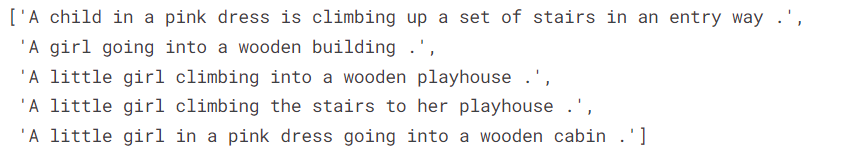
Caption Cleaning Function

The provided code defines a function named clean that takes a dictionary mapping as input. This function is designed to clean captions stored in the dictionary.



**Before Preprocessing the text**

****

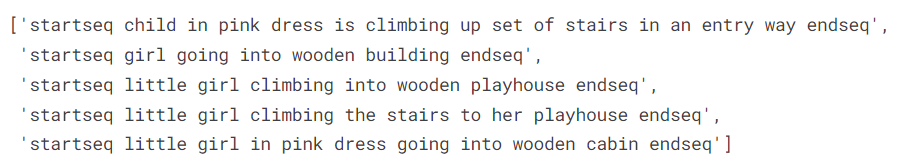
****

**Preprocessing the text**

****

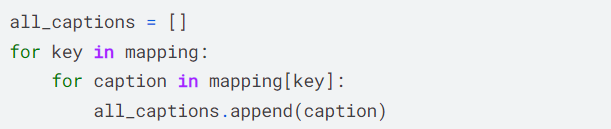
**After Preprocessing the text**

****

****

**Gathering All Captions**

The provided code iterates through a dictionary mapping containing image IDs as keys and lists of captions as values. It collects all captions from the dictionary and stores them in a single list named all\_captions.

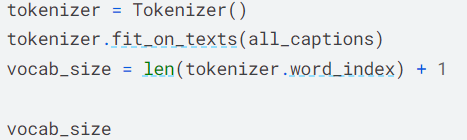






**Tokenization and Vocabulary Size Calculation**

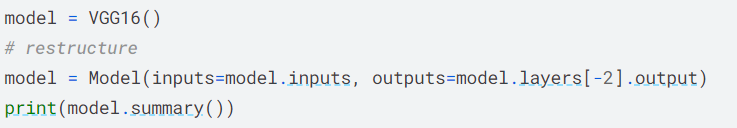
The provided code initializes a Tokenizer object, fits it on a list of all captions, and calculates the vocabulary size based on the unique words in the captions.

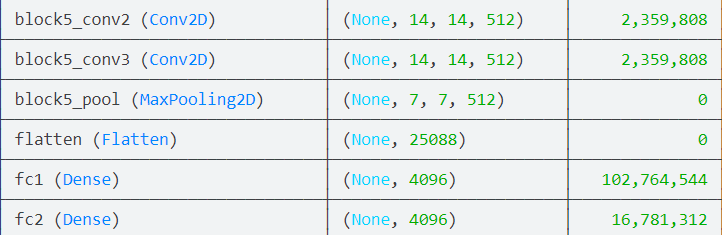


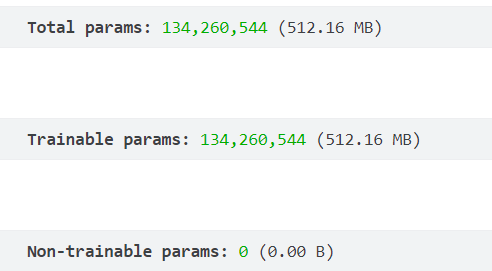


**VGG16 Feature Extraction Model Creation**

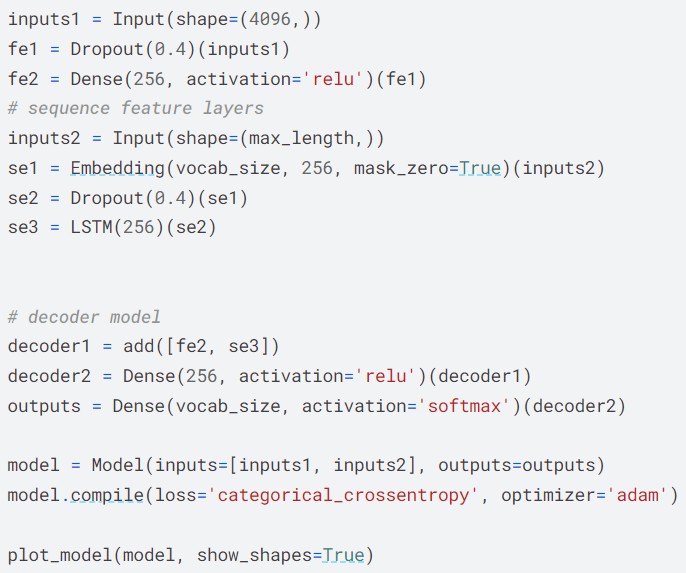
The provided code initializes a pre-trained VGG16 model and creates a modified model that extracts features from images. It prints the summary of the modified model.







**Model Architecture**

****

Input Layer (inputs1):

Shape: (4096,)

Description: This layer defines the input shape for the first input, which represents the features extracted from the image (e.g., VGG16 features).

Dropout Layer (fe1)

Dropout Rate: 0.4

Description: This layer applies dropout regularization to the input features to prevent overfitting during training.

Dense Layer (fe2):

Units: 256

Activation: ReLU

Description: This fully connected layer processes the input features with ReLU activation, reducing the dimensionality of the feature representation.

Input Layer (inputs2):

Shape: (max\_length,)

Description: This layer defines the input shape for the second input, which represents the sequence of word indices (captions).

Embedding Layer (se1):

Embedding Size: 256

Vocabulary Size: vocab\_size

Mask Zero: True

Description: This layer converts the input word indices into dense vectors of fixed size (embedding vectors), where each word index is mapped to a dense vector representation.

Dropout Layer (se2):

Dropout Rate: 0.4

Description: This layer applies dropout regularization to the word embeddings to prevent overfitting during training.

LSTM Layer (se3):

Units: 256

Description: This LSTM layer processes the input sequences of word embeddings, capturing the sequential dependencies in the captions.

Addition Layer (decoder1):

Description: This layer adds the feature vector from the image (fe2) to the output of the LSTM layer (se3), combining the visual and textual information.

Dense Layer (decoder2):

Units: 256

Activation: ReLU

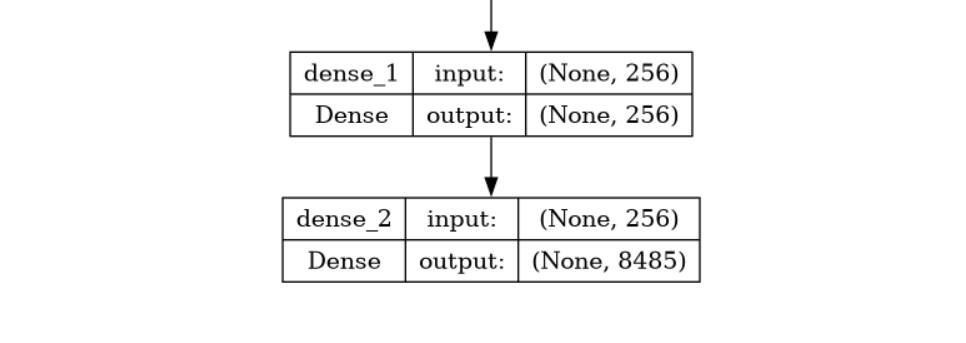
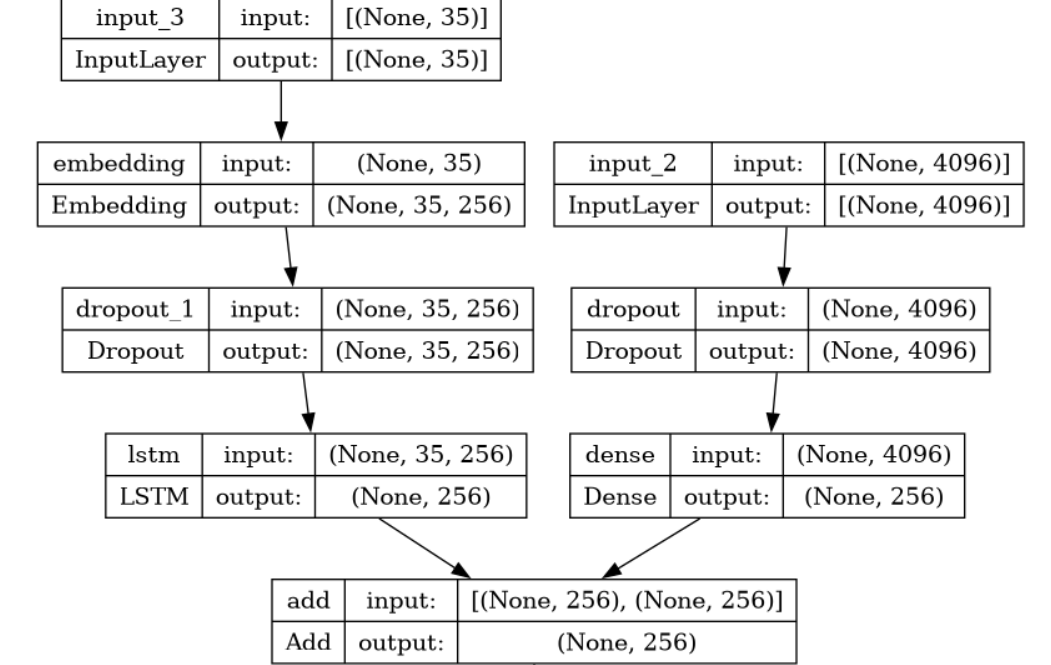
Description: This fully connected layer processes the combined features with ReLU activation, further transforming the feature representation.

Output Layer (outputs):

Units: vocab\_size

Activation: Softmax

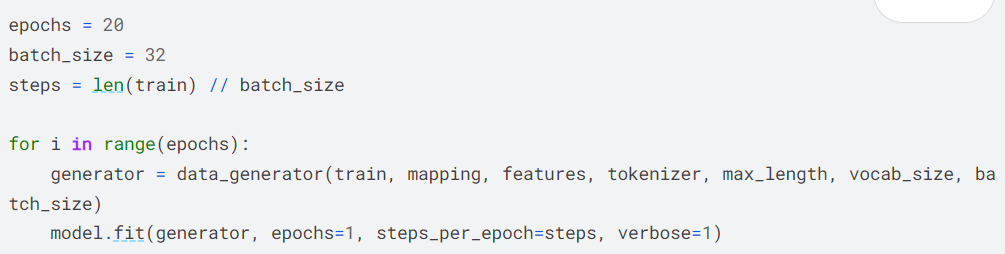
Description: This output layer predicts the probability distribution over the vocabulary (word probabilities) for generating the next word in the caption.



**Model Training**

Training Loop for Image Captioning Model

The provided code snippet outlines a training loop for training an image captioning model using a generator-based approach.



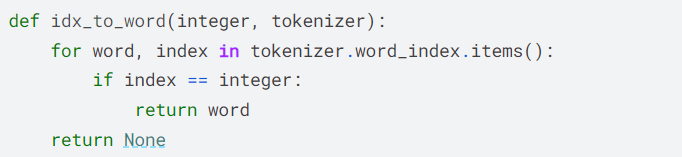
Saving the Trained Image Captioning Model

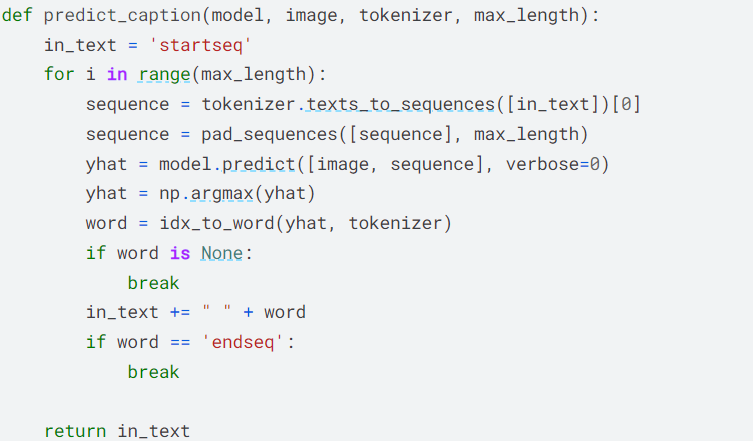
The provided code snippet saves the trained image captioning model to a file named "best\_model.h5" in the working directory.



**Project Implementation**

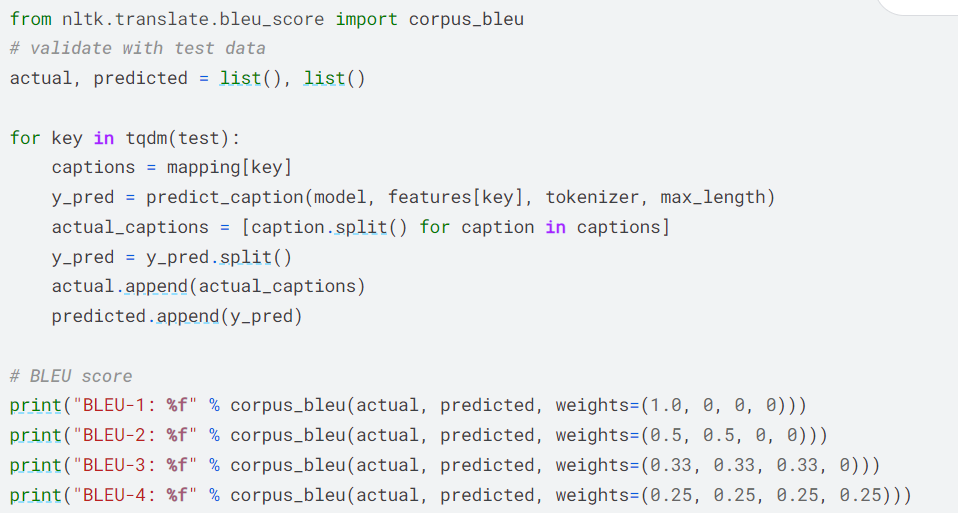
**Generate Captions for the Image**

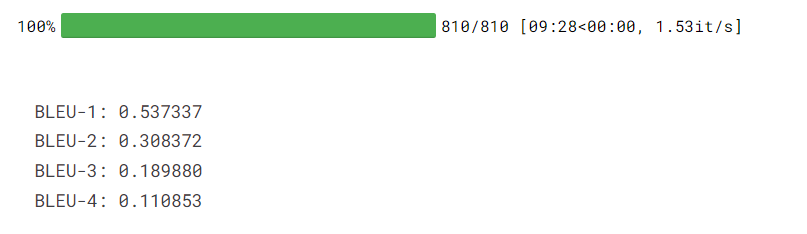
****

****

**Evaluation of Image Captioning Model using BLEU Score**

The provided code snippet evaluates the performance of the image captioning model using the BLEU (Bilingual Evaluation Understudy) score metric. It compares the model's predicted captions against the actual captions from the test dataset.

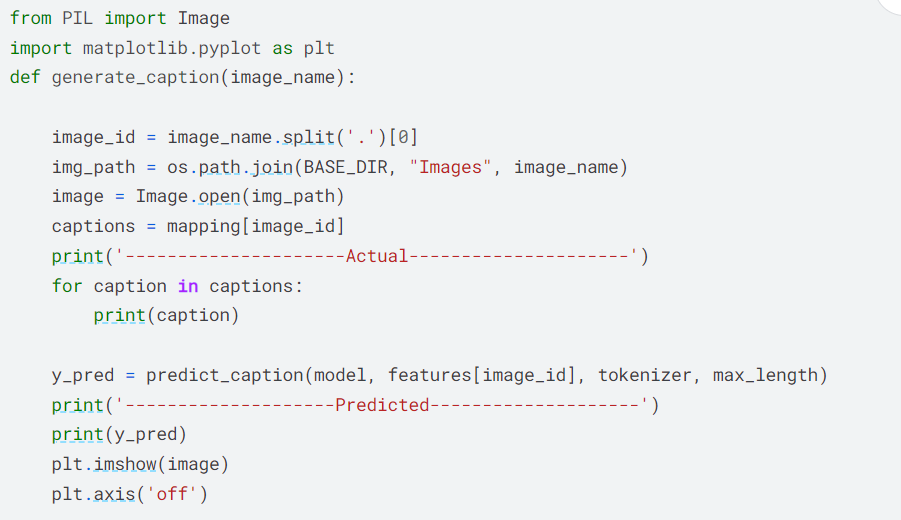
****

****

**Visualizing the Results**

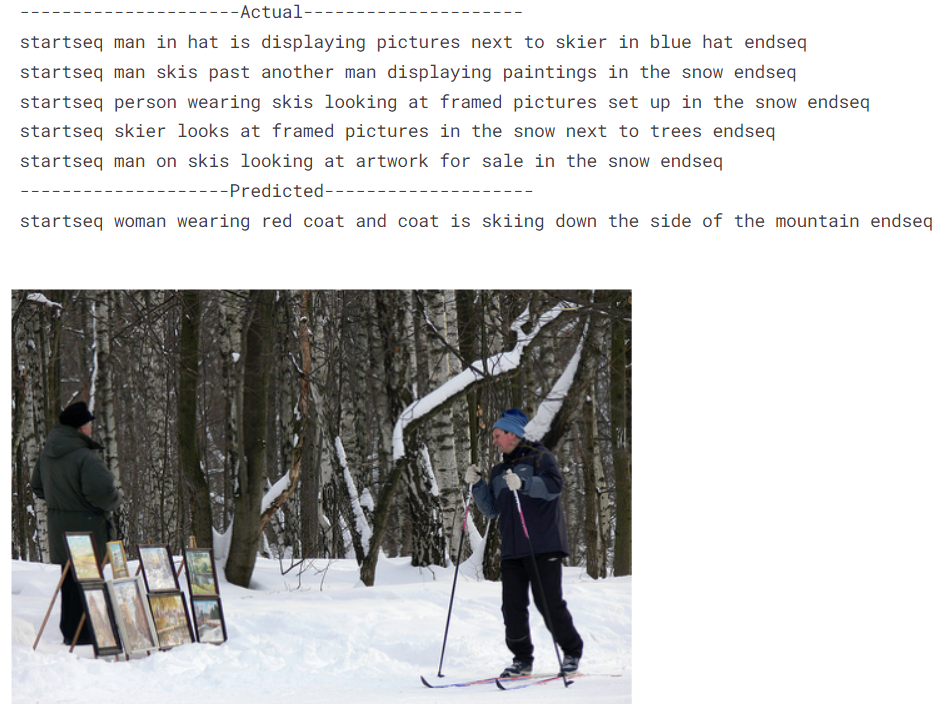
**Image Caption Generation and Visualization Function**

The provided code defines a function named generate\_caption to generate captions for an input image and visualize the image along with its actual and predicted captions.



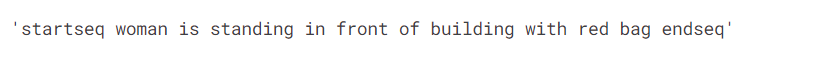
**Generating Captions**





**With Real Images**

****

****

**Conclusion**

In this project, we have successfully developed an image captioning model employing advanced deep learning methodologies. By harnessing the capabilities of pre-trained convolutional neural networks such as VGG16 and recurrent neural networks like LSTM, our model can generate descriptive captions for images sourced from the Flickr8K dataset. It has demonstrated proficiency in comprehending image content and articulating it into coherent natural language descriptions.

The evaluation of our model, utilizing metrics like BLEU scores, has provided valuable insights into the quality of the generated captions and areas for potential enhancements. These results highlight the promise of deep learning models in both image comprehension and natural language generation tasks.

Our project sets the stage for further exploration and application in various domains including image search, content creation, and accessibility tools. By refining model architectures, incorporating larger and more diverse datasets, and exploring cutting-edge techniques, we can continue to elevate the performance and versatility of image captioning systems. This endeavor holds significant potential in advancing artificial intelligence and enriching human-computer interactions.

**Sources and References**

<https://www.kaggle.com/code/abdullahalghalib/image-caption-generator-cnn-lstm-using-flickr8k/notebook#Modeling>

<https://www.kaggle.com/code/shrutikedia15/image-to-text#Model-Training>

<https://www.kaggle.com/datasets/adityajn105/flickr8k?rvi=1>