20/05/2024

MINI PROJECT REPORT

PREPARED BY :

- SEHILI Chaima
- HACHOUD Mohammed
- BELLAGHA Ayoub
- FRIHAOUI Ayoub

Table of Contents

AUTOMATIC IMAGE CAPATIONING USING DEEP LEARNING

Introduction

Automatic image captioning is an innovative technology that combines **Natural Language Processing (NLP)** and **Computer Vision (CV)** to create **descriptive text for images**.

This task involves not only **identifying objects** within an image but also **understanding their relationships and context** to produce accurate and meaningful descriptions.

The core mechanism used in image captioning is **the encoder-decoder framework.**
In the **encoding phase**, **convolutional neural networks (CNNs)** or similar models extract visual features from an image, creating a compact representation of its content.
During the decoding phase, this representation is translated into a **natural language description** using **sequence-generating** models such as **recurrent neural networks (RNNs)**, **long short-term memory networks (LSTMs)**, or **transformers.**

This integration of NLP and CV has a wide range of applications, including **aiding the visually impaired, enhancing image search engines, improving social media experiences, and streamlining content management systems.** Automatic image captioning represents a significant advancement in artificial intelligence, enabling machines to understand and describe visual information in human language.

Problem Statemenet

In today's digital world, transmitting large image files can be slow and bandwidth-intensive, especially in real-time applications. To address this, our project aims to develop an automatic image captioning system that converts images into concise textual descriptions. This approach will reduce data size, enabling faster and more efficient transmission.
Key objectives include:

1. **Real-Time Caption Generation:** Create a system that quickly analyzes images and generates accurate textual descriptions.
2. **Data Compression and Efficiency:** Significantly reduce transmission time and bandwidth usage by sending text instead of large image files.
3. **Context and Accuracy:** Ensure captions accurately reflect image content and context.
4. **Integration and Scalability:** Design the system to integrate easily with existing infrastructures and handle large volumes of images.
5. **User Experience and Reliability:** Provide a consistent and reliable system that performs well under various conditions.

This solution will revolutionize visual data transmission, enabling faster and more efficient communication across various platforms and industries.

ABOUT DATASET

A new benchmark collection for sentence-based image description and search, consisting of **8,000 images** that are each paired with five different captions which provide clear descriptions of the salient entities and events. ...

The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations.

- **Visualizing some of the images along with the corresponding captions**

Caption Text Preprocessing

In the preprocessing stage of our image captioning system, we implement several steps to clean and standardize the textual data. These steps are as follows:

1. **Convert to Lowercase:** Transform all characters in the sentences to lowercase to maintain consistency.
2. **Remove Special Characters and Numbers:** Eliminate any special characters and numbers to simplify the text.
3. **Remove Extra Spaces:** Trim any extra spaces between words to ensure clean, uniform sentences.
4. **Remove Single Characters**: Delete isolated single characters that do not contribute meaningful information.
5. **Add Starting and Ending Tags:** Append specific tags at the beginning and end of each sentence to clearly mark the start and end of the textual data.

These preprocessing steps help in refining the input data, making it more suitable for further processing and analysis in our system.

```
['startseq startseq child in pink dress is climbing up set of stairs in an entry wa
y endseq endseq',
 'startseq startseq girl going into wooden building endseq endseq',
 'startseq startseq little girl climbing into wooden playhouse endseq endseq',
 'startseq startseq little girl climbing the stairs to her playhouse endseq endse
q',
```

Tokenization and Encoded Representation

- The words in a sentence are separated/tokenized and encoded in a one hot representation
- These encodings are then passed to the embeddings layer to generate word embeddings

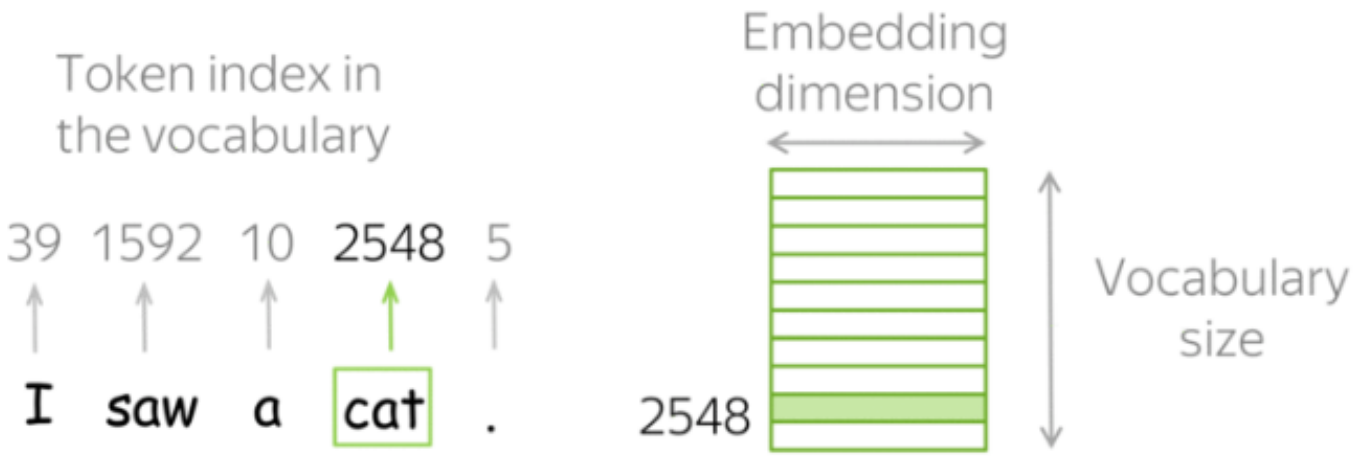
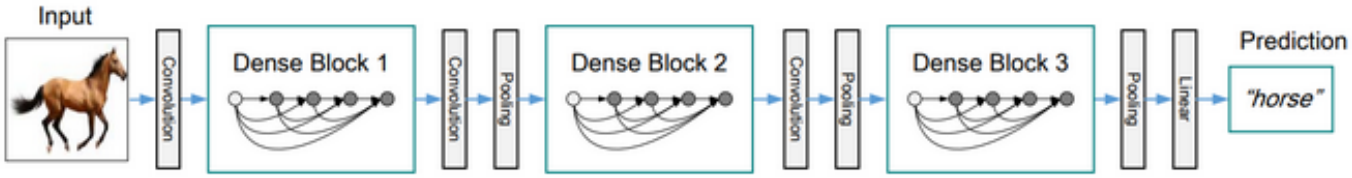


Image Feature Extraction

- **DenseNet 201** Architecture is used to extract the features from the images
- Since the **Global Average Pooling layer** is selected as the final layer of the DenseNet201 model for our feature extraction, our image embeddings will be a vector of size **1920**



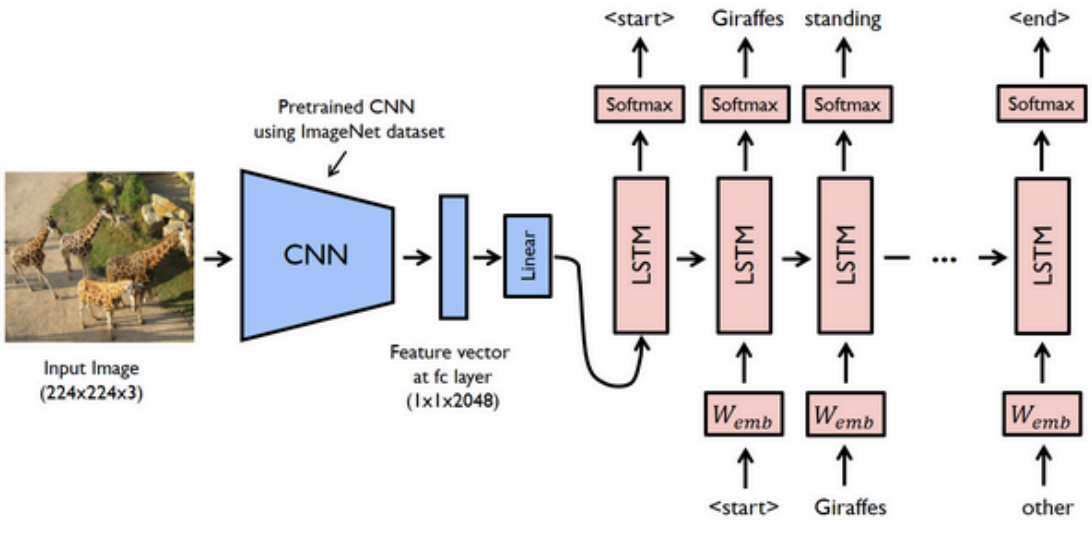
Data Generation

Training an Image Captioning model, like other neural network models, is a resource-intensive process. Due to the large size of the dataset, it is not feasible to load all the data into the main memory at once. Therefore, we **generate the data in the required format batch-wise to optimize memory usage and computational efficiency.**
During the training process, the inputs consist of image embeddings and their corresponding caption text embeddings. These embeddings are processed in batches to facilitate efficient training.
For the caption generation during inference, **the text embeddings are passed word by word.** This sequential processing enables the model to generate coherent and contextually accurate captions for the given images.
This batch-wise **data generation approach** ensures that **our model can handle large datasets** effectively while maintaining high performance and accuracy.

Model Architecture

The Image Captioning model leverages both image and text features to generate accurate captions. The architecture combines a Convolutional Neural Network (CNN) for image feature extraction with a Long Short-Term Memory (LSTM) network for sequence modeling of captions. Additionally, an attention mechanism is applied to align image features with the generated text.

- **Image Model:**
  - **Input**: The model takes an input image embedding of shape (1920,).
  - **Dense Layer:** This layer transforms the image embeddings to a 256-dimensional space with **ReLU** activation.
  - **Repeat Vector:** The image features are repeated to match the length of the caption sequence (max_length).
- **Language Model:**
  - **Input**: Takes caption text embeddings of shape (max_length,).
  - **Embedding Layer:** Converts words into a 256-dimensional space, allowing the model to process the textual data.
  - **LSTM Layers:** The embeddings are passed through LSTM layers to capture the sequence information, with the output being a sequence of 256-dimensional vectors.
  - **TimeDistributed Dense Layer:** Applies a dense layer to each time step of the sequence, maintaining the sequential structure.
- **Attention Mechanism:**
  - Aligns the image features with the corresponding text features, helping the model to focus on relevant parts of the image while generating each word in the caption.
  - The attention output is concatenated with the repeated image features.
- **Sequence Processing:**
  - **LSTM Layers:** The combined features are processed through additional **LSTM** layers, converting the merged sequence into a fixed-length vector.
  - **Dropout Layer**: Applied to prevent overfitting.
  - **Residual Connection:** The original image features are added to the LSTM output to enhance the representation.
  - **Dense Layer:** Processes the combined features through a dense layer with ReLU activation.
  - **Final Output:** The processed features are passed through a dense layer with softmax activation to generate the final word probabilities for the caption.



Compilation and Training:

The Image Captioning model is compiled and trained using several key components to ensure efficient training and evaluation, as well as to handle overfitting and learning rate adjustments dynamically.

- **Compilation**

The model is compiled with the following configurations:

  - **Loss Function: categorical_crossentropy,** which is suitable for multi-class classification problems.
  - **Optimizer: adam,** an adaptive learning rate optimization algorithm.

- **Data Generators**

Custom data generators are used to load the training and validation data in batches, which helps in managing memory efficiently:

  - **Training Generator:** Loads training data in batches of 64 images and their corresponding captions.
  - **Validation Generator:** Loads validation data similarly in batches of 64 images and captions.

- **Callbacks**

To enhance the training process, several callbacks are employed:

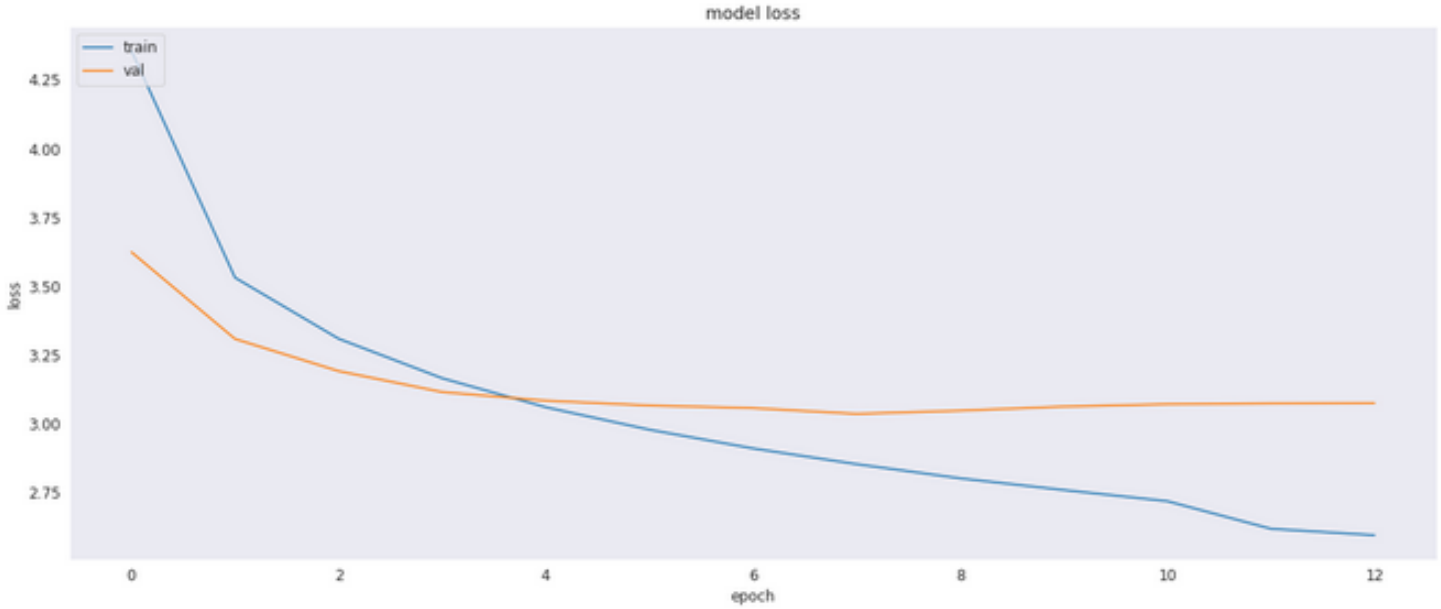
  - **Model Checkpoint:** Saves the model with the best validation loss during training.
  - **Early Stopping:** Stops the training early if the validation loss does not improve for 5 consecutive epochs, restoring the best weights.
  - **Learning Rate Reduction:** Reduces the learning rate if the validation loss plateaus, helping the model to converge more effectively.

- **Training**

The model is trained with the following configurations:

  - **Epochs:** The training is set to run for **50** epochs.
  - **Training Data:** Provided by the train_generator.
  - **Validation Data:** Provided by the validation_generator.
  - **Callbacks**: Includes model checkpointing, early stopping, and learning rate reduction.

Learning curves:



Evaluation

Run all

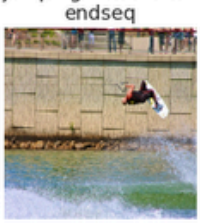
startseq man is standing on the snow endseq

startseq man is standing on the mountain endseq

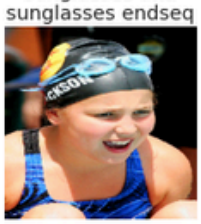
startseq man in blue shirt is standing in the grass endseq

startseq man in blue shirt is standing in front of the window endseq

startseq man is jumping into the air endseq



startseq man with sunglasses and sunglasses endseq



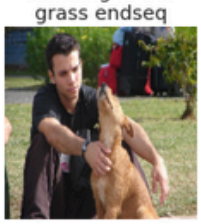
startseq man in blue shirt is walking down the street endseq

startseq two dogs are running through the grass endseq

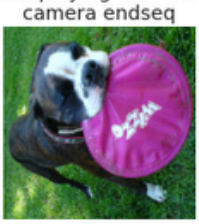
startseq two dogs are running in the grass endseq

startseq two children are playing in the snow endseq

startseq dog is standing in the grass endseq



startseq two dogs are playing with the camera endseq

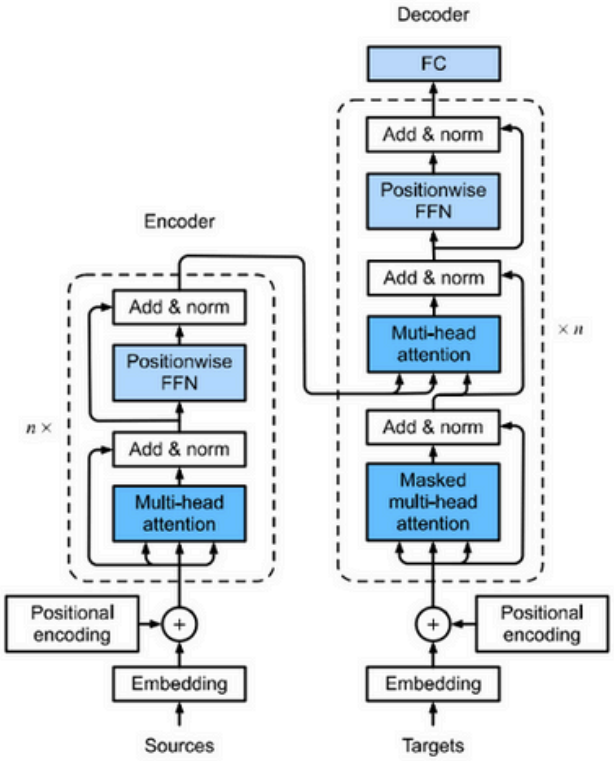


startseq man in blue shirt is standing in the street endseq

startseq man in red shirt is walking down the street endseq

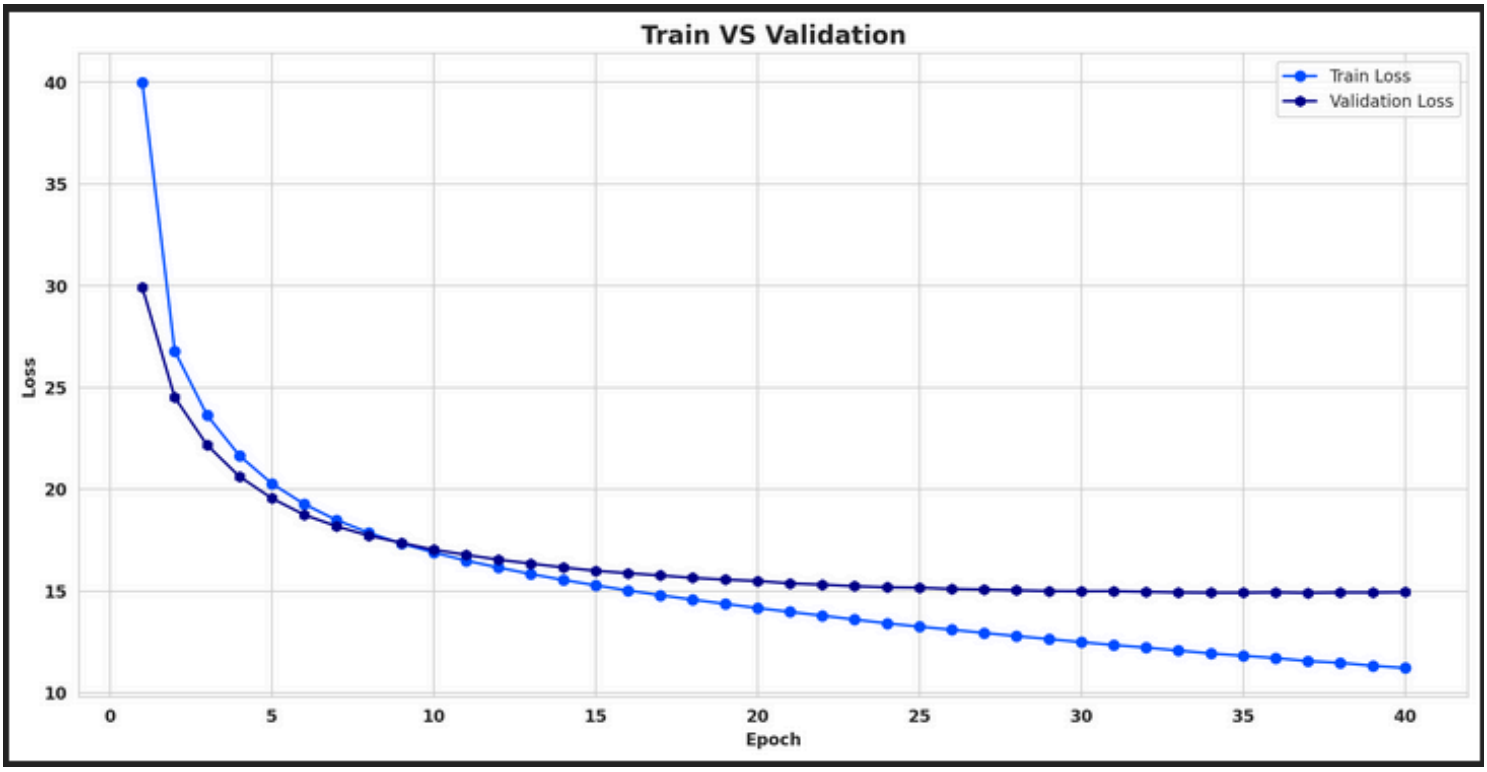
startseq man in black shirt and glasses and glasses is standing in front of the camera endseq

Transformers Based Model Architecture



The model architecture comprises a pre-trained EfficientNetB0 CNN for image feature extraction. The extracted features undergo normalization and dense layers to align dimensions with the encoder. The encoder incorporates multi-head attention and layer normalization. The decoder, fed with the target caption and encoder's output, employs positional embedding, masked multi-head attention, layer normalization, multi-head cross-attention, feed-forward layers. Ultimately a Dense layer with softmax activation serves as the output layer.

Learning curves:



Evaluation

Generated Caption: a group of people are standing in a river

Generated Caption: a man and a woman are sitting in a room with a stuffed animal

Conclusion and Future Work

Our proposed model leverages a combination of **CNNs** and **LSTMs** with **attention mechanisms** to generate captions for images. The **CNN** encodes the image into a compact feature representation, which is then used by the **LSTM** network to generate corresponding sentences. The model has shown promising results, producing accurate and contextually relevant captions for a variety of images. However, the quality of the input images significantly impacts feature extraction and caption generation, indicating areas for **further enhancement.** While the current model performs well, there are several avenues for future improvement:

- **Larger Dataset:** Expanding the dataset can help the model learn more diverse features, improving its ability to generalize across different types of images.
- **Beam Search:** Implementing Beam Search during the caption generation process can enhance the quality of the generated captions by considering multiple possible sequences and selecting the most likely one.
- **Performance Metrics:** Incorporating additional performance metrics such as **BLEU** scores can provide a more comprehensive evaluation of the model's accuracy and
- **Quality.Text-to-Speech Integration:** Adding text-to-speech capabilities can make the model more versatile, enabling it to generate audible descriptions of images.

By addressing these aspects, we can further refine our model to produce more accurate and reliable image captions, enhancing its applicability in real-world scenarios.