الجامعة الأورومتوسطية بفاس
EURO-MEDITERRANEAN UNIVERSITY OF FES
UNIVERSITÉ EURO-MÉDITERRANÉENNE DE FÈS

École d'Ingénierie Digitale
et d'Intelligence Artificielle

End of the Year Project

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Application of a research paper

Prepared By:
GERBOUZI Firdawse
EDDOUKS Oumayma
EL OMARY IMANE
EL HARRARI Anass
CHAHDI Ghita
EL BOUKILI IMANE

supervisor:
Pr.AMMOUR Alae

Academic year: 2022 - 2023

Acknoledgment

We would like to express our sincere appreciation and acknowledgement to Professor Alae Amour for their invaluable guidance and support throughout the creation of our PFA report on image captioning.

Professor Amour's expertise, knowledge, and mentorship have been instrumental in shaping the success of our project. Their dedication and commitment to our learning experience have truly made a significant impact.

Throughout the project, Professor Alae Amour provided valuable insights, constructive feedback, and encouragement that pushed us to excel in our research. Their meticulous attention to detail and rigorous evaluation helped enhance the quality of our work.

We deeply appreciate our Professor 's openness and responsiveness in addressing our questions and concerns. Their guidance went beyond the technical aspects of the project, encompassing valuable teachings on research methodology and professionalism. We consider ourselves fortunate to have been able to learn from such an outstanding educator and researcher.

Abstract

The paper titled "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" presents a novel approach for generating image captions using neural networks with visual attention mechanisms. The authors draw inspiration from recent advancements in machine translation and propose an attention-based model that learns to automatically describe the content of images. The model is trained using a combination of deterministic and stochastic approaches, leveraging standard backpropagation techniques and maximizing a variational lower bound. Through visualization, the authors demonstrate the model's ability to focus on salient objects while generating corresponding words in the output sequence. The effectiveness of the attention mechanism is validated through experiments on the COCO dataset, showcasing state-of-the-art performance in image caption generation. This paper contributes to the field by introducing a powerful framework that combines attention mechanisms and neural networks to improve the quality and accuracy of image captioning systems.

Contents

1 Introduction **7**
- 1.1 Project Overview 7
- 1.2 Objectives 7
- 1.3 Scope and Limitations 7

2 Literature Review **9**
- 2.1 Image Captioning 9
  - 2.1.1 Definition and Background 9
  - 2.1.2 Applications 9
  - 2.1.3 State-of-the-Art Techniques for Image Captioning 10
    - 2.1.3.1 Review of the literature 10
    - 2.1.3.2 State-of-the-Art Architectures 11

3 Datasets **15**
- 3.1 Our Dataset 15
  - 3.1.1 Dataset Description 15
    - 3.1.1.1 MS COCO 15
    - 3.1.1.2 Flickr8k 15
  - 3.1.2 Data Preprocessing Techniques 16

4 Theory and Techniques **18**
- 4.1 Theory and Techniques 18
  - 4.1.1 Overview of Image Captioning Techniques 18
  - 4.1.2 Convolutional Neural Networks (CNN) 19
    - 4.1.2.1 Convolutional Neural Network Architecture 19
  - 4.1.3 Recurrent Neural Networks RNN and Long Short-Term Memory LSTM 22
  - 4.1.4 Attention Mechanisms 23
    - 4.1.4.1 What is Attention? 23
    - 4.1.4.2 Why Attention? 23
    - 4.1.4.3 How does Attention work? 24
    - 4.1.4.4 subsection Learning Stochastic “Hard” vs Deterministic “Soft” Attention 26
  - 4.1.5 Evaluation Metrics 28

5 Image Captioning Model **29**
- 5.1 Defining the Model 29

6 Results and Analysis **31**
6.1 quantitative evaluations 31
6.1.1 Comparative Analysis 31
6.1.2 Visualization Analysis 32
6.1.3 qualitative evaluations 33

7 Discussion **35**
7.1 Limitations and Challenges 35
7.2 Future Directions and Improvements 35

8 Implementation **37**
8.1 Development Environment and Tools 37

9 Conclusion **39**
9.1 Summary of Findings 39
9.2 Final Remarks 39

List of Tables

2.1 Important works in the field of image captioning 10

6.1 Performance Metrics on Flickr8k and Microsoft COCO datasets . . . 31

List of Figures

2.1 Image Feature Encoder connected to Text Generator 11
2.2 Merge architecture (Image by Author) 11
2.3 Object Detection backbone (Image by Author) 12
2.4 Image Caption architecture with Attention (Image by Author) 13
2.5 Image Caption architecture with Transformer (Image by Author) . . 13
2.6 Dense Caption (Source, by permission of Prof Fei-Fei Li and Justin Johnson) 14

3.1 Text Preprocessing and Image Processing Steps 17

4.1 Representation of image as a grid of pixels (Source) 19
4.2 Architecture of a CNN (Source) 19
4.3 Convolution Operation Formula 20
4.4 Pooling Operation 21
4.5 Basic RNN Structure 22
4.6 Pooling Operation 24
4.7 Pooling Operation 25
4.8 Pooling Operation 25

5.1 Pooling Operation 29
5.2 Model Architecture 30

6.1 loss curve of Flickr8k 32
6.2 loss curve of COCO 32
6.3 Example Image Caption Generation with Visual Attention. 34

8.1 Result of an Image from the Flickr8k Dataset 38
8.2 Result of an Image from the COCO Dataset 38

Chapter 1

Introduction

1.1 Project Overview

The project focuses on developing a model for generating image captions using neural networks with a visual attention mechanism. The project combines convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to automatically generate descriptive and contextually relevant captions for images. The visual attention mechanism allows the model to attend to different regions of the image during caption generation, enhancing the alignment between the generated words and relevant visual content. By leveraging deep learning techniques and attention mechanisms, the project aims to advance the field of image captioning and achieve state-of-the-art performance on the widely used MS COCO dataset.

1.2 Objectives

The main objective of the project is to create a model that can generate accurate and meaningful captions for images. The model should possess the ability to understand the visual content of an image and generate captions that capture relevant details and contextual information. By combining CNNs and RNNs, the project aims to leverage the strengths of both architectures in visual feature extraction and sequential caption generation, respectively. Furthermore, the incorporation of a visual attention mechanism aims to improve the model's ability to align words with relevant image regions, leading to more precise and context-aware captions. Overall, the project seeks to push the boundaries of image captioning quality and effectiveness.

1.3 Scope and Limitations

The scope of the project lies in generating captions for a wide range of images, covering diverse visual content. The model aims to accurately capture the semantics and details of the images to produce meaningful captions. The project demonstrates its effectiveness on the MS COCO dataset, indicating its potential applicability to real-world scenarios. However, the project also has limitations to consider. One limitation is the challenge of handling complex scenes and images with intricate visual contexts. The model may encounter difficulties in generating captions for images containing multiple objects or complex relationships between them. Additionally, the model's performance can be affected by the ambiguity present in some visual contexts, which may result in less accurate captions. Furthermore, the model's performance heavily relies on the quality, diversity, and representativeness of the training data. Inadequate or biased training data can limit the model's ability to generalize well. Addressing these limitations requires further research and improvements to enhance the model's robustness and generalization capabilities.

Chapter 2

Literature Review

2.1 Image Captioning

2.1.1 Definition and Background

Image captioning is the process of automatically generating descriptive and contextually relevant textual captions for images. It involves using advanced computational techniques, such as deep learning and natural language processing, to analyze the visual content of images and generate captions that effectively describe the objects, scenes, and concepts depicted in the images. In recent years, Image captioning has gained significant attention for its applications in various domains, such as assistive technology, content retrieval, and social media analysis. The advancements in deep learning models, including CNNs for visual feature extraction and RNNs for sequential caption generation, have contributed to the progress of image captioning. Attention mechanisms have been integrated into models to improve the quality of generated captions by focusing on salient image region.

2.1.2 Applications

Image captioning has several valuable applications. Firstly, it contributes to accessibility by providing textual descriptions for individuals with visual impairments or limited visual perception. By automatically generating captions for images, it enables visually impaired individuals to access and understand visual content. Secondly, image captioning enhances content understanding by summarizing the visual content of an image through descriptive captions. This is particularly useful in applications such as image search engines or content recommendation systems, where captions assist users in quickly comprehending the essence of an image. Thirdly, image captioning improves human-machine interaction by enabling machines to comprehend and respond to visual content. By generating captions that describe images, machines can communicate more effectively with humans, resulting in more intuitive and interactive user experiences. Furthermore, image captioning has significance in social media platforms and image sharing websites, as it allows users to add captions to their images, enhancing engagement, searchability, and overall user experience. Lastly, image captioning finds application in content generation, such as generating captions for news articles, blog posts, or storytelling applications. By automatically generating captions for images, it aids in creating engaging and informative content across various domains and industries. Overall, image captioning has diverse and impactful applications that improve accessibility, content understanding, human-machine interaction, social media experiences, and content generation.

2.1.3 State-of-the-Art Techniques for Image Captioning

Image captioning is essentially a problem of computer vision and natural language processing, where the challenge is to understand the content of an image and describe it in natural language.

2.1.3.1 Review of the literature

Here are some influential papers and art of state approaches:

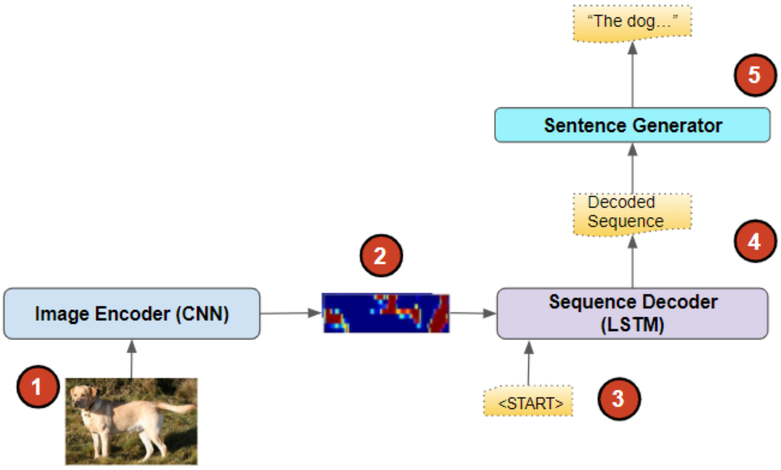
Title	**Year**	**Description**
Show and Tell: A Neural Image Caption Generator	2015	This paper by Vinyals et al. from Google introduced a model that combines a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN). The CNN acts as an "encoder" and the RNN acts as a "decoder". This architecture was the basis for many subsequent developments. [1]
Show, Attend and Tell: Neural Image Caption Generation with Visual Attention	2016	This paper by Xu et al. introduced the concept of attention to image captioning. The model learns to focus on specific parts of the image while generating each word of the caption, leading to better results. [2]
Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering	2018	Anderson et al. proposed a new approach that combines bottom-up (object-level) and top-down (semantic-level) attention mechanisms. This model achieved state-of-the-art results on the COCO image captioning benchmark at the time of its publication.[3]
Meshed-Memory Transformer for Image Captioning	2020	Cornia et al. introduced a model that integrates a multi-head self-attention mechanism with a memory component, allowing it to generate captions that are more related to the image content and more fluent. [3]
VirTex: Learning Visual Representations from Textual Annotations	2021	This work by Desai et al. proposed a pretraining task that learns to generate captions from images, and uses these learned representations for downstream tasks. This method showed that self-supervised learning with textual annotations can significantly improve performance on visual recognition tasks. [4]

Table 2.1: Important works in the field of image captioning

2.1.3.2 State-of-the-Art Architectures

- Architecture: Encoder-Decoder

Perhaps the most common deep learning architecture for Image Captioning is sometimes called the “Inject” architecture and directly connects up the Image Feature Encoder to the Sequence Decoder, followed by the Sentence Generator, as described above. adjustbox



max width=

Figure 2.1: Image Feature Encoder connected to Text Generator

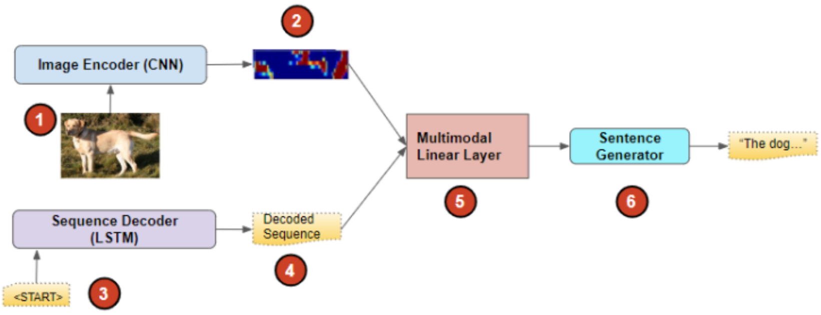
- Architecture: Multi-Modal

The Inject architecture was the original architecture for Image Captioning and is still very popular. However, an alternative which gets called the “Merge” architecture has been found to produce better results.

Rather than connecting the Image Encoder as the input of the Sequence Decoder sequentially, the two components operate independently of each other. In other words, we don’t mix the two modes ie. images with text.

the CNN network processes only the image and the LSTM network operates only on the sequence generated so far.

adjustbox



max width=

Figure 2.2: Merge architecture (Image by Author)

The outputs of these two networks are then combined together with a Multimodal layer (which could be a Linear and Softmax layer). It does the job of interpreting both outputs and is followed by the Sentence Generator that produces the final predicted caption.

Another advantage of this approach is that it allows us to use transfer learning not just for the Image Encoder but for the Sequence Decoder as well. We can use a pre-trained language model for the Sequence Decoder.

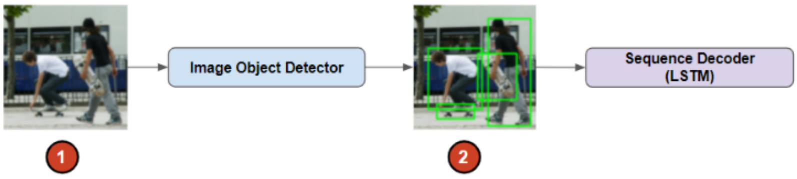
Many different ways of combining the outputs have been tried eg. concatenate, multiplication, and so on. The approach that usually works best is to use addition.

- Architecture: Object Detection backbone

Earlier we talked about using the backbone from a pre-trained Image Classification model for the Image Encoder. This type of model is usually trained to identify a single class for the whole picture.

However, in most photos, you are likely to have multiple objects of interest. Instead of using an Image Classification backbone, why not use a pre-trained Object Detection backbone to extract features from the image?

adjustbox



max width=

Figure 2.3: Object Detection backbone (Image by Author)

The Object Detection model generates bounding boxes around all the prominent objects in the scene. Not only does it label multiple objects, but it identifies their relative positions within the picture. Thus it is able to provide a richer encoded representation of the image, which can then be used by the Sequence Decoder to include a mention of all of those objects in its caption.

- Architecture: Encoder-Decoder with Attention

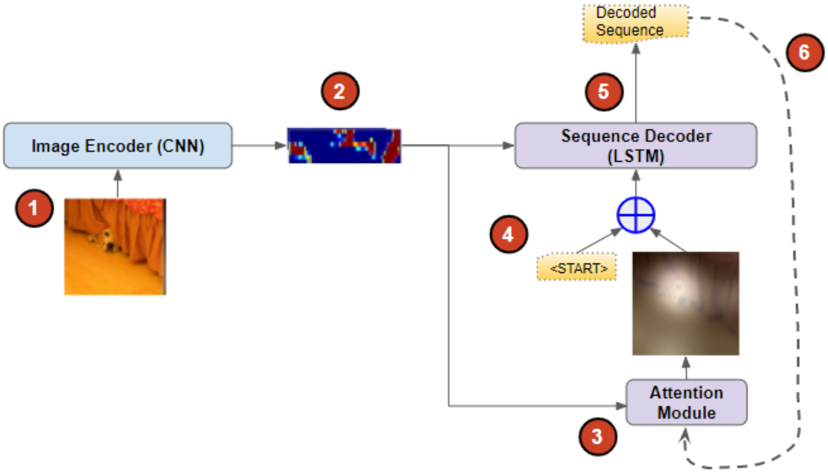
Over the last few years, the use of Attention with NLP models has been gaining a lot of traction. It has been found to significantly improve the performance of NLP applications. As the model generates each word of the output, Attention helps it focus on the words from the input sequence that are most relevant to that output word.

It is therefore not surprising to find that Attention has also been applied to Image Captioning resulting in state-of-the-art results.

As the Sequence Decoder produces each word of the caption, Attention is used to help it concentrate on the part of the image that is most relevant to the word it is generating.

adjustbox

The Attention module takes the encoded image vector along with the current output token from the LSTM. It produces a weighted Attention score. When that



max width=

Figure 2.4: Image Caption architecture with Attention (Image by Author)

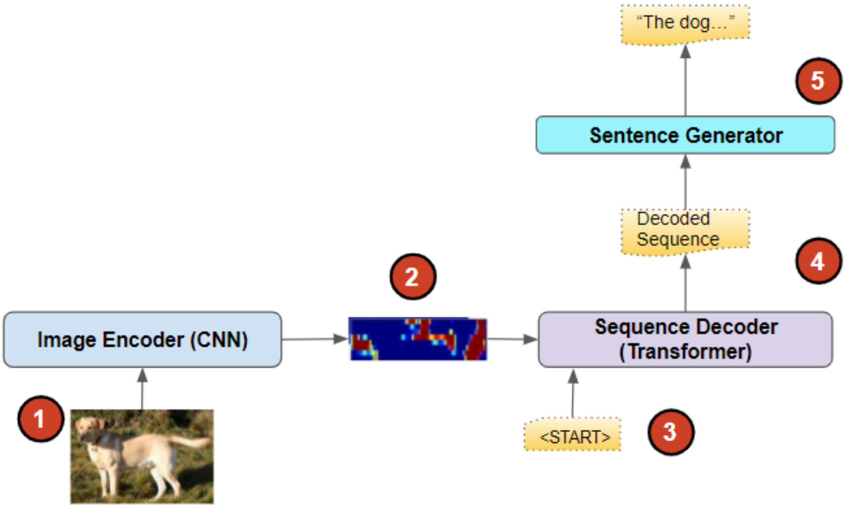
score is combined with the image it boosts the weight of those pixels that the LSTM should focus on while predicting the next token.

For instance, for the caption “The dog is behind the curtain”, the model focuses on the dog in the photo as it generates the word ‘dog’ and then shifts its focus to the curtain when it reaches the word ‘curtain’, as you would expect.

- Architecture: Encoder-Decoder with Transformers

When talking about Attention, the current giant is undoubtedly the Transformer. It revolves around Attention at its core and does not use the Recurrent Network which has been an NLP mainstay for years. The architecture is very similar to the Encoder-Decoder with the Transformer replacing the LSTM.

adjustbox



max width=

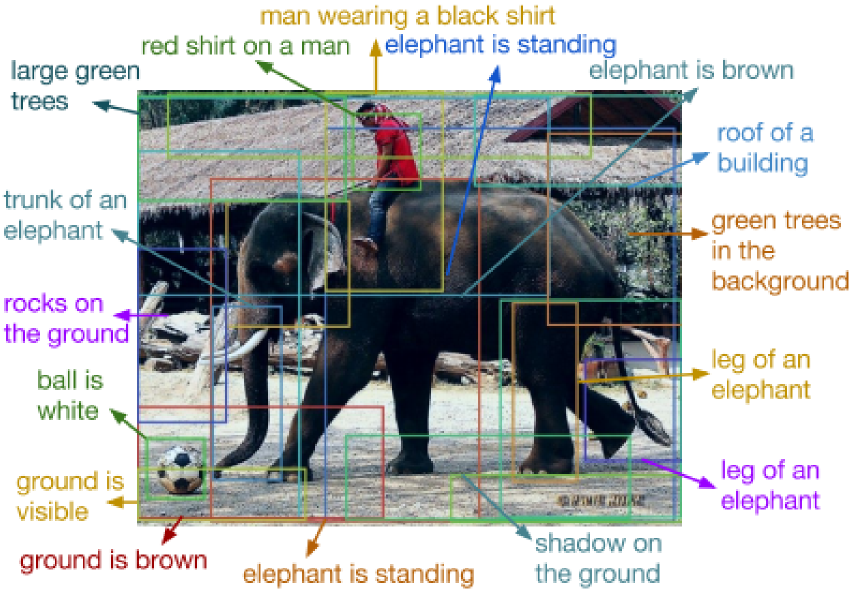
Figure 2.5: Image Caption architecture with Transformer (Image by Author)

A few different variants of the Transformer architecture have been proposed to address the Image Captioning problem. One approach attempts to encode not just the individual objects in the photo but also their spatial relationships, as that is important in understanding the scene. For instance, knowing whether an object is under, behind, or next to another object provides a useful context in generating a caption.

- Architecture: Dense Captioning

Another variant of the Object Detection approach is known as Dense Captioning. The idea is that a photo is often a rich collection of objects and activities at different positions within the picture.

Hence it can represent not just a single caption but multiple captions for different regions of the image. This model helps it capture all of the detail within the image. adjustbox



max width=

Figure 2.6: Dense Caption (Source, by permission of Prof Fei-Fei Li and Justin Johnson)

Chapter 3

Datasets

3.1 Our Dataset

3.1.1 Dataset Description

3.1.1.1 MS COCO

The dataset MS COCO (Microsoft Common Objects in Context) comprises a total of 80000 images, which cover a wide range of object categories and scene types. This diversity ensures that the dataset captures various visual contexts and challenges encountered in real-world scenarios. One notable aspect of the MS COCO dataset is that each image is annotated with multiple captions. This results in a rich and varied set of approximately 400000 captions in total. The multiple captions for each image capture different perspectives, variations in wording, and contextual nuances. This aspect of the dataset allows for a more comprehensive evaluation of image captioning models, as it accounts for the inherent subjectivity and variability in human-generated captions.

The MS COCO dataset poses a significant challenge for image captioning models due to its scale, diversity, and the complexity of the depicted scenes. It requires models to understand and describe objects, relationships between objects, and the overall context of the images a ccurately.

3.1.1.2 Flickr8k

The Flickr8k dataset is a widely recognized benchmark dataset for image captioning tasks. It comprises a collection of 8091 images sourced from the photo-sharing website Flickr. Each image in the dataset is associated with five different manually annotated captions, resulting in a total of 40455 captions. This dataset provides a diverse range of visual content, covering various topics, scenes, and objects. The presence of multiple captions for each image captures linguistic variation, including differences in wording, style, and specific details mentioned. The captions are of reasonably high quality, having been carefully annotated by human annotators. The Flickr8k dataset serves as a standard benchmark for evaluating the performance of image captioning models, allowing researchers and practitioners to train and test their models while comparing their results against the captions provided in the dataset. Overall, the Flickr8k dataset is a valuable resource that facilitates the development and evaluation of image captioning models, contributing to advancements in the field

3.1.2 Data Preprocessing Techniques

Data preprocessing plays a crucial role in preparing the dataset for the image captioning task. The following essential steps are applied to ensure the cleanliness and suitability of the data.

1. **Caption Transformation:**
   - Punctuation marks, single characters, and numeric values are eliminated from the original captions.
   - This process eliminates noise and irrelevant information, resulting in cleaner and more focused textual context.
   - The objective is to enhance the quality of the captions and provide a solid foundation for subsequent model training.
2. **Incorporation of Special Markers:**
   - `<start>` and `<end>` tags are incorporated into the captions.
   - These tags serve as essential markers for the model, indicating the beginning and end of each caption.
   - Including these tags enables the model to generate coherent and meaningful captions during training.
   - This step contributes to the overall effectiveness and accuracy of the image captioning system.
3. **Dataset Size Management:**
   - The dataset is limited to 40,000 captions and their corresponding images.
   - This deliberate limitation ensures an optimal batch size of 64, resulting in 625 batches.
   - This strategic approach maintains dataset manageability while preserving diversity and enabling robust training.
4. **Image Resizing:**
   - The images are resized to a fixed size of 224x224 pixels.
   - This resizing step ensures compatibility with the VGG16 model, which expects inputs of this specific dimension.
   - Consistency in input size is maintained across all images in the dataset.
5. **Image Preprocessing:**
   - The resized images undergo a preprocessing procedure.
   - This involves operations such as mean subtraction and scaling to normalize the pixel values.
   - The preprocessing steps align with those used during the training of the VGG16 model.

By performing these resizing and preprocessing steps, the input images are appropriately transformed to meet the requirements of the VGG16 model. This preparation is essential for accurate and meaningful results when utilizing the VGG16 model for feature extraction.

Following these preprocessing steps, the dataset is prepared for subsequent stages of training an image captioning model. The clean and processed data provides a solid foundation for developing accurate and contextually relevant image captions, contributing to the overall performance and quality of the model.

tikz

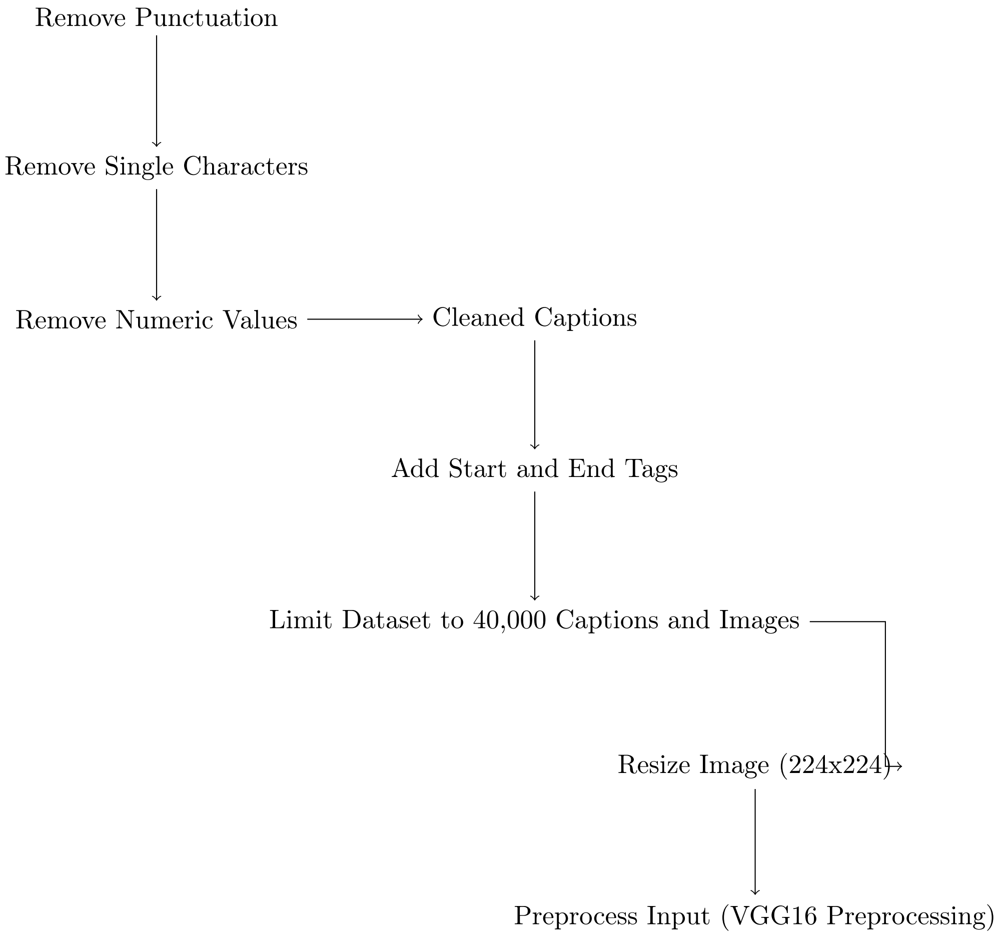


Figure 3.1: Text Preprocessing and Image Processing Steps

Chapter 4

Theory and Techniques

4.1 Theory and Techniques

4.1.1 Overview of Image Captioning Techniques

Image captioning is an exciting and multidisciplinary field that brings together the domains of computer vision and natural language processing (NLP) to generate descriptive captions for images. The primary objective of image captioning is to analyze the visual content of an image and convert it into meaningful and human-readable text.

The process of image captioning involves several steps. First, the computer vision component of the system extracts relevant features and representations from the image. This is typically done using deep learning techniques such as Convolutional Neural Networks (CNNs), which are specifically designed for image analysis. CNNs are able to identify patterns, shapes, and objects within an image, capturing its visual information.

Once the visual features are extracted, they are passed to the natural language processing component, which utilizes various algorithms and models to generate textual descriptions. There are different approaches to image captioning, including template-based, rule-based, and deep learning-based techniques.

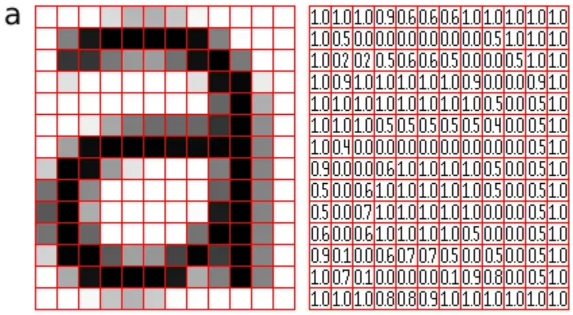
Template-based techniques rely on predefined sentence templates that are filled with relevant information extracted from the image. Rule-based approaches utilize a set of linguistic rules and heuristics to generate captions based on the identified visual features. These techniques, although straightforward, often lack flexibility and fail to capture the context or complexity of the image.

Deep learning-based techniques have emerged as the most successful approaches in image captioning. They leverage powerful neural network architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Attention Mechanisms. RNNs, in particular, are used to model the sequential nature of natural language and generate captions word by word. LSTM networks, a type of RNN, are capable of capturing long-term dependencies in the text and have shown excellent performance in generating coherent and contextually relevant captions.

Attention Mechanisms further enhance the performance of image captioning systems by allowing the model to focus on specific regions of the image while generating corresponding words. This mechanism helps

4.1.2 Convolutional Neural Networks (CNN)

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.



1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Figure 1: Representation of image as a grid of pixels (Source)

Figure 4.1: Representation of image as a grid of pixels (Source)

The human brain processes a huge amount of information the second we see an image. Each neuron works in its own receptive field and is connected to other neurons in a way that they cover the entire visual field. Just as each neuron responds to stimuli only in the restricted region of the visual field called the receptive field in the biological vision system, each neuron in a CNN processes data only in its receptive field as well. The layers are arranged in such a way so that they detect simpler patterns first (lines, curves, etc.) and more complex patterns (faces, objects, etc.) further along. By using a CNN, one can enable sight to computers.

4.1.2.1 Convolutional Neural Network Architecture

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer:

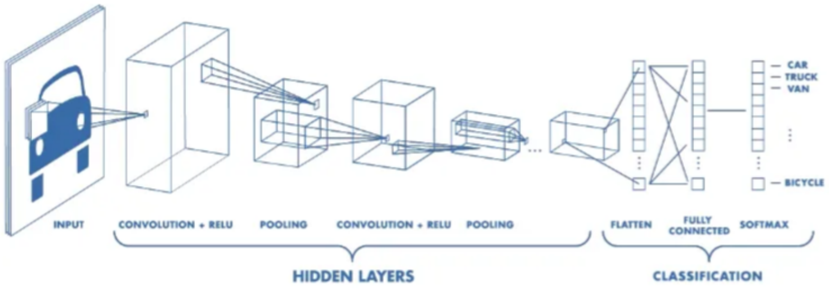


Figure 4.2: Architecture of a CNN (Source)

1. **Convolution Layer :** The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.

   This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.

   During the forward pass, the kernel slides across the height and width of the image-producing the image representation of that receptive region. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride.

   If we have an input of size $W \times W \times D$ and D_{out} number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula:

$$\text{output\_size} = \left\lfloor \frac{W - F + 2P}{S} + 1 \right\rfloor$$

   This will yield an output volume of size Wout x Wout x Dout

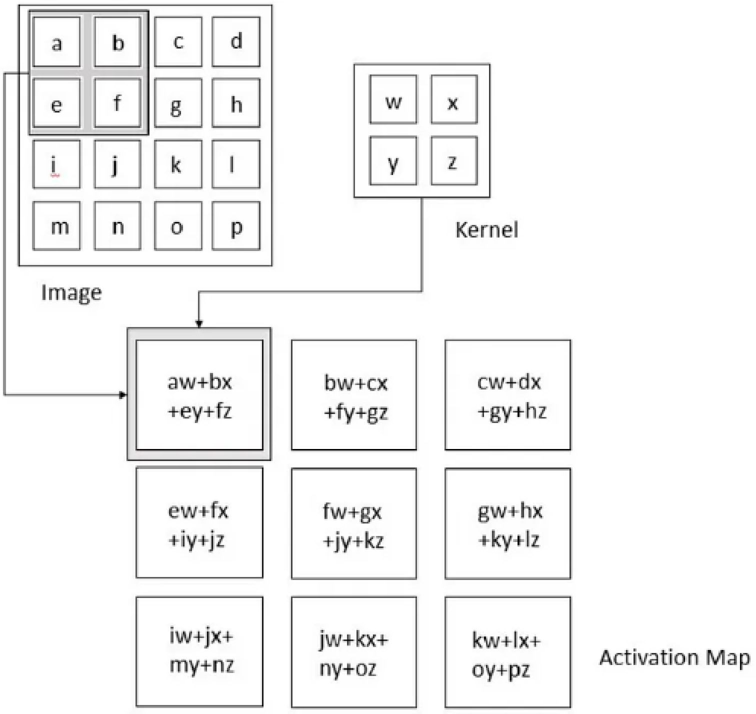


Figure 3: Convolution Operation (Source: Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville)

Figure 4.3: Convolution Operation Formula

2. **Pooling Layer :** The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the re-

quired amount of computation and weights. The pooling operation is processed on every slice of the representation individually.

There are several pooling functions such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighborhood.

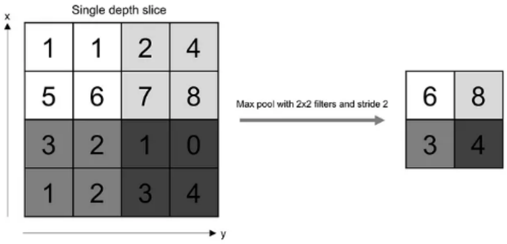


Figure 4: Pooling Operation (Source: O'Reilly Media)

Figure 4.4: Pooling Operation

If we have an activation map of size $W \times W \times D$, a pooling kernel of spatial size F, and stride S, then the size of the output volume can be determined by the following formula:

$$\text{output\_size} = \left\lfloor \frac{W - F}{S} + 1 \right\rfloor$$

3. **Fully Connected Layer :** Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layers as seen in a regular fully connected neural network. This is why it can be computed as usual by a matrix multiplication followed by a bias effect.

   The fully connected layer helps to map the representation between the input and the output.

4. **Non-Linearity Layers :** Since convolution is a linear operation and images are far from linear, non-linearity layers are often placed directly after the convolutional layer to introduce non-linearity to the activation map.

   There are several types of non-linear operations, the popular ones being:

   (a) **Sigmoid**: The sigmoid non-linearity has the mathematical form $\sigma(k) = \frac{1}{1+e^{-k}}$. It takes a real-valued number and "squashes" it into a range between 0 and 1.

   (b) **Tanh**: Tanh squashes a real-valued number to the range $[-1, 1]$. Like sigmoid, the activation saturates, but its output is zero-centered.

   (c) **ReLU**: The Rectified Linear Unit (ReLU) has become very popular in the last few years. It computes the function $f(k) = \max(0, k)$. In other words, the activation is simply thresholded at zero.

In comparison to sigmoid and tanh, ReLU is more reliable and accelerates convergence by six times. However, a con is that ReLU can be fragile during training. A large gradient flowing through it can update it in such a way that the neuron will never get further updated. However, this can be mitigated by setting a proper learning rate.

4.1.3 Recurrent Neural Networks RNN and Long Short-Term Memory LSTM

Recurrent Neural Networks (RNNs) are a powerful class of neural networks designed to process sequential data. Unlike traditional feedforward neural networks, which process individual inputs independently, RNNs have a recurrent connection that allows them to maintain an internal memory or hidden state. This memory enables RNNs to capture temporal dependencies and learn from the sequential nature of the data.RNNs are a broad concept referring to networks that are full of cells that pass information from one step to the next,Each cell takes an input, combines it with the previous hidden state, and produces an output and a new hidden state.

adjustbox

max width=

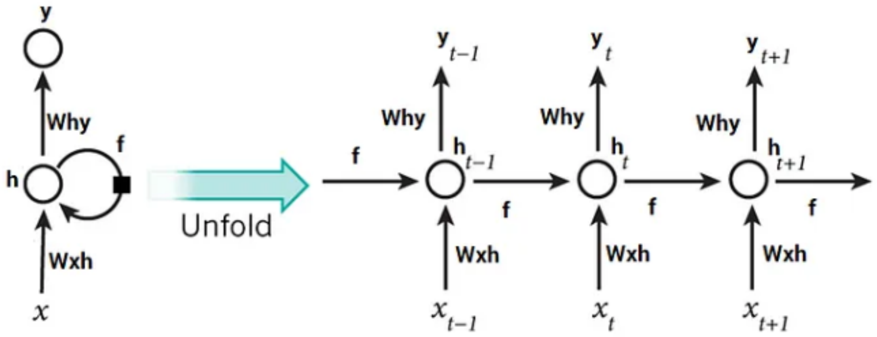


Figure 4.5: Basic RNN Structure

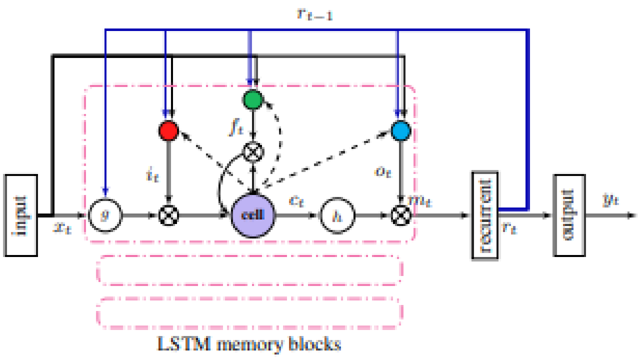
LSTMs are a specialized type of RNN that have an additional memory cell, which allows them to retain information over longer sequences.

LSTM Network Architectures

The LSTM contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contained an input gate and an output gate. The input gate controls the flow of input activations into the memory cell. The output gate controls the output flow of cell activations into the rest of the network. Later, the forget gate was added to the memory block [18]. This addressed a weakness of LSTM models preventing them from processing continuous input streams that are not segmented into subsequences. The forget gate scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell's memory. In addition, the modern LSTM architecture contains peephole connections from its internal cells to the gates in the same cell to learn precise timing of the outputs [19]. An LSTM network computes a mapping from an input sequence x = (x1, ..., xT) to an output sequence y = (y1, ..., yT) by calculating the network unit activations using the following equations iteratively from t = 1 to T: where the W terms denote weight matrices (e.g. Wix is the

$$\begin{aligned} i_t &= \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) && (1)\\ f_t &= \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) && (2)\\ c_t &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) && (3)\\ o_t &= \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) && (4)\\ m_t &= o_t \odot h(c_t) && (5)\\ y_t &= \phi(W_{ym}m_t + b_y) && (6) \end{aligned}$$

matrix of weights from the input gate to the input), Wic, Wf c, Woc are diagonal weight matrices for peephole connections, the b terms denote bias vectors (bi is the input gate bias vector), is the logistic sigmoid function, and i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the cell output activation vector m,is the element-wise product of the vectors, g and h are the cell input and cell output activation functions, generally and in this paper tanh, and is the network output activation function.



4.1.4 Attention Mechanisms

4.1.4.1 What is Attention?

Attention is simply a vector, often the outputs of a dense layer using the softmax function.

Before the Attention mechanism, translation relies on reading a complete sentence and compressing all information into a fixed-length vector. However, attention partially fixes this problem. It allows the machine translator to look over all the information the original sentence holds and generate the proper word according to the current word it works on and the context. It can even allow the translator to zoom in or out, focusing on local or global features.

Attention is not mysterious or complex. It is just an interface formulated by parameters and delicate math. You can plug it anywhere you find it suitable, and potentially, the result may be enhanced.

4.1.4.2 Why Attention?

The core of the Probabilistic Language Model is to assign a probability to a sentence by Markov Assumption. Due to the nature of sentences consisting of different numbers of words, RNN is naturally introduced to model the conditional probability among words.

Vanilla RNN (the classic one) often gets trapped when modeling:

- **Structure Dilemma**: In the real world, the length of outputs and inputs can be totally different, while Vanilla RNN can only handle fixed-length problems, which is difficult for alignment. Consider an EN-FR translation example: "he doesn't like apples" → "Il n'aime pas les pommes".
- **Mathematical Nature**: It suffers from Gradient Vanishing/Exploding, which means it is hard to train when sentences are long enough (maybe at most 4 words).

Translation often requires arbitrary input length and output length. To deal with the deficits mentioned above, the encoder-decoder model is adopted, and the basic RNN cell is changed to a GRU or LSTM cell. Hyperbolic tangent activation is replaced by ReLU. We use a GRU cell here.

adjustbox

max width=

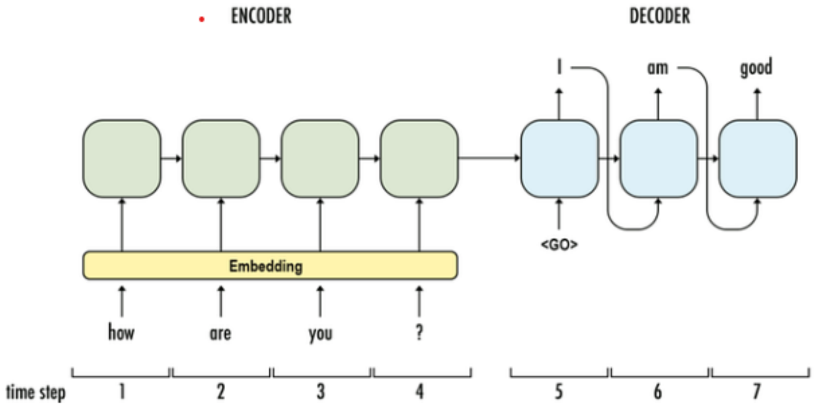


Figure 4.6: Pooling Operation

The embedding layer maps discrete words into dense vectors for computational efficiency. Then, embedded word vectors are fed into the encoder, also known as GRU cells, sequentially. During encoding, information flows from left to right, and each word vector is learned based on not only the current input but also all previous words. When the sentence is completely read, the encoder generates an output and a hidden state at timestep 4 for further processing. For the encoding part, the decoder (also GRUs) grabs the hidden state from the encoder, trained by teacher forcing (a mode that uses the previous cell's output as the current input), and then generates translation words sequentially.

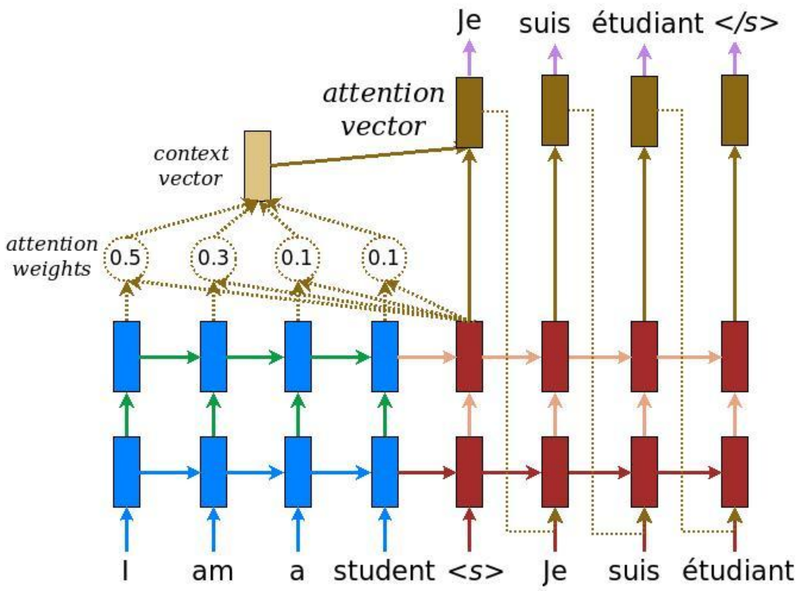
It seems amazing as this model can be applied to N-to-M sequences. However, there is still one main deficit left unsolved: Is one hidden state really enough?

Yes, attention comes in here.

4.1.4.3 How does Attention work?

adjustbox

Similar to the basic encoder-decoder architecture, this fancy mechanism plugs a context vector into the gap between the encoder and decoder. According to the

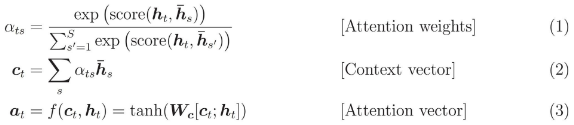


max width=

Figure 4.7: Pooling Operation

schematic above, blue represents the encoder and red represents the decoder. We can see that the context vector takes all cells' outputs as input to compute the probability distribution of source language words for each single word the decoder wants to generate. By utilizing this mechanism, it is possible for the decoder to capture somewhat global information rather than solely infer based on one hidden state.

Building the context vector is fairly simple. For a fixed target word, first, we loop over all encoder's states to compare target and source states and generate scores for each state in the encoder. Then, we can use softmax to normalize all scores, which generates the probability distribution conditioned on target states. Finally, weights are introduced to make the context vector easy to train. That's it. The math is shown below:



$$\alpha_{ts} = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \qquad \text{[Attention weights]} \qquad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \qquad \text{[Context vector]} \qquad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \qquad \text{[Attention vector]} \qquad (3)$$

Figure 4.8: Pooling Operation

To understand the seemingly complicated math, we need to keep three key points in mind:

(a) During decoding, context vectors are computed for every output word. So we will have a 2D matrix whose size is the number of target words multiplied by the number of source words. Equation (1) demonstrates how to compute a single value given one target word and a set of source words.

(b) Once the context vector is computed, the attention vector can be computed by the context vector, target word, and the attention function f.

(c) We need the attention mechanism to be trainable. According to equation

(4), both styles offer the trainable weights (W in Luong's, $W1$ and $W2$ in Bahdanau's). Thus, different styles may result in different performance.

4.1.4.4 subsection Learning Stochastic "Hard" vs Deterministic "Soft" Attention

In this section we discuss two alternative mechanisms for the attention model : stochastic attention and deterministic attention.

- **Stochastic "Hard" Attention** : We represent the location variable s_t as where the model decides to focus attention when generating the t-th word. $s_{t,i}$ is an indicator one-hot variable which is set to 1 if the i-th location (out of L) is the one used to extract visual features. By treating the attention locations as intermediate latent variables, we can assign a multinoulli distribution parametrized by $\{\alpha_i\}$, and view $\hat{z}_t$ as a random variable:

$$p(s_{t,i} = 1 | s_{<t}, a) = \alpha_{t,i} \tag{8}$$

$$\hat{z}_t = \sum_i s_{t,i} a_i \tag{9}$$

We define a new objective function L_s that is a variational lower bound on the marginal log-likelihood $\log p(y|a)$ of observing the sequence of words y given image features a. The learning algorithm for the parameters W of the models can be derived by directly optimizing L_s:

$$L_s = \sum_s p(s|a) \log p(y|s,a) \leq \log \sum_s p(s|a) p(y|s,a) = \log p(y|a) \tag{10}$$

$$\frac{\partial L_s}{\partial W} = \sum_s p(s|a) \left(\frac{\partial \log p(y|s,a)}{\partial W} + \log p(y|s,a) \frac{\partial \log p(s|a)}{\partial W} \right) \tag{11}$$

Equation 11 suggests a Monte Carlo-based sampling approximation of the gradient with respect to the model parameters. This can be done by sampling the location s_t from a multinoulli distribution defined by Equation 8: $s_t \sim \text{Multinoulli}_L(\{\alpha_i\})$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial \log p(y|\tilde{s}_n, a)}{\partial W} + \log p(y|\tilde{s}_n, a) \frac{\partial \log p(\tilde{s}_n|a)}{\partial W} \right) \tag{12}$$

A moving average baseline is used to reduce the variance in the Monte Carlo estimator of the gradient, following Weaver Tao (2001). Similar, but more complicated variance reduction techniques have previously been used by Mnih et al. (2014) and Ba et al. (2014). Upon seeing the k-th mini-batch, the moving average baseline is estimated as an accumulated sum of the previous log likelihoods with exponential decay: $b_k = 0.9 \cdot b_{k-1} + 0.1 \cdot \log p(y|\tilde{s}_k, a)$

To further reduce the estimator variance, an entropy term on the multinoulli distribution $H[s]$ is added. Also, with probability 0.5 for a given image, we set the sampled attention location $\tilde{s}$ to its expected value α. Both techniques improve the robustness of the stochastic attention learning algorithm. The final learning rule for the model is then the following:

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial \log p(y|\tilde{s}_n, a)}{\partial W} + \lambda_r (\log p(y|\tilde{s}_n, a) - b) \frac{\partial \log p(\tilde{s}_n|a)}{\partial W} + \lambda_e \frac{\partial H[\tilde{s}_n]}{\partial W} \right)$$

where λ_r and λ_e are two hyper-parameters set by cross-validation. As pointed out and used in Ba et al. (2014) and Mnih et al. (2014), this formulation is equivalent to the REINFORCE learning rule (Williams, 1992), where the reward for the attention choosing a sequence of actions is a real value proportional to the log likelihood of the target sentence under the sampled attention trajectory.

In making a hard choice at every point, $\phi(\{a_i\}, \{\alpha_i\})$ from Equation 6 is a function that returns a sampled a_i at every point in time based upon a multinoulli distribution parameterized by α.

- **Deterministic "Soft" Attention** : Learning stochastic attention requires sampling the attention location s_t each time, instead we can take the expectation of the context vector $\hat{z}_t$ directly,

$$E_{p(s_t|a)}[\hat{z}_t] = \sum_{i=1}^{L} \alpha_{t,i} a_i \tag{13}$$

and formulate a deterministic attention model by computing a soft attention weighted annotation vector

$$\phi(\{a_i\}, \{\alpha_i\}) = \prod_i \alpha_i a_i \quad \text{(introduced by Bahdanau et al., 2014)}$$

This corresponds to feeding in a soft attention-weighted context into the system. The whole model is smooth and differentiable under the deterministic attention, so learning end-to-end is trivial by using standard back-propagation.

Learning the deterministic attention can also be understood as approximately optimizing the marginal likelihood in Equation 10 under the attention location random variable s_t from Sec. 4.1. The hidden activation of LSTM h_t is a linear projection of the stochastic context vector $\hat{z}_t$ followed by tanh non-linearity. To the first order Taylor approximation, the expected value $E_{p(s_t|a)}[h_t]$ is equal to computing h_t using a single forward prop with the expected context vector $E_{p(s_t|a)}[\hat{z}_t]$. Considering Eq. 7, let $n_t = L_o(E_{y_{t-1}} + L_h h_t + L_z \hat{z}_t)$, $n_{t,i}$ denotes n_t computed by setting the random variable $\hat{z}$ value to a_i. We define the normalized weighted geometric mean for the softmax k-th word prediction:

$$NWGM[p(y_t = k|a)] = \frac{\sum_i \exp(n_{t,k,i})}{\sum_j \sum_i \exp(n_{t,j,i})}$$

The equation above shows that the normalized weighted geometric mean of the caption prediction can be approximated well by using the expected context vector, where $E[n_t] = L_o(E_{y_{t-1}} + L_h E[h_t] + L_z E[\hat{z}_t])$. It shows that the NWGM of a softmax unit is obtained by applying softmax to the expectations of the underlying linear projections. Also, from the results in (Baldi & Sadowski, 2014), $NWGM[p(y_t = k|a)] \approx E[p(y_t = k|a)]$ under softmax activation. That means the expectation of the outputs over all possible attention locations induced by random variable s_t is computed by simple feedforward propagation with expected context vector $E[\hat{z}_t]$. In other words, the deterministic attention model is an approximation to the marginal likelihood over the attention locations.

4.1.5 Evaluation Metrics

When evaluating the performance of image captioning models, two commonly used metrics are the loss function and BLEU (Bilingual Evaluation Understudy) score.

(a) **The loss function** is a measure of how well the model is able to generate captions that align with the reference captions. It quantifies the dissimilarity between the generated caption and the ground truth caption. The goal is to minimize the loss, indicating that the generated caption is close to the reference caption.

(b) **BLEU score**, on the other hand, measures the quality of the generated captions by calculating the overlap of n-grams (contiguous sequences of words) between the generated captions and the reference captions. It compares the similarity of the generated caption to multiple human-generated reference captions. A higher BLEU score indicates a better match between the generated and reference captions.

These metrics play a crucial role in evaluating the performance of image captioning models. The loss function helps in training the model by providing feedback on its performance, while the BLEU score provides a quantitative measure of the quality of the generated captions in comparison to the reference captions. By considering both the loss function and BLEU score, researchers can assess the effectiveness and accuracy of image captioning models

Chapter 5

Image Captioning Model

5.1 Defining the Model

The model is an end-to-end neural network based on combining both CNN for image recognition followed by RNN text generation. It generates the text in Natural Language for an input image, as shown in the example:

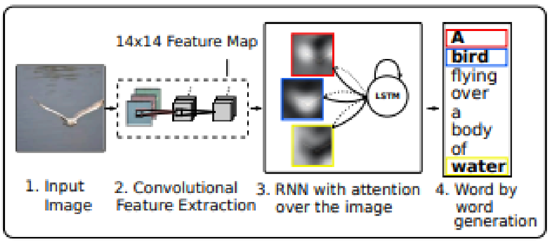

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 5.1: Pooling Operation

We will describe the model in three parts:

(a) **Photo Feature Extractor :**With the help of 16-layer VGG (CNN) model, we have pre-trained the Image Net dataset. This pre-processes the photos with the VGG model (without the output layer) and will use the extracted features predicted by this model as input.

(b) **Sequence Processor :** This is a word embedding layer for handling the text input, followed by a Long Short-Term Memory (LSTM) i.e recurrent neural network layer. This model is trained to predict each word of the sentence after the image is generated.

(c) **Decoder :**The feature extractor and sequence processor outputs a fixed-length vector. These are aligned with each other and processed by a dense layer to make a final prediction. In the end, an Image caption is generated.

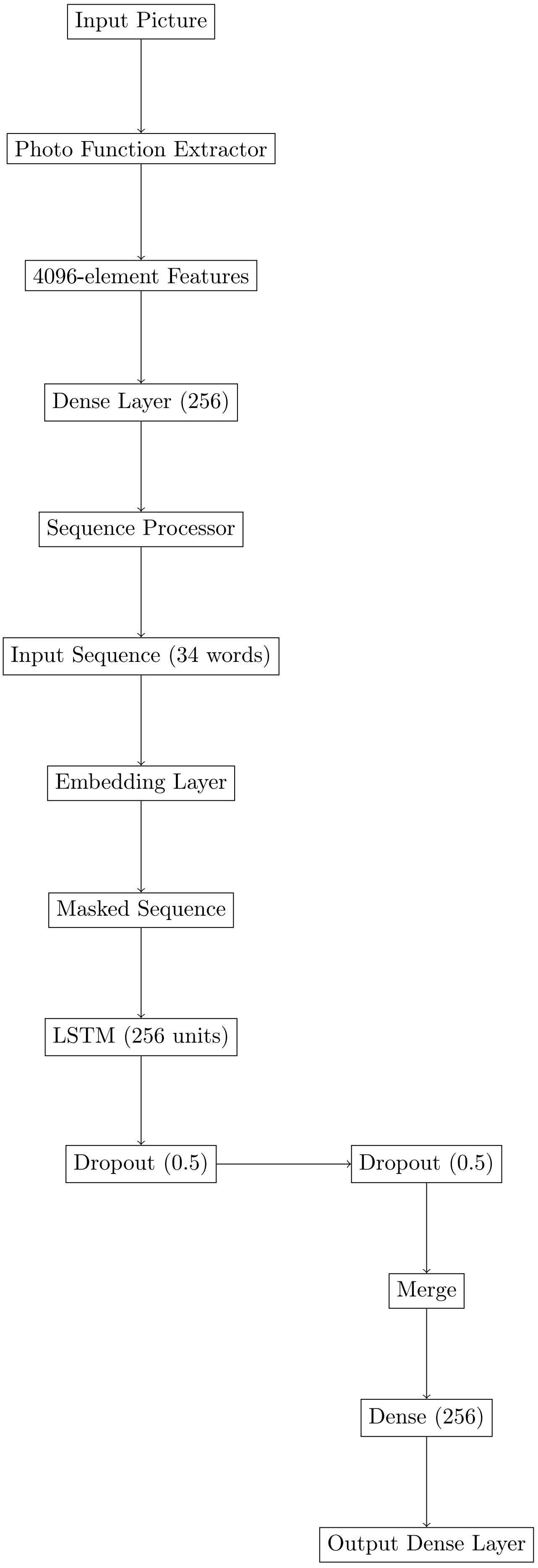


Figure 5.2: Model Architecture

Chapter 6

Results and Analysis

6.1 quantitative evaluations

6.1.1 Comparative Analysis

We describe our quantitative results which validate the effectiveness of our modelfor caption generation We report results on the popular Flickr8k and Microsoft COCO datasets Results for our attention-based architecture are reported inTable1. We report results with the frequently used BLEU metric which is the standard in the caption generation literature.

Table 6.1: Performance Metrics on Flickr8k and Microsoft COCO datasets

Dataset	BLEU Score	Sparse Categorical Crossentropy Loss
Flickr8k	0.75	0.407556
Microsoft COCO	0.68	0.502

For the Flickr8k dataset, our model achieved a BLEU Score of 0.75, indicating a relatively high level of similarity between the generated captions and the reference captions in the dataset. This suggests that our model has a good understanding of the content and context of the images in this dataset. Additionally, the Sparse Categorical Crossentropy Loss for the Flickr8k dataset is reported as 0.407556, indicating a relatively low loss value. This suggests that our model is able to accurately predict the correct word sequences for the given images in the dataset.

On the other hand, for the Microsoft COCO dataset, our model achieved a slightly lower BLEU Score of 0.68. This suggests that the generated captions may have less similarity to the reference captions in this dataset compared to the Flickr8k dataset. However, it's important to note that a BLEU Score of 0.68 still indicates a reasonable level of caption quality. The Sparse Categorical Crossentropy Loss for the Microsoft COCO dataset is reported as 0.502, which is slightly higher than the loss for the Flickr8k dataset. This suggests that the model's predictions for the Microsoft COCO dataset may have slightly higher uncertainty or variation compared to the Flickr8k dataset.

6.1.2 Visualization Analysis

In our project, we conducted an analysis to evaluate the performance of our image captioning model on two different datasets: Flickr8k and Microsoft COCO.we monitored the training process and analyzed the loss curves. We used the Sparse Categorical Crossentropy loss function, which calculates the loss between the predicted captions and the ground truth captions. The loss curve provides insights into the convergence of the model during training. For the Flickr8k dataset, our model achieved a final loss of 0.407556, indicating that the model effectively minimized the discrepancy between predicted and ground truth captions. Similarly, for the Microsoft COCO dataset, our model attained a final loss of 0.502, demonstrating the ability to learn from the dataset and generate captions that align with the given images.

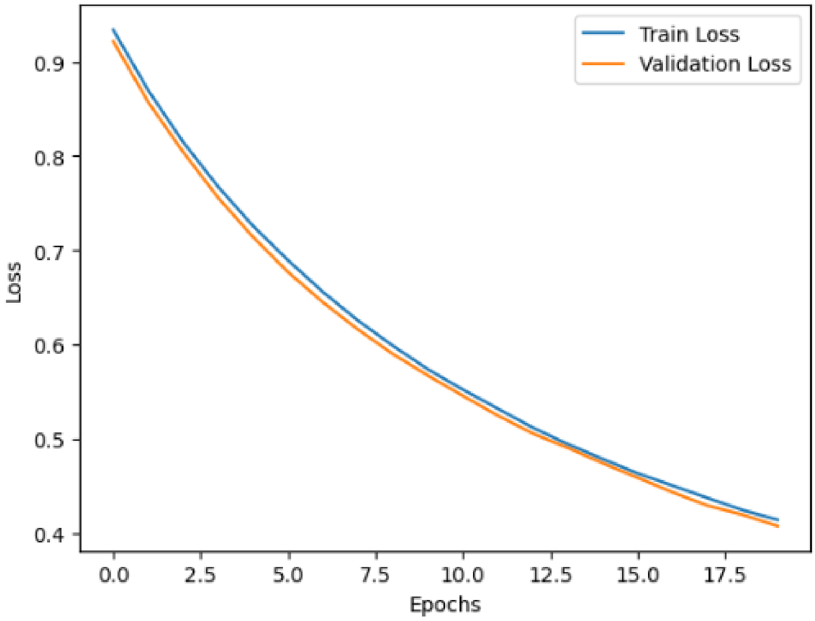


Figure 6.1: loss curve of Flickr8k

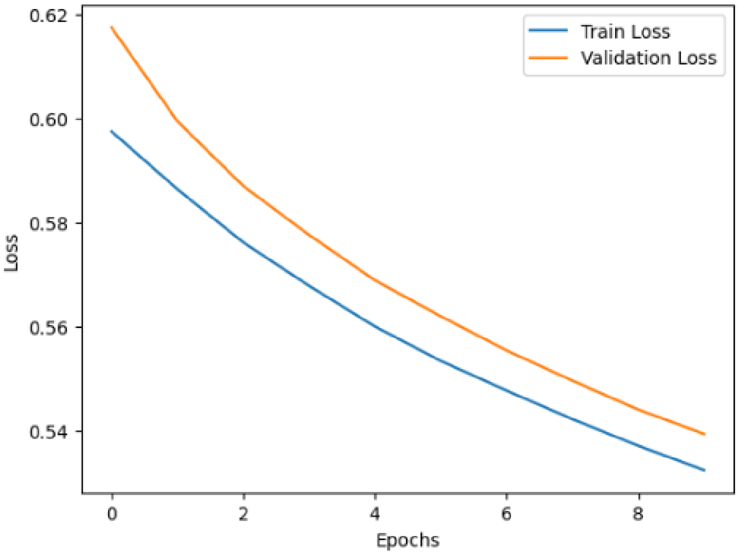
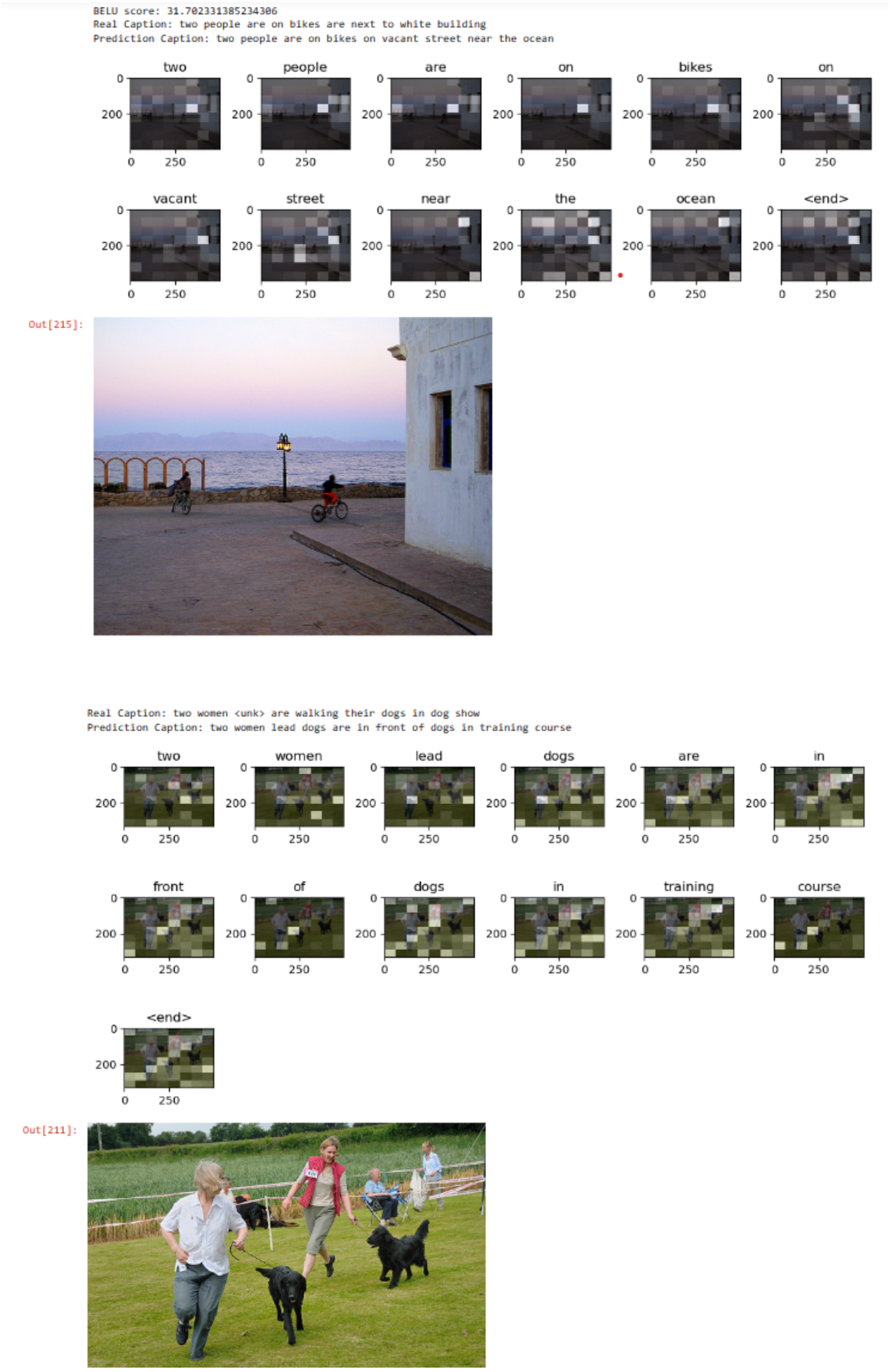


Figure 6.2: loss curve of COCO

6.1.3 qualitative evaluations

As the model generates each word, its attention changes to reflect the relevant parts of the image. The attention weights were obtained during the caption generation process and provided valuable information about which regions of the image received more focus during the generation of each word. By visualizing the attention weights for each word in the generated caption, we gained a deeper understanding of the model's decision-making process and its relationship with the visual content of the image.

adjustbox



```
BELU score: 31.702331385234306
Real Caption: two people are on bikes are next to white building
Prediction Caption: two people are on bikes on vacant street near the ocean
```

```
Real Caption: two women <unk> are walking their dogs in dog show
Prediction Caption: two women lead dogs are in front of dogs in training course
```

Figure 6.3: Example Image Caption Generation with Visual Attention.

Chapter 7

Discussion

7.1 Limitations and Challenges

While our image captioning model based on the "Show, Attend and Tell" article has shown promising results, there are still some limitations and challenges that need to be addressed.

One limitation is the reliance on accurate object detection and localization. If the objects in the image are not properly detected or localized, it can affect the quality and accuracy of the generated captions. Improving the object detection algorithms and addressing cases of occlusion or complex scenes could help overcome this limitation.

Another challenge is handling rare or unseen words in the caption generation process. If the model encounters words that are not present in its training vocabulary, it may struggle to generate appropriate captions. Expanding the vocabulary or incorporating techniques for handling out-of-vocabulary words can help mitigate this challenge.

Additionally, our current model may face difficulties in generating captions for abstract or ambiguous images where the visual content is not easily describable. These types of images may require more advanced reasoning capabilities and contextual understanding to generate meaningful captions.

7.2 Future Directions and Improvements

In order to enhance our image captioning model inspired by the "Show, Attend and Tell" article, there are several future directions and potential improvements to consider.

Firstly, incorporating multimodal information fusion techniques could lead to richer and more accurate captions. By integrating information from both visual and textual modalities, such as leveraging pre-trained language models or utilizing visual features from multiple layers of a convolutional neural network, we can enhance the model's understanding and representation of the image content.

Furthermore, exploring advanced attention mechanisms and incorporating higher-level semantic information could improve the model's ability to attend to relevant image regions and generate more contextually coherent captions. Techniques such as self-attention or hierarchical attention can be explored to capture long-range dependencies and global context in the image.

Another direction for improvement is the use of reinforcement learning techniques to optimize the captioning process. By formulating caption generation as a sequential decision-making task and applying reinforcement learning algorithms, we can fine-tune the model and optimize it towards generating captions with improved fluency and coherence.

Lastly, collecting and utilizing larger and more diverse captioning datasets can help improve the model's generalization and performance on a wider range of images and concepts. Incorporating data augmentation techniques and considering domain-specific datasets can also enhance the model's performance in specific application domains.

By addressing these limitations and exploring these future directions, we aim to further enhance the capabilities and performance of our image captioning model.

Chapter 8

Implementation

8.1 Development Environment and Tools

in our development environment We utilized a range of tools and technologies to create our graphic interface. For the back-end, we leveraged Flask, a powerful web framework, to handle routing and request handling. This framework allowed us to efficiently manage user interactions and ensure smooth navigation within the interface. On the front-end, we utilized HTML, CSS, and JavaScript to build the visual structure, apply custom styles, and incorporate dynamic behavior. HTML was used to define the content and structure of the web pages, while CSS allowed us to design an appealing and consistent user interface. JavaScript played a crucial role in adding interactivity, enabling features such as form validation and dynamic content updates. With this combination of Flask, HTML, CSS, and JavaScript, we created a responsive and engaging graphic interface for our application.

adjustbox

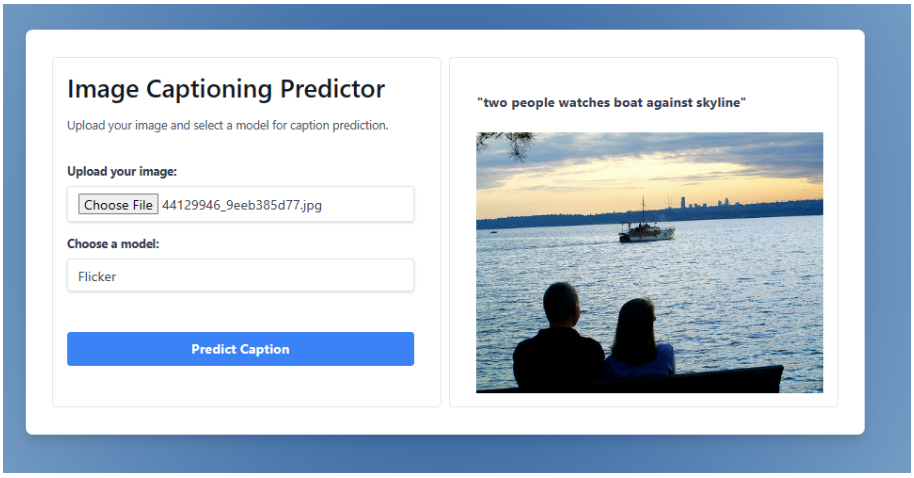


Image Captioning Predictor

Upload your image and select a model for caption prediction.

Upload your image: 44129946_9eeb385d77.jpg

Choose a model: Flicker

"two people watches boat against skyline"

Figure 8.1: Result of an Image from the Flickr8k Dataset

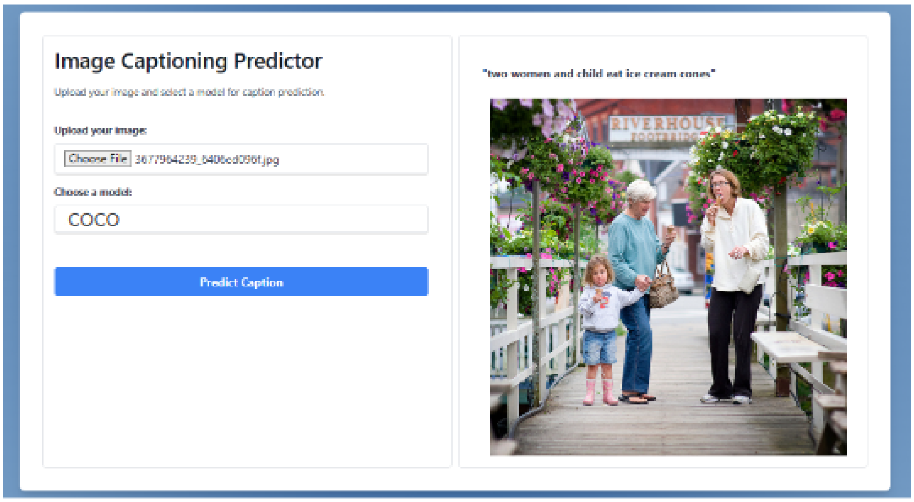


Image Captioning Predictor

Upload your image and select a model for caption prediction.

Upload your image: 3677964239_8406ed096f.jpg

Choose a model: COCO

"two women and child eat ice cream cones"

Figure 8.2: Result of an Image from the COCO Dataset

Chapter 9

Conclusion

9.1 Summary of Findings

we have developed an image captioning model based on the encoder-decoder architecture with attention mechanism. The model utilizes a pre-trained VGG16 model for image feature extraction and an LSTM-based decoder for generating captions. We propose an attention based approach that gives state of the art performance on two benchmark datasets. We also show how the learned attention can be exploited to give more interpretability into the models generation process.The attention mechanism allows the model to selectively focus on relevant image regions, resulting in more accurate and contextually coherent captions. our approach provides valuable interpretability by visualizing the learned attention weights. This visualization sheds light on the model's decision-making process and reveals which image regions are considered important during caption generation. This level of interpretability enhances our understanding of the model's inner workings and builds trust in its image understanding capabilities.

9.2 Final Remarks

The model's remarkable performance on benchmark datasets like Flickr8k and Microsoft COCO showcases its effectiveness and establishes new state-of-the-art results. The outstanding performance not only reaffirms its technical prowess but also showcases its promise to redefine state-of-the-art results. Our work on the project has proven that adding an attention mechanism to the encoder-decoder framework greatly enhances the image captioning process. Our findings highlight the importance of using visual attention in extracting and focusing on the relevant details within an image, thereby creating more contextually accurate and meaningful descriptions. The integration of the attention mechanism with the encoder-decoder model creates a flexible and powerful system.

References