MINOR PROJECT

on

Image Captioning Using VGG16 with Bahdanau Attention and LSTM: An Advanced Approach for Generating Descriptive Text

[ARP 455]

Name: **ANIRUDH SHUKLA**

Enrollment Number: **00619011921**

Under the supervision of

MS. HIMANI TYAGI

**UNIVERSITY SCHOOL OF AUTOMATION AND
ROBOTICS GURU GOBIND SINGH INDRAPRASTHA
UNIVERSITY EAST DELHI CAMPUS, SURAJMAL VIHAR,
DELHI- 110032**

INDEX

LIST OF TABLES

LIST OF FIGURES

1. ABSTRACT

This project focuses on developing an advanced **Image captioning system** that leverages a combination of **Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs),** and the **Bahdanau Attention mechanism.** The system is designed to automatically generate accurate and context-aware textual descriptions for images, using deep learning techniques to understand and correlate visual content with language.

At the core of this project is the use of the **VGG16 model**, a widely used convolutional neural network architecture, for feature extraction from images. VGG16, pre-trained on the ImageNet dataset, provides a robust mechanism to extract high-level visual features that encapsulate important details about the objects, scenes, and textures in the image. These features form the foundation for generating meaningful captions. CNNs like VGG16 have been instrumental in solving various computer vision tasks, including image classification, object detection, and, in this case, image captioning.

The project also integrates a **Recurrent Neural Network (RNN**), specifically a **Long Short-Term Memory (LSTM) [2]** network, which is highly effective in processing sequential data such as natural language. LSTMs are designed to remember long-term dependencies, making them suitable for tasks like caption generation, where the context and order of words matter significantly. The RNN component of this system takes the image features extracted by the CNN and uses them to generate captions word by word.

A key innovation in this project is the use of the **Bahdanau Attention mechanism**. Attention mechanisms have transformed natural language processing and machine learning by allowing models to focus on specific parts of the input data when making predictions. In the context of image captioning, Bahdanau Attention enables the model to focus on different regions of the image when generating each word in the caption, rather than treating the entire image as a single entity. This allows for more fine-grained and contextually relevant captions, as the model learns to attend to the most important image features at each step of the sentence generation process.

The system is trained on the **Flickr 8k dataset**, which consists of 8,000 images, each paired with five human-annotated captions. Preprocessing steps include resizing the images to fit the VGG16 model's input requirements and tokenizing the captions to transform them into a

sequence of integers that the model can understand. Once trained, the model is able to take new images, extract features, and generate appropriate captions, producing natural-sounding and accurate descriptions of the visual content.

Results of this project are evaluated using **BLEU scores**, which are a standard metric for assessing the quality of machine-generated text by comparing it to human-generated reference captions. The inclusion of attention leads to captions that are more descriptive and better aligned with the image context, outperforming baseline models that do not use attention mechanisms.

The applications of this system are broad and impactful, ranging from **Automatic Image captioning in search engines** to **assistive technologies** for individuals with visual impairments. It could also be used in fields such as **robotic vision systems**, where understanding and describing the environment is crucial.

In the future, incorporating more advanced architectures, such as **transformer-based models**, or training on larger and more diverse datasets, could further enhance the system's performance, enabling it to handle more complex images and provide richer, more detailed captions.

2. INTRODUCTION

Image captioning is an advanced field of artificial intelligence that focuses on generating descriptive textual captions based on the content of images. By bridging computer vision and natural language processing, this task holds tremendous potential across a wide range of applications, from assisting visually impaired individuals to improving the organization and retrieval of multimedia content. The process of image captioning relies on deep learning techniques that enable machines to learn from large datasets of images paired with human-generated captions.

In this project, we utilize a combination of two major techniques: Convolutional Neural Networks (CNNs), specifically the VGG16 model, and Recurrent Neural Networks (RNNs) enhanced by the Bahdanau Attention mechanism. This hybrid approach leverages the strengths of both models to effectively extract features from images and generate coherent textual descriptions. The ultimate aim of this project is to create a system capable of generating accurate, human-like captions for images using advanced deep learning methods.

2.1 ROLE OF VGG16

The VGG16 model plays a pivotal role in this image captioning system as the backbone for feature extraction. VGG16 is a convolutional neural network known for its simplicity and powerful feature extraction capabilities. Developed by the Visual Geometry Group at Oxford, the model consists of 16 layers, which include 13 convolutional layers followed by fully connected layers. VGG16 is pre-trained on the ImageNet dataset, which consists of millions of labeled images, making it highly effective at capturing a wide range of visual features.

In the context of image captioning, VGG16 processes an image by analyzing its spatial features. It breaks down the image into smaller components such as edges, shapes, and textures. These features are then passed through successive layers of the network, each time extracting increasingly abstract representations of the visual content. The output from the final layer of VGG16 serves as a rich feature vector that encapsulates the most critical elements of the image. These features are used as input for the subsequent caption generation process.

2.2 ROLE OF ATTENTION MECHANISM

The Bahdanau Attention mechanism, also known as soft attention, significantly improves the quality of captions by allowing the model to focus on different parts of the image during the caption generation process. Traditional CNN-RNN [3] models treat the entire image as a single entity, which can lead to less specific and inaccurate captions. However, the attention mechanism enables the model to dynamically adjust its focus on different regions of the image when generating each word in the caption.

By doing so, the attention mechanism ensures that the model attends to relevant image features at the appropriate time, thereby improving the contextual accuracy of the generated caption. This leads to more nuanced and detailed descriptions, as the model learns to highlight the most important aspects of the image while generating the next word in the caption sequence. This capability is crucial, especially when dealing with complex images containing multiple objects or intricate scenes.

2.3 MOTIVATION

The motivation behind this project is driven by the growing need for systems that can understand and interpret visual data in a way that is accessible to humans. With the vast amount of visual content being generated every day, there is a critical need for technologies that can automatically tag, describe, and organize images. Current systems often require manual labeling, which is time-consuming and prone to errors. By automating the captioning process, this project seeks to reduce the burden of manual labeling and make large collections of images easier to navigate.

Another key motivation is the potential social impact. Image captioning systems can play a crucial role in assistive technologies for individuals with visual impairments, enabling them to better understand their surroundings. These systems can also be integrated into search engines, making it easier to retrieve images based on their visual content.

2.4 OBJECTIVES

The primary objective of this project is to design and implement a robust image captioning system that generates accurate and context-aware captions for a given image. The system aims to combine CNNs for image feature extraction and RNNs, specifically LSTM networks, with attention mechanisms for natural language generation. The focus is on creating captions that not only describe the image but also capture important details and relationships between objects in the image.

2.5 SCOPE AND SIGNIFICANCE

The scope of this project extends to multiple areas of application, including content management systems, assistive technologies, and automated multimedia generation. The significance of the project lies in its ability to reduce the need for human intervention in captioning large datasets, making it highly relevant for businesses and organizations dealing with image-heavy data. Additionally, the use of advanced attention mechanisms positions this project at the forefront of image captioning research, contributing to ongoing efforts to improve the efficiency and accuracy of these systems.

In conclusion, this project aims to leverage state-of-the-art deep learning techniques to enhance the quality and precision of automatically generated image captions, thereby pushing the boundaries of what machines can achieve in terms of visual understanding and language generation.

3. LITERATURE SURVEY

S. No.	Dataset	Year	Model	Attention Mechanism	Remarks
1.	ILSVRC-2012	2015	ConvNet [1]	✕	No focus on non-visual objects
2.	ImageNet	2015	ResNet 101 [2]	✕	No focus on attention mechanism
3.	Pascal VOC 2008, Flickr8k, Flickr30k, MSCOCO, SBU	2015	CNN-LSTM [3]	✕	No focus on attention mechanism
4.	Pascal VOC 2012, Pascal Context, Pascal Part, Cityscape	2017	ResNet 101 [2]	✕	Proposed model failed to capture boundaries
5.	COCO, Flickr30k, Flickr8k	2016	Bi-LSTM [4]	✕	No focus on attention mechanism
6.	COCO, Flickr30k, Flickr8k	2017	CNN-RNN [1]	✓	Proposed model requires more investigation to avoid overfitting
7.	COCO dataset	2018	CNN[5]	✓	Framework is not end-to-end

8.	COCO dataset, Flickr30k, Flickr8k	2018	FCN-LSTM [7]	✓	Did not evaluate different pre-trained models
9.	MSCOCO, COCO-stuff	2018	R-CNN + ResNet 101-LSTM [16]	✓	Proposed model is computationally costly
10.	MSCOCO, Visual Genome dataset, VQA v2.0 dataset	2018	Faster R-CNN-LSTM [17]	✓	Evaluated their model only on one dataset
11.	COCO dataset	2019	Graph CNN-LSTM	✕	Did not incorporate attention mechanism
12.	MS-COCO, Flickr30k	2019	Reference LSTM	✓	Attention on attention may lead to loss of significant information
13.	MS-COCO	2019	RCNN-LSTM	✓	Training GAN is computationally very costly
14.	Flickr30k, Flickr8k	2019	CNN-LSTM	✓	Did not evaluate different pre-trained models
15.	MS-COCO	2022	CNN-GRU	✓	Evaluated the model only on one dataset

Table 3.1 Summarized Details of Litertature Survey.

4. PROBLEM STATEMENT

In recent years, the intersection of artificial intelligence and computer vision has brought significant advancements in understanding and interpreting visual content. One of the prominent challenges in this domain is image captioning, which involves generating descriptive textual information from images. This task is crucial for a variety of applications, including enhancing accessibility for visually impaired individuals, improving functionalities in autonomous vehicles and robotics, and refining image search engines. Despite these advancements, generating accurate and contextually relevant captions from images remains a complex problem due to the intricate nature of visual content and the need for sophisticated linguistic interpretation.

The core challenge in image captioning lies in the ability to understand and describe the diverse and nuanced information contained within images. Images encompass a wide range of elements, including objects, scenes, and their relationships, which can vary greatly. Traditional models often struggle to capture the full complexity of these visual elements, leading to captions that may be incomplete or lack context. For example, while a model might correctly identify that an image contains a dog, it may fail to provide a detailed description of the dog's action or surroundings, resulting in a generic caption that does not fully convey the image's content.

To address these challenges, this project proposes a hybrid image captioning system that combines Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs), enhanced by the Bahdanau Attention mechanism. The system utilizes VGG16, a well-established CNN pre-trained on the ImageNet dataset, for effective feature extraction. VGG16's deep convolutional layers provide a robust representation of the image, capturing essential characteristics that form the bas is for generating descriptive text. However, the mere extraction of features is not sufficient for generating meaningful captions. The complexity of integrating these visual features with linguistic patterns requires a sophisticated approach to aligning visual and textual data.

To improve the generation of contextually relevant captions, the project incorporates the Bahdanau Attention mechanism. This attention mechanism enhances the model's ability to focus on different regions of the image dynamically during each step of caption generation. Unlike traditional attention mechanisms, Bahdanau Attention provides a more refined approach to weighting various parts of the image, allowing the model to generate captions that are more accurate and contextually appropriate. This dynamic focus is crucial for creating detailed and coherent descriptions that align with the visual content.

The Long Short-Term Memory (LSTM) network, a type of RNN, is employed as the decoder for sequential text generation. The LSTM network generates captions based on the features extracted by VGG16 and the attention-enhanced context provided by Bahdanau Attention. By leveraging the LSTM's capability to handle sequential data and maintain contextual information over time, the system ensures that the generated captions are not only coherent but also contextually aligned with the image.

The significance of developing an effective image captioning system extends across multiple domains. For accessibility, it provides visually impaired individuals with meaningful descriptions of their environment, enhancing their ability to interact with the world. In autonomous systems, accurate image captions contribute to better decision-making and interaction capabilities, improving safety and efficiency. In image search engines, descriptive captions enhance search accuracy and user experience by providing more relevant results. Ultimately, this project aims to advance the state-of-the-art in image captioning, contributing to more intelligent and context-aware artificial intelligence systems.

5. METHODOLOGY ADOPTED

The methodology for the image captioning project involves a carefully designed approach that integrates Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and attention mechanisms to generate descriptive text from images. This section details the key components and steps involved in developing and implementing the model.

5.1 FLOW CHART

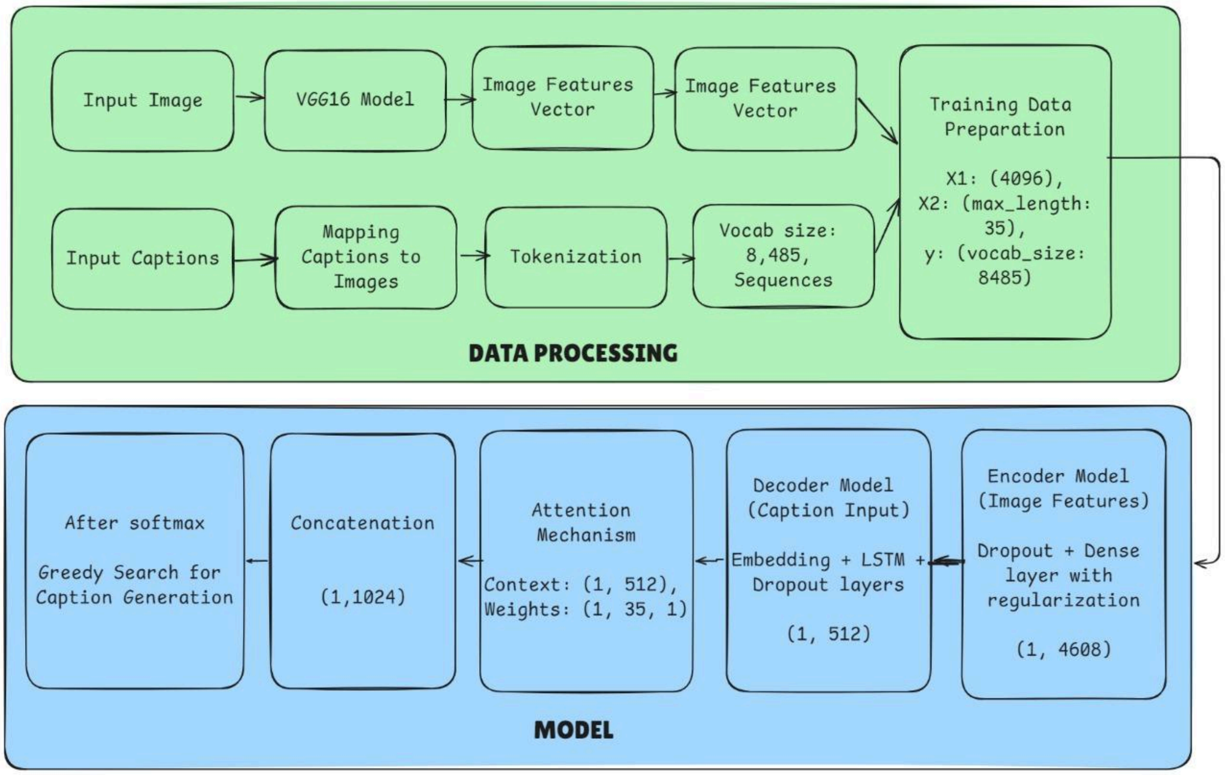


Figure 5.1 End-to-End Model Workflow.

Figure 5.1 illustrates the architecture of an image captioning model using the VGG16 for feature extraction and an attention mechanism for caption generation. It is divided into:

5.1.1 FEATURE EXTRACTION WITH VGG16

The first step in the methodology is feature extraction from images using VGG16, a pre-trained Convolutional Neural Network. VGG16, developed by the Visual Geometry Group at the University of Oxford, is known for its deep architecture and its performance on the ImageNet dataset. It consists of 16 layers, including 13 convolutional layers and 3 fully connected layers, making it adept at capturing hierarchical features from images.

In our project, VGG16 is used as a feature extractor by leveraging its pre-trained weights. The network is initialized with weights trained on the ImageNet dataset, which contains a wide variety of images and classes. This pre-training allows VGG16 to extract rich, high-level features from input images. We utilize the output from the last fully connected layer (before the softmax layer) as the feature vector representing the image. These feature vectors capture the essential characteristics and visual information necessary for generating descriptive captions.

5.1.2 BAHDANAU ATTENTION MECHANISM

The Bahdanau Attention mechanism is employed to enhance the model's ability to generate contextually relevant captions by dynamically focusing on different parts of the image during the captioning process. Introduced by Dzmitry Bahdanau and his colleagues, this attention mechanism improves traditional encoder-decoder models by allowing the decoder to selectively focus on various regions of the image.

The attention mechanism operates in two main steps:

- **Scoring and Alignment:** The attention mechanism computes alignment scores between the current state of the decoder and different regions of the image. This is achieved through a set of learnable parameters that transform the encoder's output and the decoder's current state into a common space. The alignment scores, obtained by applying a score function, indicate the relevance of each region of the image to the current step of the caption generation.

- **Context Vector Calculation:** Based on the alignment scores, the attention mechanism computes a weighted sum of the image features, resulting in a context vector. This context vector represents the region of the image that is most relevant to the current word being generated. The context vector is then used by the decoder to produce the next word in the caption.

By incorporating the Bahdanau Attention mechanism, the model gains the ability to focus on specific parts of the image at each step of the caption generation process, leading to more accurate and contextually appropriate descriptions.

5.1.3 CAPTION GENERATION WITH LSTM

The caption generation process is carried out using a Long Short-Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN) designed to handle sequential data and maintain long-term dependencies. The LSTM network serves as the decoder in the image captioning model, generating sequences of words based on the features extracted by VGG16 and the context vector provided by the Bahdanau Attention mechanism.

The LSTM network operates as follows:

- **Initialization:** The LSTM network is initialized with the start token, indicating the beginning of the caption generation process. The initial input to the LSTM consists of the image feature vector and the start token.

- **Sequence Generation:** At each time step, the LSTM network receives the current word and the context vector as inputs. The network generates the probability distribution of the next word in the sequence, using its internal state to keep track of the context. The most probable word is selected and added to the growing caption sequence.

- **Termination:** The process continues until an end token is generated or the maximum sequence length is reached. The end token signifies the completion of the caption.

5.1.4 DATA PROCESSING

Data preprocessing is a crucial step in preparing the dataset for training the image captioning model. The following preprocessing tasks are performed:

- **Image Resizing:** Images from the Flickr8k dataset are resized to 224x224 pixels to match the input requirements of VGG16. This resizing ensures that the image dimensions are consistent with the network's architecture.

- **Caption Tokenization:** Captions are tokenized into individual words and converted into sequences of integers. This tokenization process involves creating a vocabulary from the captions and mapping each word to a unique integer.

- **Padding:** Sequences of words are padded to a uniform length to ensure that all captions have the same number of tokens. Padding is necessary for batch processing during model training.

5.1.5 MODEL TRAINING AND EVALUATION

The image captioning model is trained using the Flickr8k dataset, which consists of 8,000 images with five captions each. The training process involves optimizing the model's parameters using categorical crossentropy loss and the Adam optimizer. The model's performance is evaluated using the BLEU (Bilingual Evaluation Understudy) score, which measures the quality of generated captions by comparing them to ground truth captions in the dataset.

5.1.6 IMPLEMENTATION AND USER INTERFACE

A user interface is developed using Streamlit, a framework for creating interactive web applications. The interface allows users to upload images and view the generated captions. This provides a practical demonstration of the model's capabilities and enables users to interact with the image captioning system.

5.2. HARDWARE AND SOFTWARE REQUIREMENTS

5.2.1 HARDWARE REQUIREMENTS

1. CPU/GPU:

For developing and training deep learning models, especially those involving large datasets and complex architectures like VGG16 and LSTM networks, a powerful CPU or GPU is essential. The hardware specifications can significantly impact training times and model performance.

- CPU: A multi-core processor with high clock speed, such as Intel i7 or AMD Ryzen 7, is recommended for general tasks and initial development.

- GPU: For accelerated training, a dedicated GPU is crucial. NVIDIA GPUs with CUDA support, such as the NVIDIA GeForce RTX 3060, RTX 3070, or higher, are ideal for handling the computational load of deep learning tasks. The GPU's large memory capacity (8GB or more) allows for processing large batches and complex models efficiently.

2. RAM:

Adequate RAM is necessary to handle large datasets and support smooth execution of the model. A minimum of 16GB RAM is recommended to ensure that the system can handle data preprocessing, model training, and other concurrent tasks without significant performance degradation.

3. Storage:

Sufficient storage is needed to accommodate the datasets, model weights, and other project files. A combination of SSDs for fast read/write operations and HDDs for additional storage is recommended.

- SSD: At least 512GB SSD for the operating system, software, and active project files.

- HDD: Additional 1TB HDD or more for storing large datasets and backups.

4. Network:

A stable and high-speed internet connection is necessary for downloading large datasets, accessing cloud-based resources, and deploying the model online. A connection with at least 10 Mbps download and upload speeds is recommended to ensure efficient data transfers and minimal disruptions.

5.2.2 SOFTWARE REQUIREMENTS

1. Operating System:

The project can be developed and deployed on various operating systems. However, Linux-based systems (such as Ubuntu) are preferred for their compatibility with deep learning libraries and ease of deployment. Windows or macOS can also be used, depending on user preference and compatibility.

2. Python:

Python is the primary programming language used for developing and implementing the model. A Python version compatible with the deep learning libraries used in the project, typically Python 3.7 or later, is required.

3. Deep Learning Libraries:

Several libraries and frameworks are necessary for building and training the image captioning model:

- **TensorFlow/Keras**: For constructing and training the VGG16, Bahdanau Attention, and LSTM components. TensorFlow 2.x or Keras 2.x is recommended for their user-friendly APIs and extensive support.

- **NumPy and Pandas:** For data manipulation, preprocessing, and handling arrays. NumPy provides support for large multi-dimensional arrays, while Pandas offers data structures and operations for manipulating numerical tables and time series.

- **Matplotlib and Seaborn:** For visualizing data, model performance, and training metrics.

4. Flask:

Flask is used for creating a web server to handle API requests and serve the image captioning model. Flask should be installed with the necessary extensions for routing and handling HTTP requests.

5. Streamlit:

Streamlit is utilized for building the user interface to interact with the model. The Streamlit framework simplifies the process of creating interactive web applications and integrates seamlessly with the Flask backend.

These hardware and software requirements ensure that the image captioning project can be developed efficiently, trained effectively, and deployed smoothly, providing a robust and responsive system for generating descriptive captions from images.

5.3. MODEL ARCHITECTURE

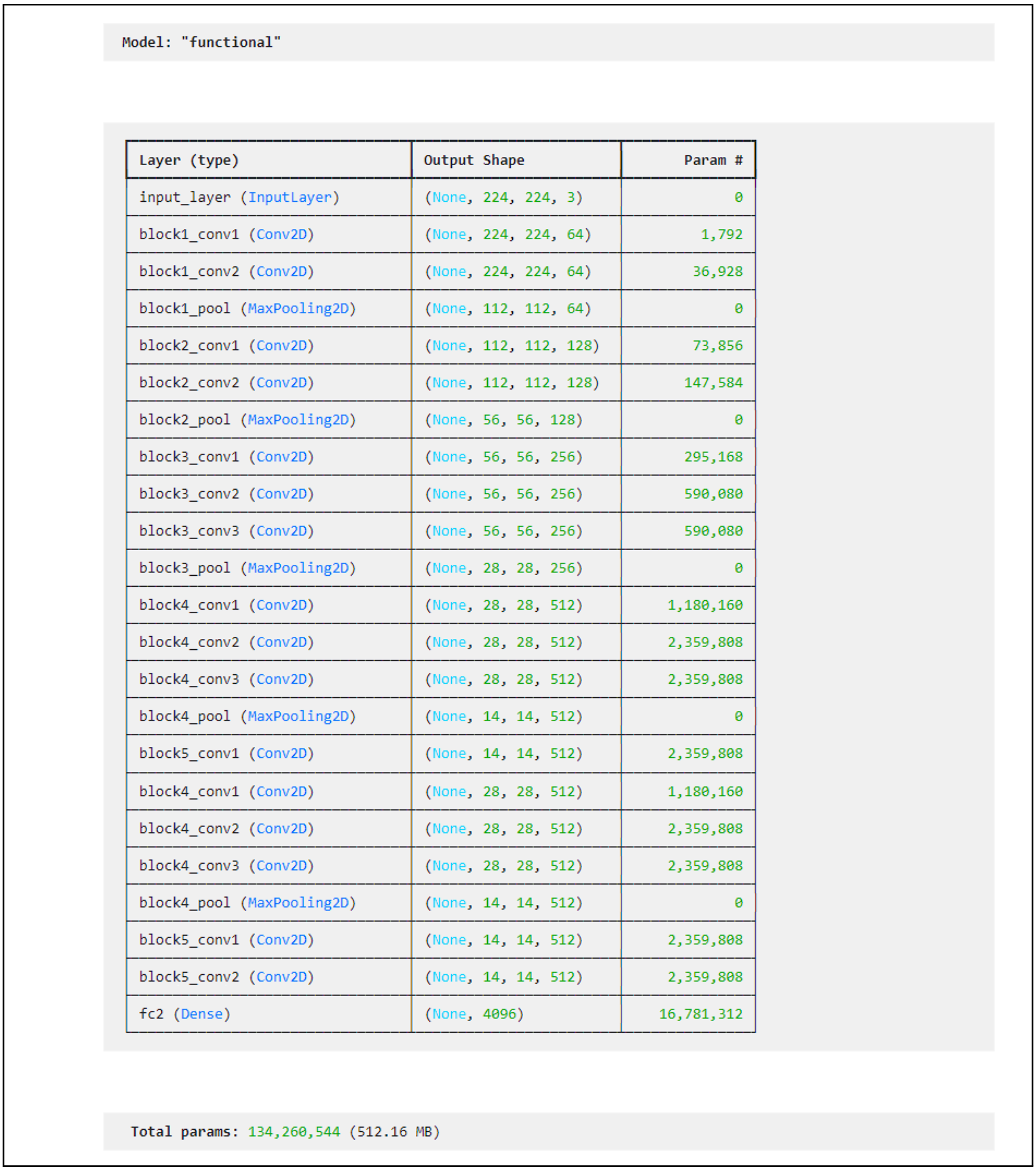
The image captioning model architecture integrates Convolutional Neural Networks (CNNs) for feature extraction, Bahdanau Attention for contextual focus, and Long Short-Term Memory (LSTM) networks for sequential text generation. This combined approach leverages the strengths of each component to generate accurate and descriptive captions for images. The architecture is designed to effectively capture visual features, dynamically focus on relevant image regions, and generate coherent and contextually appropriate textual descriptions.

MODEL 1: VGG16 for Feature Extraction

The foundation of the model is the VGG16 architecture, a widely used Convolutional Neural Network known for its ability to extract high-level features from images. VGG16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers use small 3x3 filters and a max-pooling operation to progressively extract hierarchical features from input images. The final output of the last fully connected layer, just before the softmax layer, is used as the image feature vector.

In the image captioning model, VGG16 is utilized as a pre-trained feature extractor. The model is initialized with weights trained on the ImageNet dataset, enabling it to capture detailed and representative features from images. The feature vector produced by VGG16 serves as a compact yet rich representation of the image, providing the necessary visual information for the caption generation process.

The Below figure 5.3.1 demonstrate the structure of VGG16 model.



```
Model: "functional"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━┓
┃ Layer (type)                 ┃ Output Shape          ┃     Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━┩
│ input_layer (InputLayer)     │ (None, 224, 224, 3)   │           0 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block1_conv1 (Conv2D)        │ (None, 224, 224, 64)  │       1,792 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block1_conv2 (Conv2D)        │ (None, 224, 224, 64)  │      36,928 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block1_pool (MaxPooling2D)   │ (None, 112, 112, 64)  │           0 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block2_conv1 (Conv2D)        │ (None, 112, 112, 128) │      73,856 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block2_conv2 (Conv2D)        │ (None, 112, 112, 128) │     147,584 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block2_pool (MaxPooling2D)   │ (None, 56, 56, 128)   │           0 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block3_conv1 (Conv2D)        │ (None, 56, 56, 256)   │     295,168 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block3_conv2 (Conv2D)        │ (None, 56, 56, 256)   │     590,080 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block3_conv3 (Conv2D)        │ (None, 56, 56, 256)   │     590,080 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block3_pool (MaxPooling2D)   │ (None, 28, 28, 256)   │           0 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_conv1 (Conv2D)        │ (None, 28, 28, 512)   │   1,180,160 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_conv2 (Conv2D)        │ (None, 28, 28, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_conv3 (Conv2D)        │ (None, 28, 28, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_pool (MaxPooling2D)   │ (None, 14, 14, 512)   │           0 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block5_conv1 (Conv2D)        │ (None, 14, 14, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_conv1 (Conv2D)        │ (None, 28, 28, 512)   │   1,180,160 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_conv2 (Conv2D)        │ (None, 28, 28, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_conv3 (Conv2D)        │ (None, 28, 28, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block4_pool (MaxPooling2D)   │ (None, 14, 14, 512)   │           0 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block5_conv1 (Conv2D)        │ (None, 14, 14, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ block5_conv2 (Conv2D)        │ (None, 14, 14, 512)   │   2,359,808 │
├──────────────────────────────┼───────────────────────┼─────────────┤
│ fc2 (Dense)                  │ (None, 4096)          │  16,781,312 │
└──────────────────────────────┴───────────────────────┴─────────────┘

 Total params: 134,260,544 (512.16 MB)
```

Figure 5.3.1 VGG16 Model Architecture.

MODEL 2: Bahdanau Attention Mechanism

To enhance the model's ability to generate contextually relevant captions, the Bahdanau Attention mechanism is incorporated. Bahdanau Attention allows the model to focus on different parts of the image dynamically during the caption generation process. This mechanism improves upon traditional attention models by providing a more refined approach to aligning visual features with textual data.

The Bahdanau Attention mechanism operates in two primary steps:

- Alignment Score Computation
- Context Vector Calculation
-

MODEL 3: LSTM Network for Caption Generation

The Long Short-Term Memory (LSTM) network serves as the core component for sequential text generation in the model. LSTMs are a type of Recurrent Neural Network (RNN) designed to handle sequential data and maintain long-term dependencies, making them well-suited for generating captions from the context provided by the attention mechanism.

The Below figure 5.3.1 demonstrate the Entire Model Architecture.

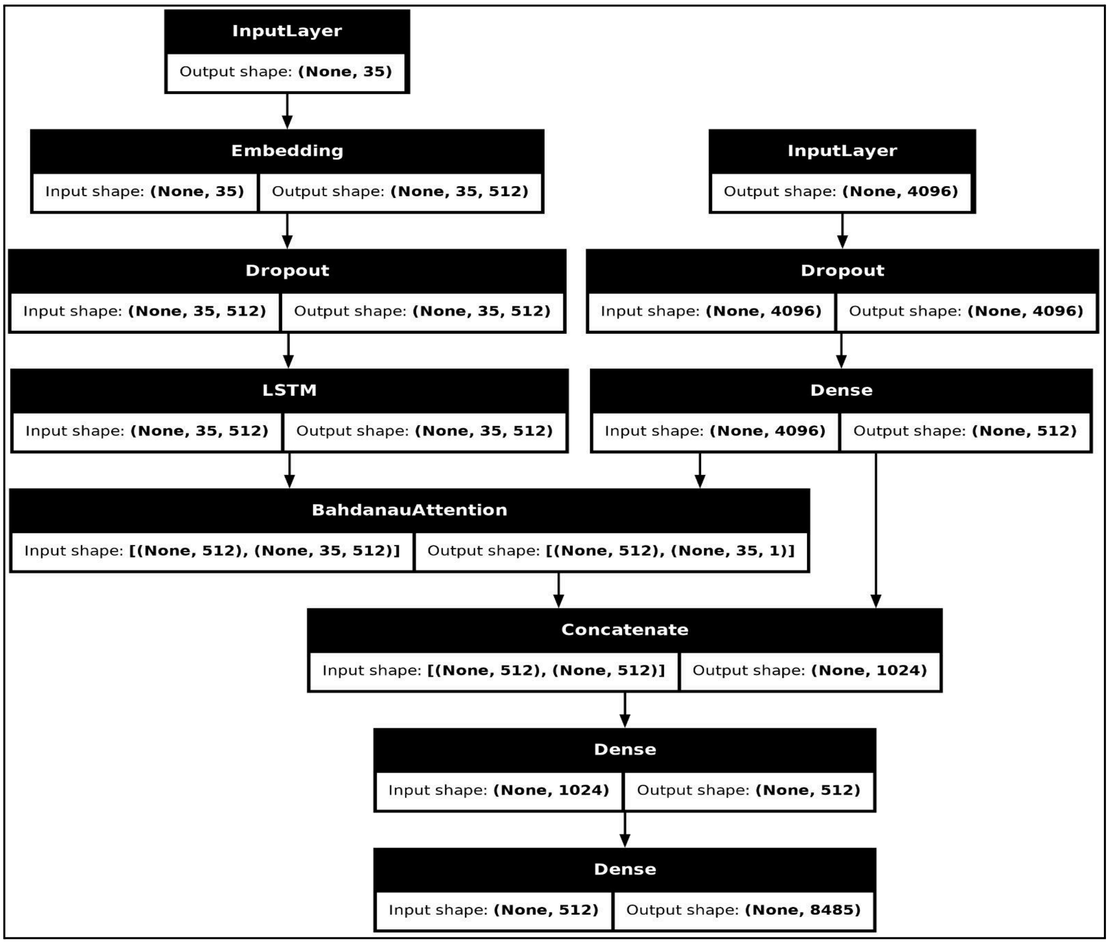
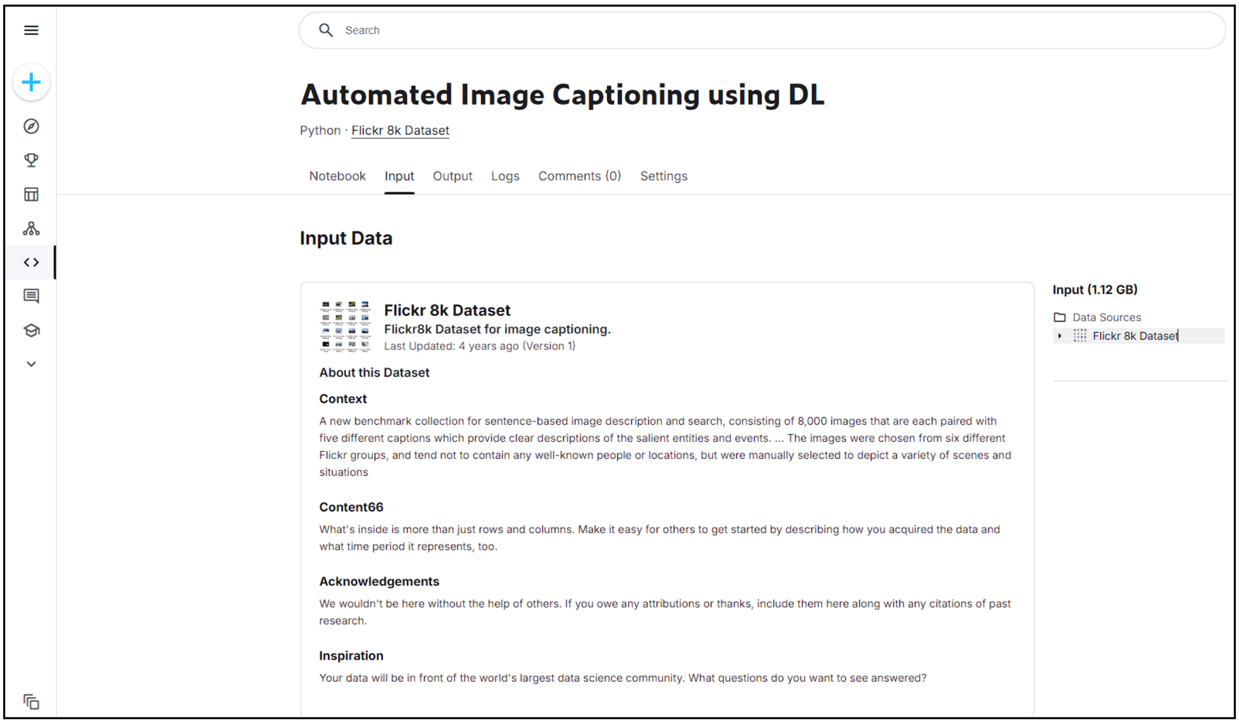

InputLayer
Output shape: (None, 35)
Embedding
Input shape: (None, 35)
Output shape: (None, 35, 512)
InputLayer
Output shape: (None, 4096)
Dropout
Input shape: (None, 35, 512)
Output shape: (None, 35, 512)
Dropout
Input shape: (None, 4096)
Output shape: (None, 4096)
LSTM
Input shape: (None, 35, 512)
Output shape: (None, 35, 512)
Dense
Input shape: (None, 4096)
Output shape: (None, 512)
BahdanauAttention
Input shape: [(None, 512), (None, 35, 512)]
Output shape: [(None, 512), (None, 35, 1)]
Concatenate
Input shape: [(None, 512), (None, 512)]
Output shape: (None, 1024)
Dense
Input shape: (None, 1024)
Output shape: (None, 512)
Dense
Input shape: (None, 512)
Output shape: (None, 8485)

Figure 5.3.2 Entire Model Architecture.

5.4. SCREENSHOTS

In this section, we present snapshots and screenshots capturing key aspects of the entire project, from data preprocessing to model training to deployment, was undertaken.

5.4.1 FLICKR 8K DATASET



Automated Image Captioning using DL

Python · Flickr 8k Dataset

Input Data

Flickr 8k Dataset

Flickr8k Dataset for image captioning.

Last Updated: 4 years ago (Version 1)

About this Dataset

Context

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

Content66

What's inside is more than just rows and columns. Make it easy for others to get started by describing how you acquired the data and what time period it represents, too.

Acknowledgements

We wouldn't be here without the help of others. If you owe any attributions or thanks, include them here along with any citations of past research.

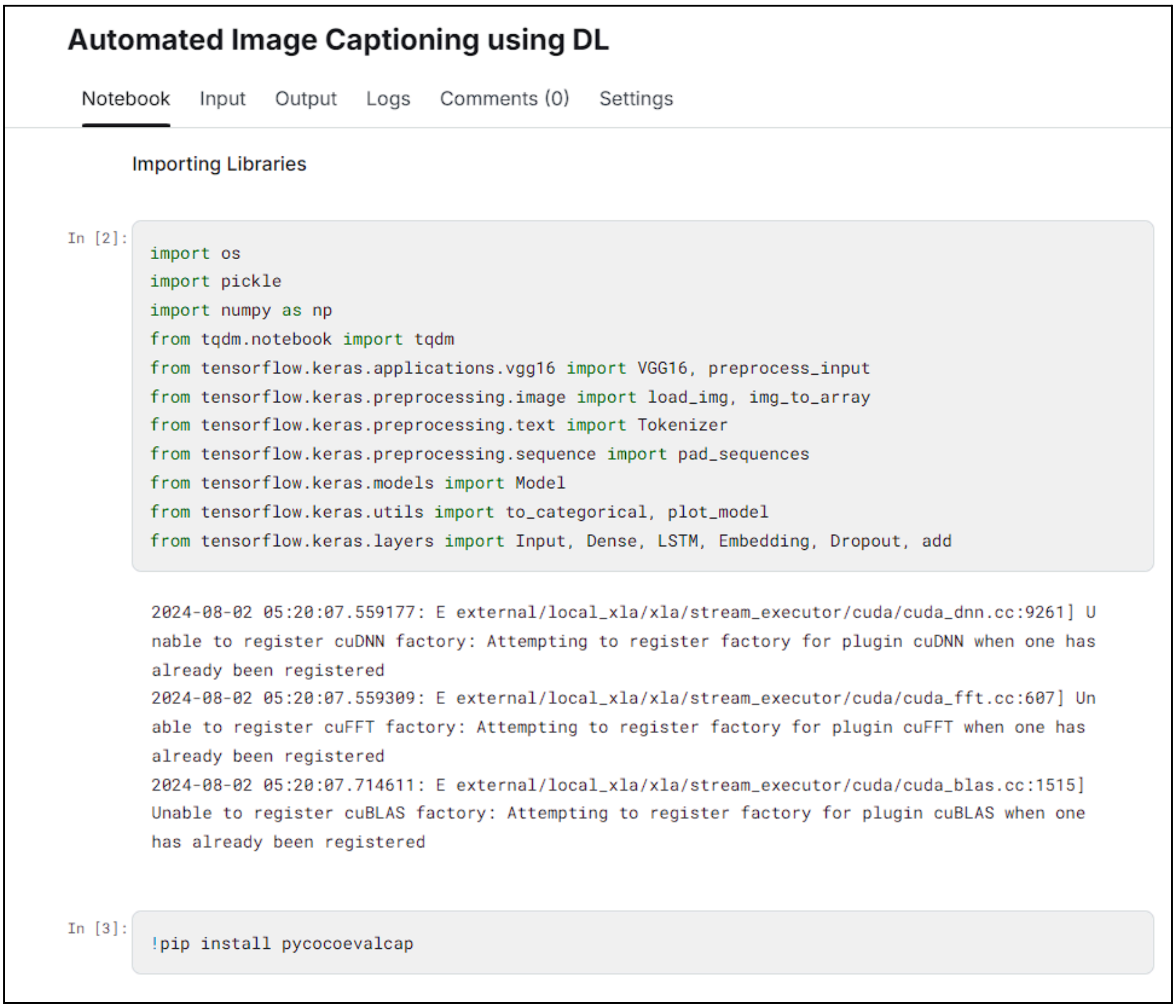
Inspiration

Your data will be in front of the world's largest data science community. What questions do you want to see answered?

Figure 5.4.1 A snapshot providing an overview of the dataset used.

5.4.2 CODE IMPLEMENTATION

5.4.2.1 LOADING LIBRARIES



```python
import os
import pickle
import numpy as np
from tqdm.notebook import tqdm
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Model
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add
```

```
2024-08-02 05:20:07.559177: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] U
nable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has
already been registered
2024-08-02 05:20:07.559309: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Un
able to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
2024-08-02 05:20:07.714611: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515]
Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one
has already been registered
```

```
!pip install pycocoevalcap
```

Figure 5.4.2.1 Importing essential libraries for data preprocessing and deep learning.

5.4.2.2 FEATURES EXTRACTED USING VGG16 MODEL

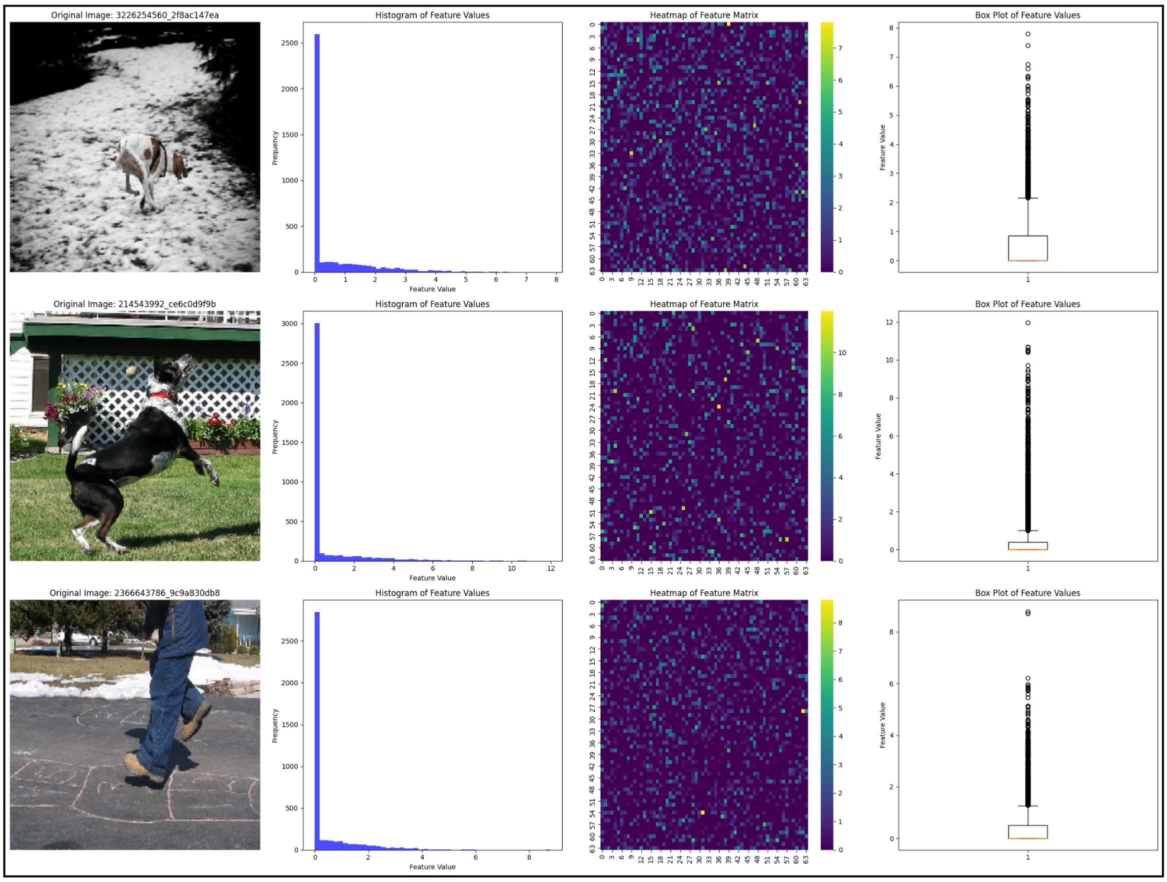
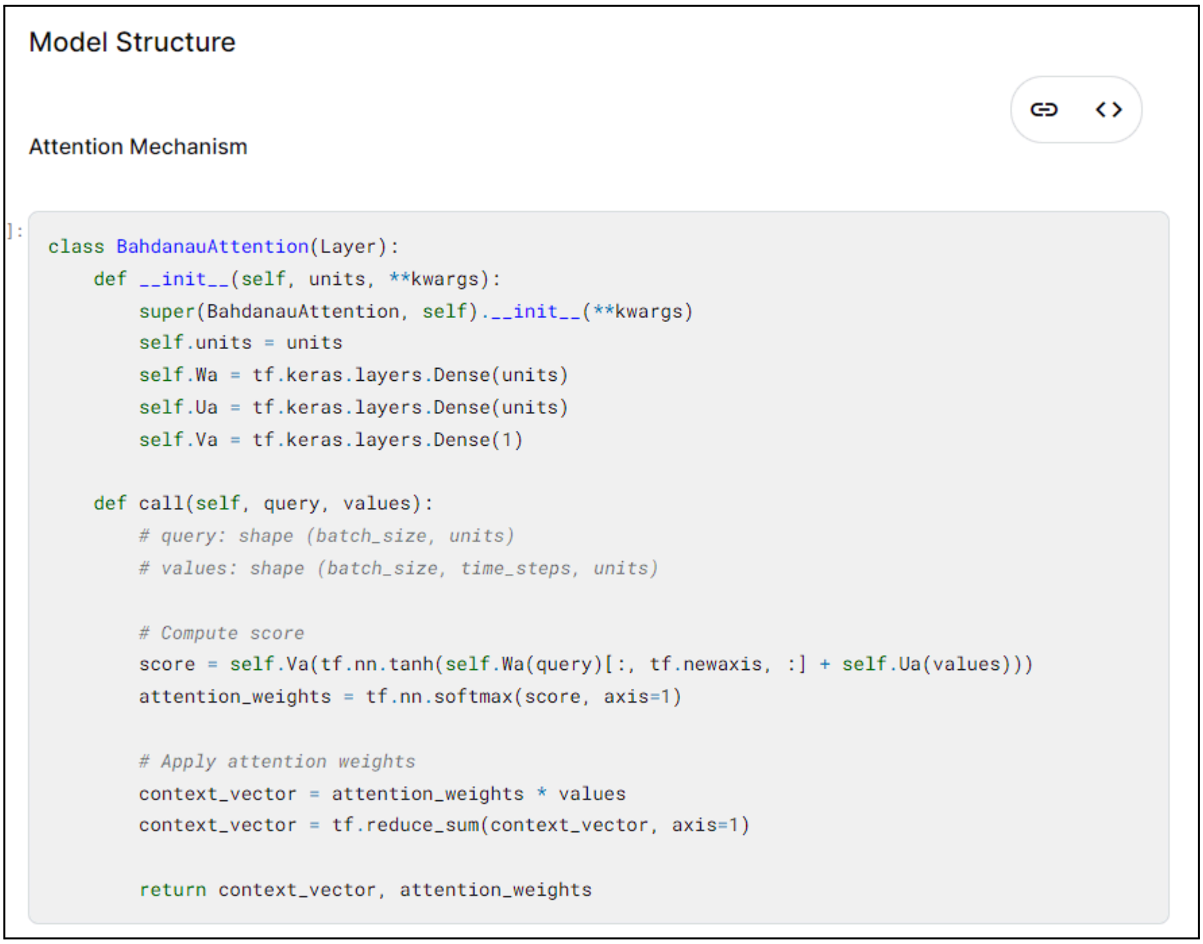


Figure 5.4.2.2 Features Extracted using VGG16 after preprocessing of Images.

5.4.2.3 MODEL STRUCTURE

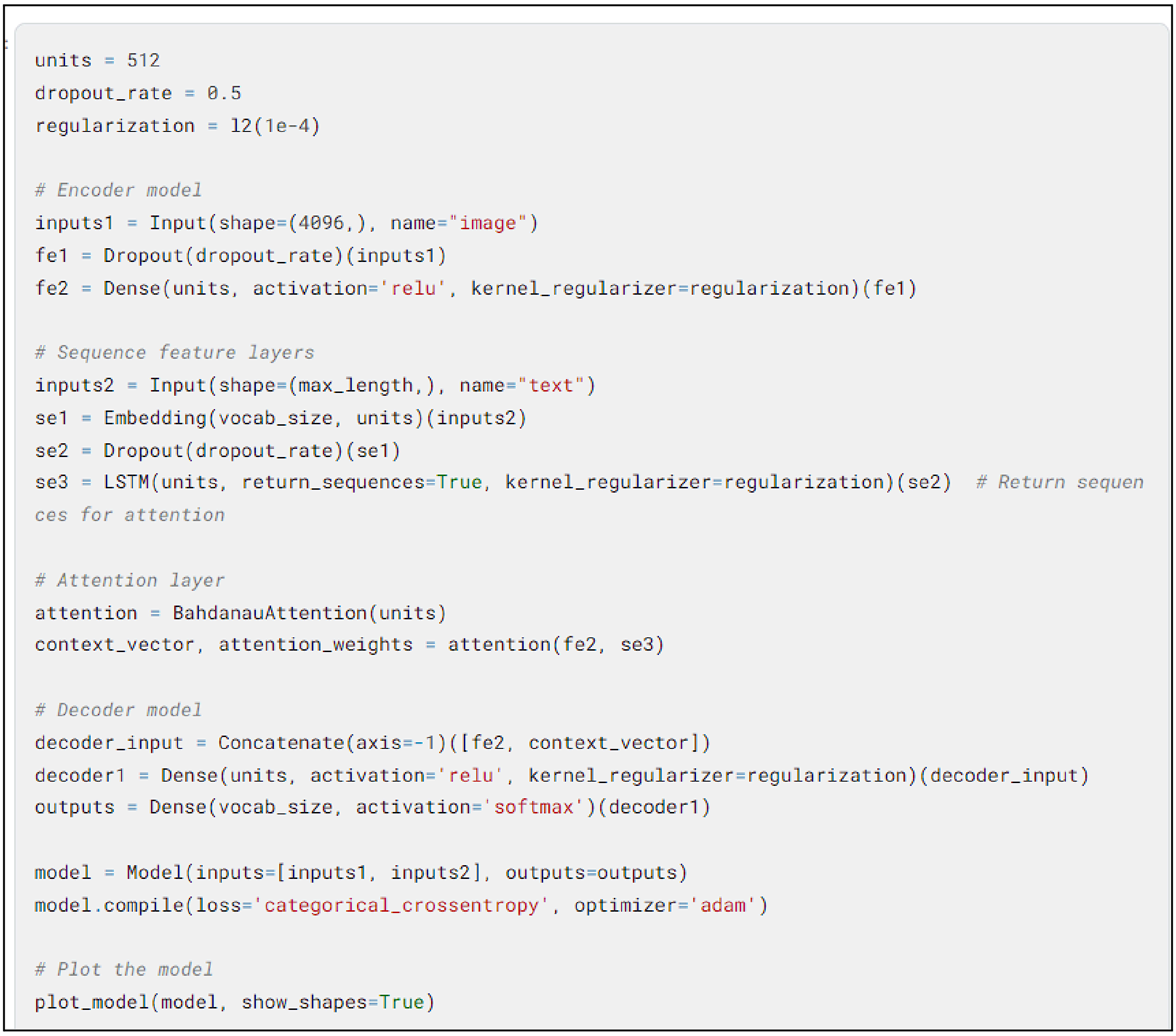


Model Structure

Attention Mechanism

```
class BahdanauAttention(Layer):
    def __init__(self, units, **kwargs):
        super(BahdanauAttention, self).__init__(**kwargs)
        self.units = units
        self.Wa = tf.keras.layers.Dense(units)
        self.Ua = tf.keras.layers.Dense(units)
        self.Va = tf.keras.layers.Dense(1)

    def call(self, query, values):
        # query: shape (batch_size, units)
        # values: shape (batch_size, time_steps, units)

        # Compute score
        score = self.Va(tf.nn.tanh(self.Wa(query)[:, tf.newaxis, :] + self.Ua(values)))
        attention_weights = tf.nn.softmax(score, axis=1)

        # Apply attention weights
        context_vector = attention_weights * values
        context_vector = tf.reduce_sum(context_vector, axis=1)

        return context_vector, attention_weights
```

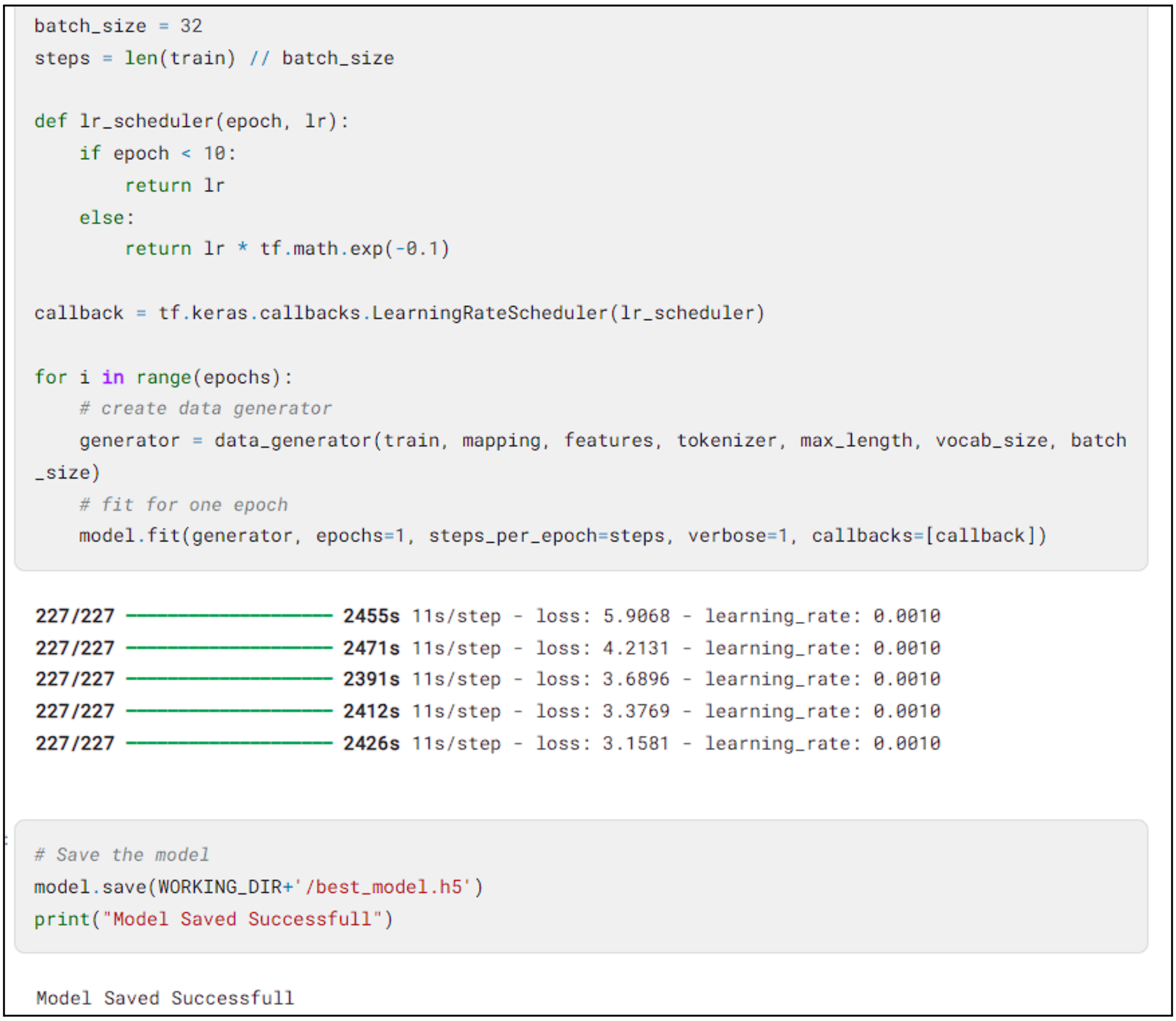
Figure 5.4.2.3.1 Bahdanau Attention Mechanism Code Snippet.



```python
units = 512
dropout_rate = 0.5
regularization = l2(1e-4)

# Encoder model
inputs1 = Input(shape=(4096,), name="image")
fe1 = Dropout(dropout_rate)(inputs1)
fe2 = Dense(units, activation='relu', kernel_regularizer=regularization)(fe1)

# Sequence feature layers
inputs2 = Input(shape=(max_length,), name="text")
se1 = Embedding(vocab_size, units)(inputs2)
se2 = Dropout(dropout_rate)(se1)
se3 = LSTM(units, return_sequences=True, kernel_regularizer=regularization)(se2)  # Return sequences for attention

# Attention layer
attention = BahdanauAttention(units)
context_vector, attention_weights = attention(fe2, se3)

# Decoder model
decoder_input = Concatenate(axis=-1)([fe2, context_vector])
decoder1 = Dense(units, activation='relu', kernel_regularizer=regularization)(decoder_input)
outputs = Dense(vocab_size, activation='softmax')(decoder1)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam')

# Plot the model
plot_model(model, show_shapes=True)
```

Figure 5.4.2.3.2 Building the architecture of the Encoder and Decorder integrated with above attention mechanism.

5.4.2.4 TRAINING THE MODEL



```
batch_size = 32
steps = len(train) // batch_size

def lr_scheduler(epoch, lr):
    if epoch < 10:
        return lr
    else:
        return lr * tf.math.exp(-0.1)

callback = tf.keras.callbacks.LearningRateScheduler(lr_scheduler)

for i in range(epochs):
    # create data generator
    generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch
_size)
    # fit for one epoch
    model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1, callbacks=[callback])
```

```
227/227 ━━━━━━━━━━━━━━━━━━━━ 2455s 11s/step - loss: 5.9068 - learning_rate: 0.0010
227/227 ━━━━━━━━━━━━━━━━━━━━ 2471s 11s/step - loss: 4.2131 - learning_rate: 0.0010
227/227 ━━━━━━━━━━━━━━━━━━━━ 2391s 11s/step - loss: 3.6896 - learning_rate: 0.0010
227/227 ━━━━━━━━━━━━━━━━━━━━ 2412s 11s/step - loss: 3.3769 - learning_rate: 0.0010
227/227 ━━━━━━━━━━━━━━━━━━━━ 2426s 11s/step - loss: 3.1581 - learning_rate: 0.0010
```

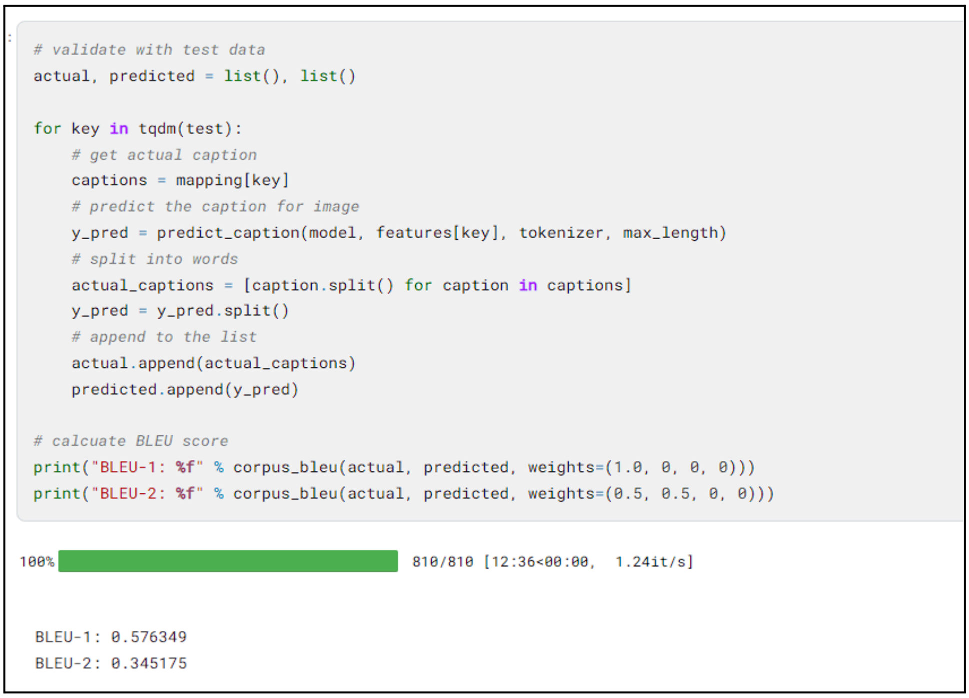
```
# Save the model
model.save(WORKING_DIR+'/best_model.h5')
print("Model Saved Successfull")
```

```
Model Saved Successfull
```

Figure 5.4.2.4 Initiating and monitoring the training process with the aid of the Kaggle GPU.

5.4.3 RESULT ANALYSIS

5.4.3.1 EVALUATION

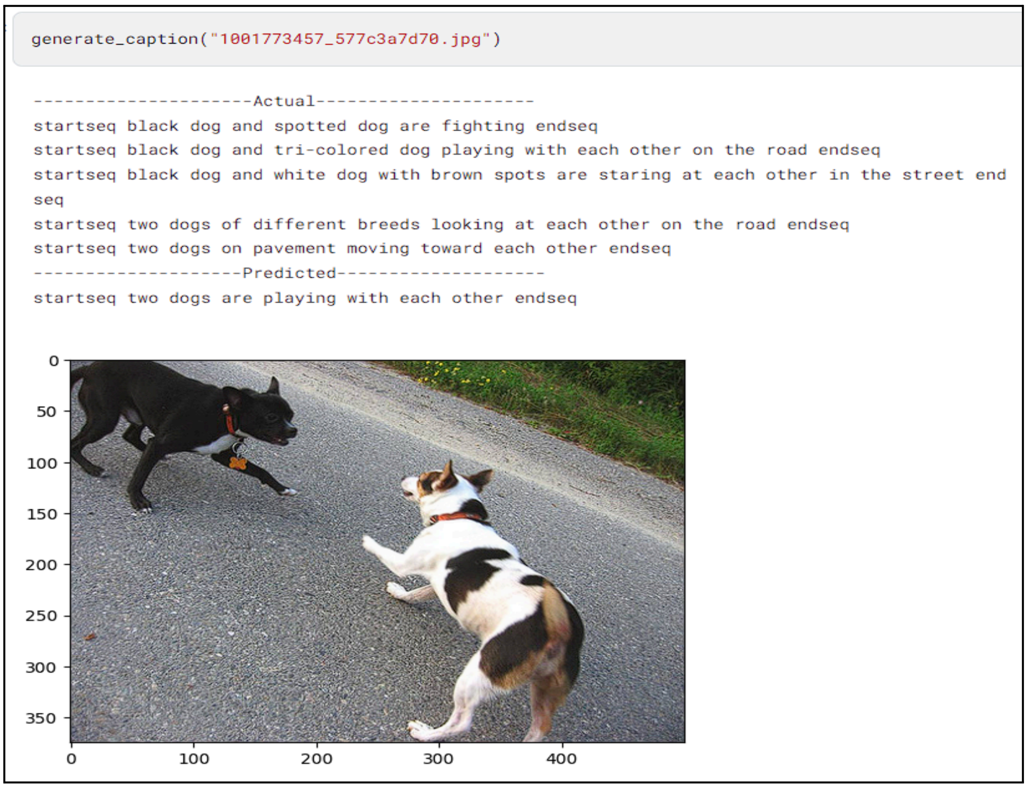


```
# validate with test data
actual, predicted = list(), list()

for key in tqdm(test):
    # get actual caption
    captions = mapping[key]
    # predict the caption for image
    y_pred = predict_caption(model, features[key], tokenizer, max_length)
    # split into words
    actual_captions = [caption.split() for caption in captions]
    y_pred = y_pred.split()
    # append to the list
    actual.append(actual_captions)
    predicted.append(y_pred)

# calcuate BLEU score
print("BLEU-1: %f" % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
print("BLEU-2: %f" % corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
```

```
100%                                      810/810 [12:36<00:00,  1.24it/s]

BLEU-1: 0.576349
BLEU-2: 0.345175
```

Figure 5.4.3.1 Get the BLEU-1 Score on test Data to be 0.57% and BLEU-2 Score 0.34%.

5.4.3.2 PREDICTION ON A IMAGE

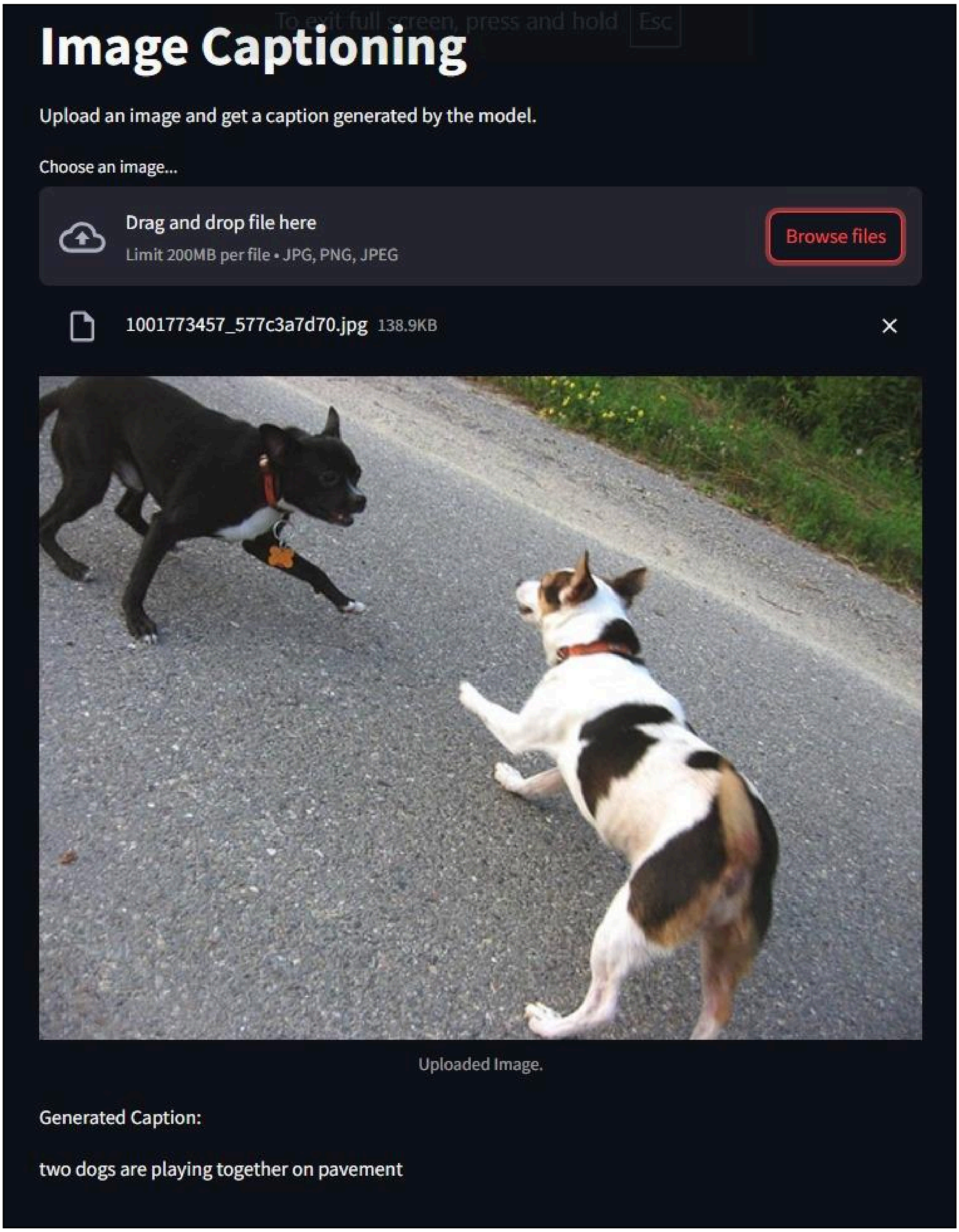


```
generate_caption("1001773457_577c3a7d70.jpg")
```

```
---------------------Actual---------------------
startseq black dog and spotted dog are fighting endseq
startseq black dog and tri-colored dog playing with each other on the road endseq
startseq black dog and white dog with brown spots are staring at each other in the street end
seq
startseq two dogs of different breeds looking at each other on the road endseq
startseq two dogs on pavement moving toward each other endseq
--------------------Predicted-------------------
startseq two dogs are playing with each other endseq
```

Figure 5.4.3.2 Model prediction on an image.

5.4.3.3 MODEL DEPLOYMENT ON STREAMLIT



To exit full screen, press and hold Esc

Image Captioning

Upload an image and get a caption generated by the model.

Choose an image...

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

Browse files

1001773457_577c3a7d70.jpg 138.9KB ×

Uploaded Image.

Generated Caption:

two dogs are playing together on pavement

Figure 5.4.3.3 Model prediction after deployment on streamlit.

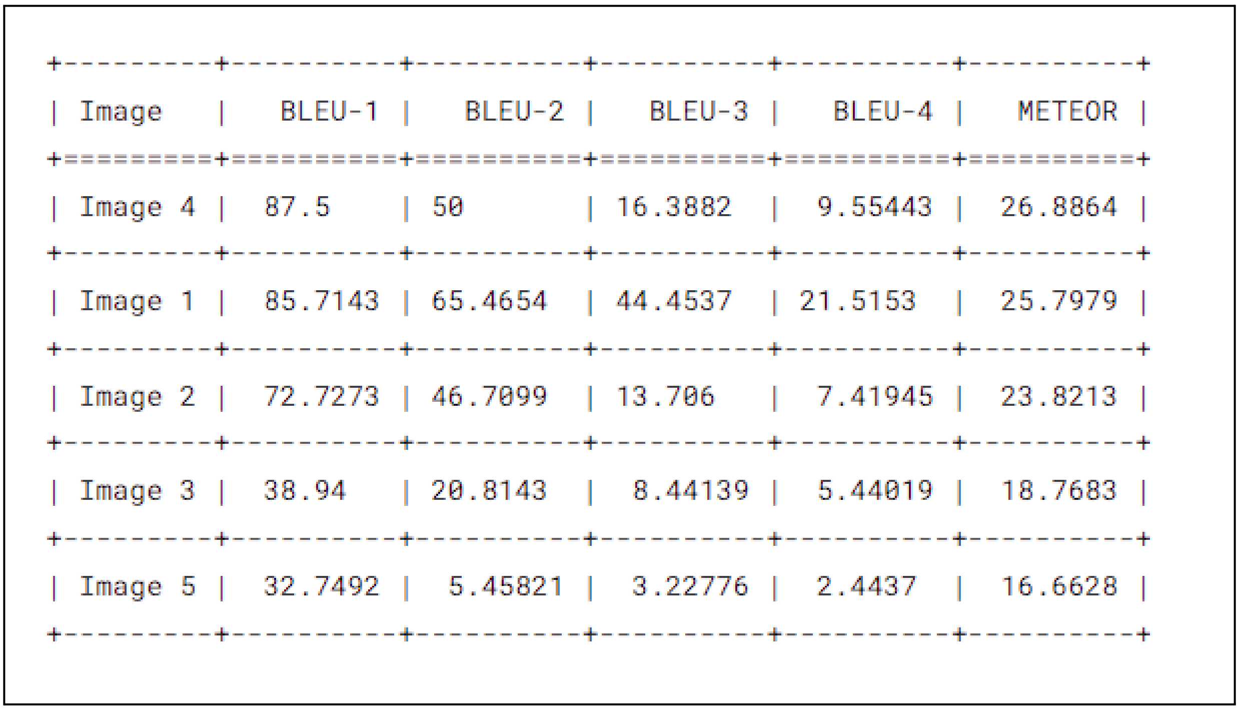
The snapshots above provide a glimpse into the workflow and execution of the project.

6. RESULTS OBTAINED

In this section, we present a comprehensive analysis of the outcomes achieved through the development and implementation of the Encoder and Decoder (Model) using Attention Mechanism by comparing results on different images individually.

6.1 MODEL SCORE ON RANDOM 5 IMAGES

The figure 6.1 below show 5 different model evaluation metrics on 5 random images and it accuracy :



```
+---------+----------+----------+----------+----------+----------+
| Image   |   BLEU-1 |   BLEU-2 |   BLEU-3 |   BLEU-4 |   METEOR |
+=========+==========+==========+==========+==========+==========+
| Image 4 | 87.5     | 50       | 16.3882  |  9.55443 |  26.8864 |
+---------+----------+----------+----------+----------+----------+
| Image 1 | 85.7143  | 65.4654  | 44.4537  | 21.5153  |  25.7979 |
+---------+----------+----------+----------+----------+----------+
| Image 2 | 72.7273  | 46.7099  | 13.706   |  7.41945 |  23.8213 |
+---------+----------+----------+----------+----------+----------+
| Image 3 | 38.94    | 20.8143  |  8.44139 |  5.44019 |  18.7683 |
+---------+----------+----------+----------+----------+----------+
| Image 5 | 32.7492  |  5.45821 |  3.22776 |  2.4437  |  16.6628 |
+---------+----------+----------+----------+----------+----------+
```

Figure 6.1 Different Evaluation metrics results.

7. CONCLUSION

The development of an image captioning system leveraging the VGG16 model with Bahdanau Attention and LSTM networks represents a significant advancement in the field of artificial intelligence and computer vision. This project successfully integrates state-of-the-art techniques to address the complex challenge of generating descriptive and contextually relevant captions from images. Through a comprehensive approach involving feature extraction, dynamic attention mechanisms, and sequential text generation, the system demonstrates the ability to produce high-quality captions that accurately reflect the content and context of the images.

The use of VGG16 as a feature extractor provides a solid foundation for understanding visual content. Pre-trained on the ImageNet dataset, VGG16 captures rich and detailed image features, which are essential for the subsequent caption generation process. The deep convolutional layers of VGG16 extract hierarchical features that represent various aspects of the image, such as objects, textures, and spatial relationships. By leveraging these features, the model can effectively interpret and describe a wide range of visual elements.

Incorporating the Bahdanau Attention mechanism enhances the model's performance by allowing it to focus dynamically on different regions of the image during each step of the caption generation. This attention mechanism improves the alignment between the visual features and the generated text, leading to more accurate and contextually appropriate captions. The ability to selectively attend to relevant parts of the image ensures that the captions are not only descriptive but also coherent and aligned with the visual content.

The LSTM network, used as the decoder, excels in handling sequential data and generating text based on the context provided by the attention mechanism. The LSTM's capability to maintain long-term dependencies and generate sequences of words enables the model to produce complete and coherent captions. By integrating the attention-enhanced context vector into the LSTM's decoding process, the system ensures that each word generated is relevant to both the current image features and the previously generated words.

The project's implementation on Kaggle allowed for the effective development and training of the model using the Flickr8k dataset, which provides a diverse set of images and captions for evaluating the system's performance. The integration of Flask and Streamlit for deployment offers a practical and user-friendly interface for interacting with the image captioning model.

This deployment setup enables users to upload images and receive generated captions in real-time, showcasing the model's capabilities and providing a valuable tool for various applications.

The results of this project underscore the potential of combining advanced deep learning techniques for image captioning. The model successfully generates descriptive captions for a range of images, demonstrating its ability to understand and articulate visual content. The use of the BLEU score as an evaluation metric provides quantitative insight into the quality of the generated captions, indicating that the model performs well in aligning with ground truth captions.

Looking ahead, there are several avenues for future work to further enhance the capabilities of the image captioning system. Exploring transformer-based architectures could offer improvements in sequence generation and attention mechanisms. Training the model on larger and more diverse datasets, such as MS COCO, may enhance its generalization and performance across different image types. Additionally, incorporating additional evaluation metrics, such as METEOR or CIDEr, could provide a more comprehensive assessment of caption quality.

In conclusion, this project represents a significant step forward in the field of image captioning, combining powerful deep learning techniques to generate accurate and contextually relevant textual descriptions from images. The successful implementation and deployment of the model demonstrate its potential for real-world applications and set the stage for future advancements in this dynamic and evolving field.

8. FUTURE WORK

The image captioning project has laid a strong foundation with the integration of VGG16, Bahdanau Attention, and LSTM networks. However, several areas present opportunities for further enhancement:

1. Transformer-Based Architectures: Exploring transformer models, such as the Vision Transformer (ViT) or models with encoder-decoder structures like BERT and GPT, could improve the efficiency and quality of caption generation. Transformers have demonstrated superior performance in various NLP tasks and could potentially enhance the model's ability to generate more coherent and contextually rich captions.

2. Larger Datasets: Expanding training to larger and more diverse datasets, such as the MS COCO dataset, could improve the model's generalization and robustness. Larger datasets provide a wider variety of images and captions, helping the model learn more comprehensive and varied descriptions.

3. Additional Evaluation Metrics: Incorporating additional metrics such as METEOR, CIDEr, or ROUGE could provide a more nuanced evaluation of caption quality. These metrics can capture different aspects of caption accuracy and relevance, offering a more thorough assessment of the model’s performance.

4. Interactive Features: Enhancing the user interface with more interactive features and capabilities could improve the overall user experience. Integrating functionalities such as real-time feedback and model refinement based on user inputs may offer valuable insights and further refine the system.

These future directions aim to advance the capabilities of the image captioning model and expand its applicability across diverse domains and scenarios.

9. CHALLENGES

The image captioning project encountered several challenges that impacted its development and performance:

1. **Complexity of Caption Generation:** Generating coherent and contextually relevant captions is inherently complex due to the need to accurately interpret diverse visual content and produce grammatically correct text. The model must handle various image types and contexts, which poses significant challenges in maintaining consistency and relevance across different scenarios.
2. **Attention Mechanism Integration:** While Bahdanau Attention enhances the model's ability to focus on relevant image regions, integrating it effectively with the LSTM network posed challenges. Ensuring that the attention mechanism accurately aligns with the visual features and generates meaningful context for each decoding step required careful tuning and optimization.
3. **Training Data Limitations:** The Flickr8k dataset, while valuable, is relatively small compared to other datasets like MS COCO. Limited data can constrain the model's ability to generalize and handle a wide range of images and contexts. Addressing this limitation requires additional data collection and preprocessing.
4. **Computational Resources:** Training deep learning models, especially those involving large networks and datasets, demands substantial computational resources. Managing and optimizing GPU usage, memory, and processing time were critical challenges, particularly for large-scale training tasks.

These challenges highlight areas for improvement and offer insights into potential enhancements for future iterations of the image captioning system.

10. REFERENCES

1. Wang, P.; Yang, A.; Men, R.; Lin, J.; Bai, S.; Li, Z.; Ma, J.; Zhou, C.; Zhou, J.; Yang, H. OFA: Unifying Architectures, Tasks,and Modalities Through a Simple Sequence-to-Sequence Learning Framework. arXiv 2022, arXiv:2202.03052. Available online: https://arxiv.org/abs/2202.03052 (accessed on 14 July 2022).
2. Hsu, T.Y.; Giles, C.L.; Huang, T.H. SCICAP: Generating Captions for Scientific Figures. In Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021.
3. Hossain, M.Z.; Sohel, F.; Shiratuddin, M.F.; Laga, H.; Bennamoun, M. Text to Image Synthesis for Improved Image Captioning. IEEE Access 2021.
4. Sehgal, S.; Sharma, J.; Chaudhary, N. Generating Image Captions Based on Deep Learning and Natural Language Processing. In Proceedings of the ICRITO 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trendsand Future Directions) IEEE, Noida, India, 4–5 June 2020.
5. Jain, H.; Zepeda, J.; Perez, P.; Gribonval, R. Learning a Complete Image Indexing Pipeline. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
6. Pang, S.; Orgun, M.A.; Yu, Z. A Novel Biomedical Image Indexing and Retrieval System via Deep Preference Learning. Comput.Methods Prog. Biomed. 2018.
7. Makav, B.; Kilic, V. A New Image Captioning Approach for Visually Impaired People. In Proceedings of the 11th InternationalConference on Electrical and Electronics Engineering (ELECO 2019), Bursa, Turkey, 28–30 November 2019.
8. Zhang, Z.; Wu, Q.; Wang, Y.; Chen, F. High-Quality Image Captioning with Fine-Grained and Semantic-Guided Visual Attention. IEEE Trans. Multimed. 2019.
9. Alam, S.; Raja, P.; Gulzar, Y. Investigation of Machine Learning Methods for Early Prediction of Neurodevelopmental Disordersin Children. Wirel. Commun. Mob. Comput. 2022.
10. Sahlan, F.; Hamidi, F.; Misrat, M.Z.; Adli, M.H.; Wani, S.; Gulzar, Y. Prediction of Mental Health Among University Students. Int. J. Perceptive Cogn. Comput. 2021.
11. Khan, S.A.; Gulzar, Y.; Turaev, S.; Peng, Y.S. A Modified HSIFT Descriptor for Medical Image Classification of Anatomy Objects. Symmetry 2021.

12. Gulzar, Y.; Khan, S.A. Skin Lesion Segmentation Based on Vision Transformers and Convolutional Neural Networks—AComparative Study. Appl. Sci. 2022.
13. Albarrak, K.; Gulzar, Y.; Hamid, Y.; Mehmood, A.; Soomro, A.B. A Deep Learning-Based Model for Date Fruit Classification.Sustainability 2022.
14. Gulzar, Y.; Hamid, Y.; Soomro, A.B.; Alwan, A.A.; Journaux, L. A Convolution Neural Network-Based Seed Classification System.Symmetry 2020.
15. Hamid, Y.; Wani, S.; Soomro, A.B.; Alwan, A.A.; Gulzar, Y. Smart Seed Classification System Based on MobileNetV2 Architecture. In Proceedings of the 2nd International Conference on Computing and Information Technology, ICCIT 2022, Tabuk, Saudi Arabia, 25–27 January 2022.
16. Hamid, Y.; Elyassami, S.; Gulzar, Y.; Balasaraswathi, V.R.; Habuza, T.; Wani, S. An Improvised CNN Model for Fake Image Detection. Int. J. Inf. Technol. 2022.
17. Faris, M.; Hanafi, F.M.; Sukri Faiz, M.; Nasir, M.; Wani, S.; Abdulkhaleq, R.; Abdulghafor, A.; Gulzar, Y.; Hamid, Y. A Real Time Deep Learning Based Driver Monitoring System. Int. J. Perceptive Cogn. Comput. 2021.
18. Sharma, H.; Jalal, A.S. Incorporating External Knowledge for Image Captioning Using CNN and LSTM. Mod. Phys. Lett. B 2020.
19. Wang, C.; Yang, H.; Bartz, C.; Meinel, C. Image Captioning with Deep Bidirectional LSTMs. In Proceedings of the 2016 ACM Multimedia Conference, Amsterdam, The Netherlands, 15–19 October 2016.
20. Aneja, J.; Deshpande, A.; Schwing, A.G. Convolutional Image Captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
21. Yang, X.; Zhang, H.; Cai, J. Learning to Collocate Neural Modules for Image Captioning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2019.
22. Khan, R.; Islam, M.S.; Kanwal, K.; Iqbal, M.; Hossain, M.I.; Ye, Z. A Deep Neural Framework for Image Caption Generation UsingGRU-Based Attention Mechanism. arXiv 2022.
23. Huang, L.; Wang, W.; Chen, J.; Wei, X.Y. Attention on Attention for Image Captioning. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
24. Hodosh, M.; Young, P.; Hockenmaier, J. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. In Proceedings of the Twenty-Fourth

International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, 25–31 July 2015.

25. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and Tell: A Neural Image Caption Generator. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
26. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Con-volutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 2018.
27. Karpathy, A.; Li, F.-F. Deep Visual-Semantic Alignments for Generating Image Descriptions. IEEE Trans. Pattern Anal. Mach. Intell. 2017.
28. Li, L.; Tang, S.; Zhang, Y.; Deng, L.; Tian, Q. GLA: Global-Local Attention for Image Description. IEEE Trans. Multimed. 2018.
29. Ding, G.; Chen, M.; Zhao, S.; Chen, H.; Han, J.; Liu, Q. Neural Image Caption Generation with Weighted Training and Reference. Cogn. Comput. 2019.
30. Yan, S.; Xie, Y.; Wu, F.; Smith, J.S.; Lu, W.; Zhang, B. Image Captioning via Hierarchical Attention Mechanism and Policy GradientOptimization. Signal Process. 2020.
31. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
32. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May2015.