SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Image Captioning System using Deep Neural Network based on Encoder-Decoder Framework.

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING
(Computer Engineering)

SUBMITTED BY

Group ID : A19

Mr. Mayur Sopan Gadakh	Exam No: B190104240
Mr. Gaurav Bhima Chaudhari	Exam No: B190104219
Ms. Akanksha Bhausaheb Gaikwad	Exam No: B190104245
Ms. Shivanjali Anil Dhage	Exam No: B190104229

Under The Guidance of

Dr. R. S. Gaikwad

DEPARTMENT OF COMPUTER ENGINEERING
Amrutvahini College of Engineering, Sangamner
Amrutnagar, Ghulewadi - 422608

2023-24

AMRUTVAHINI COLLEGE OF ENGINEERING,SANGAMNER
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Image Captioning system using deep neural network based on encoder-decoder framework.

Submitted by

Group ID: A19

Mr. Mayur Sopan Gadakh	Exam No: B190104240
Mr. Gaurav Bhima Chaudhari	Exam No: B190104219
Ms. Akanksha Bhausaheb Gaikwad	Exam No: B190104245
Ms. Shivanjali Anil Dhage	Exam No: B190104229

are bonafide students of this institute and the work has been carried out by them under the supervision of **Dr. R. S. Gaikwad** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

Dr. R. S. Gaikwad
Internal Guide
Dept. of Computer Engg.

Dr. R. G. Tambe / Dr. D. R. Patil
Project Coordinator
Dept. of Computer Engg.

Dr. S. K. Sonkar
H.O.D.
Dept. of Computer Engg.

Dr. M.A. Venkatesh
Principal
AVCOE Sangamner

SAVITRIBAI PHULE PUNE UNIVERSITY

CERTIFICATE

This is to certify that,

Group ID: A19

Mr. Mayur Sopan Gadakh	Exam No: B190104240
Mr. Gaurav Bhima Chaudhari	Exam No: B190104219
Ms. Akanksha Bhausaheb Gaikwad	Exam No: B190104245
Ms. Shivanjali Anil Dhage	Exam No: B190104229

of BE Computer Engineering was examined in the Project Examination entitled

Image Captioning system using deep neural network based on encoder-decoder framework.

on / / 2024

At

DEPARTMENT OF COMPUTER ENGINEERING

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

_____ _____

Internal Examiner External Examiner

Acknowledgment

Achievement is Finding out what you have been doing and what you have to do. The higher is submit, the harder is climb. The goal was fixed and We began with the determined resolved and put in a ceaseless sustained hard work. Greater the challenge, greater was our determination and it guided us to overcome all difficulties. It has been rightly said that we are built on the shoulders of others. For everything We have achieved, the credit goes to who had really help us to complete this project and for the timely guidance and infrastructure. Before we proceed any further, we would like to thank all those who have helped us in all the way through. We are thankful to our project guide **Dr. R. S. Gaikwad** for their guidance guidance care and support,which they offered whenever we needed it. We would like to thanks to project coordinator **Dr. R. G. Tambe** and **Dr. D. R. Patil** and also the respected Head of Department **Dr. S. K. Sonkar**.We would also thankful to Honourable Principal **Dr. M. A. Vankatesh** for his encouragement and support.

Abstract

In this work, a deep neural network-based framework consisting of a ”Gated Recurrent Unit (GRU)” decoder and an “EfficientNetV2B0-based Convolutional Neural Network (CNN)” encoder is used to offer a unique method of automatic picture captioning. The framework is designed to perceive information points within images and their contextual relationships, facilitating the generation of meaningful and contextually relevant captions. The CNN encoder built on the EfficientNetV2B0 architecture is very good at identifying objects in pictures and extracting features while preserving spatial information. Next, a language describing the visual information collected in the photographs is created using these qualities. To improve the captioning process, the GRU decoder is essential in word prediction and sentence construction using the retrieved characteristics. The suggested neural network system combines the GRU model with the effectiveness and precision of the EfficientNetV2B0 model as an image feature extractor to provide fixed-length output vectors for ultimate predictions. Popular open-source datasets like Flickr-8k and Flickr-30k are used in the study to train and evaluate the model. Using Python-Keras and TensorFlow backend, the framework is implemented, demonstrating the effectiveness of the GRUbased model and EfficientNetV2B0 in automatic picture captioning tasks. The suggested method for producing correct and contextually appropriate picture captions is shown to be successful and accurate when performance evaluation is carried out using the BLEU (BiLingual Evaluation Understudy) measure.

Synopsis

AMRUTVAHINI COLLEGE OF ENGINEERING,
SANGAMNER

DEPARTMENT OF COMPUTER ENGINEERING

2023-2024

Project Synopsis

on

"Image Captioning system using deep neural network based on encoder-decoder framework"

BE Computer Engineering

BY

Group Id-B04

Mr. Mayur Gadakh (4136)

Mr. Gaurav Chaudhari (4117)

Ms. Akanksha Gaikwad (4141)

Ms. Shivanjali Dhage (4126)

Prof. R. S. Gaikwad
Project Guide
Dept. of Computer Engineering

12/9/23
Dr. D. R. Patil/ Dr. R. G. Tambe
Project Coordinator
Dept. of Computer Engineering

Prof. R. L. Paikrao
H.O.D
Dept. of Computer Engineering

Abbreviation

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
NLP	Natual Langauage Processing
FC	Fully Connected
BLEU	BiLingual Evaluation Understudy
AI	Artificial Intelligence
VGG-19	Visual Geometry Group-19
CPU	Central Processing Units
GPU	Graphical Processing Units
TPU	Tensor Processing Units
RAM	Random Access Memory
NLTK	Natual Language Tool Kit
LTS	Long Term Support
IDE	Integrated Development Envirnoment
SSD	Solid State Drive
VM	Virtaul Machine
METEOR	Metric for Evaluation of Translation with Explicit ORdering
UML	Unified Modeling Language

List of Figures

List of Tables

INDEX

CHAPTER 1

INTRODUCTION

The field of computer vision and natural language processing has undergone a remarkable transformation with the emergence of automatic image captioning. This intricate process involves generating concise and contextually relevant textual descriptions for digital images, posing a challenge in discerning crucial objects, understanding their properties and interrelations, and articulating this understanding in coherent linguistic representation. Bridging this semantic gap requires a harmonious fusion of sophisticated computer vision techniques and robust language models derived from Natural Language Processing (NLP).

Traditionally, the synergy of machine learning and NLP has autonomously deciphered the content within images. Deep neural network strategies, notably Convolutional Neural Networks (CNNs) and advanced Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM), have addressed multifaceted challenges associated with image captioning, aiming to provide articulate natural language explanations that capture the essence of depicted scenes.

This study advances the state-of-the-art by proposing an innovative image captioning framework. At its core, the model leverages the robust feature extraction capabilities of "EfficientNetV2" as the CNN encoder, coupled with the sequential processing prowess of a Gated Recurrent Unit (GRU). Notably, our approach diverges from convention by introducing an attention mechanism tailored for images, enabling the model to selectively focus on specific regions, enriching the contextual depth of generated captions.

While preceding studies have explored diverse methodologies, our proposed model distinguishes itself through the strategic integration of EfficientNetV2 and GRU, coupled with a bespoke attention mechanism. The deliberate choice of the Flickr8k dataset for training and evaluation ensures a focused and cohesive analysis aligned with the intricacies of our model. The conventional encoder-decoder pipeline undergoes a paradigm shift, with EfficientNetV2 pre-trained on the dataset and a GRU serving as a decoder to construct meaningful image descriptions.

In contrast to traditional architectures like CNNs and LSTMs, our approach strategically leverages the unique strengths of EfficientNetV2 and GRU. The project's experimental phase encompasses rigorous training and meticulous testing on the Flickr8k

dataset, underpinned by comprehensive performance evaluations utilizing metrics such as BLEU scores. Furthermore, we draw inspiration from related datasets such as MS COCO, Flickr30k, and pertinent local datasets, ensuring a holistic assessment of the proposed image captioning generator.

A Fully Connected (FC) layer, also known as a Dense layer, is a fundamental building block in neural networks. In this layer, each neuron is connected to every neuron in the previous and next layers. These connections are characterized by weights, which are learned during training, and bias terms. The layer computes a weighted sum of its input, adds the bias, and applies an activation function to produce the output. FC layers are crucial for capturing complex relationships in data and are often used in various neural network architectures for tasks like classification and regression.

In the expansive realm of deep learning for image captioning, the synergistic marriage of EfficientNetV2 and Gated Recurrent Unit (GRU) architecture opens up a frontier of possibilities. EfficientNetV2, renowned for its superior efficiency in image feature extraction, seamlessly intertwines with the sequential processing prowess of GRU, creating a holistic model capable of unraveling the intricate narrative within visual content. As we embark on this journey, the aim is to transcend conventional boundaries, pushing the envelope of image captioning capabilities. This exploration not only represents a convergence of cutting-edge technologies but also holds the promise of fostering a deeper, more nuanced comprehension of visual data, ushering in a new era in the symbiotic relationship between deep learning, computer vision, and natural language understanding.

1. Image Captioning : Image captioning is a computer vision and natural language processing task where a model generates textual descriptions for given images. Utilizing neural networks, particularly encoder-decoder architectures, the model extracts features from the image using an encoder and then decodes these features into a coherent and descriptive sentence. Image captioning combines visual understanding and language generation, making it a multimodal task. This technology finds applications in accessibility tools, aiding visually impaired individuals, and enhances image indexing and retrieval systems. The

task involves teaching models to recognize objects, activities, and relationships within images, contributing to a more comprehensive understanding of visual content.

2. Encoder- decoder Architecture : An encoder-decoder model is a neural network architecture designed for sequence-to-sequence tasks. The encoder processes input sequences and compresses them into a fixed-size context or latent representation. The decoder then takes this representation and generates an output sequence step by step. It is commonly employed in tasks such as machine translation, where the input and output sequences can vary in length. The encoder captures the input's semantic information, enabling the decoder to produce meaningful outputs. This architecture is pivotal in natural language processing and image captioning, providing a structured approach to handle sequential data. It facilitates the transfer of information from input to output while accommodating varying lengths of sequences.

3. EfficientNetV2 : At the heart of our image captioning framework lies the EfficientNetV2, a state-of-the-art Convolutional Neural Network (CNN) architecture. EfficientNetV2 is renowned for its efficiency in terms of both computational resources and model parameters while maintaining superior performance in image classification tasks. Developed as an evolution of the EfficientNet architecture, version 2 incorporates novel techniques such as efficient scaling to balance model depth, width, and resolution. Its robust feature extraction capabilities make it an ideal candidate for the initial stage of our image captioning process, enabling the model to grasp salient features within images efficiently.

4. Gated Recurrent Unit (GRU) : Complementing the CNN encoder, our image captioning framework employs a Gated Recurrent Unit (GRU) as the sequential processing component. GRU is a type of Recurrent Neural Network (RNN) that excels in capturing long-term dependencies in sequential data. Unlike traditional RNNs, GRUs feature gating mechanisms that enhance their ability to retain essential information and discard irrelevant details. This makes

GRUs well-suited for the task of decoding and generating meaningful natural language descriptions of images based on the features extracted by the CNN encoder.

5. Teacher Forcing : Teacher forcing is a training approach in sequence-to-sequence models. During training, the model is fed true output sequences as input. This accelerates learning by guiding the model with correct sequences. In contrast, during inference, the model generates outputs based on its own predictions, potentially leading to exposure bias. Teacher forcing helps in capturing dependencies between input and output sequences. It is widely used in tasks like language translation and image captioning. The method aids faster convergence during training. However, the model might face challenges during testing due to discrepancies between training and inference conditions. To address this, a combination of teacher forcing and other techniques is often employed.

6. Attention Mechanism : A pivotal innovation in our image captioning framework is the incorporation of an attention mechanism tailored for images. Attention mechanisms have proven invaluable in natural language processing tasks, allowing models to focus on specific parts of input sequences when generating corresponding outputs. In the context of image captioning, our attention mechanism enables the model to selectively attend to relevant regions of the input image during the decoding phase. This adaptive focus enriches the contextual depth of the generated captions, ensuring that the model attends to the most pertinent visual features when constructing textual descriptions.

As the project unfolds, substantial contributions to the field are envisaged. By addressing the limitations of existing models and introducing novel components like EfficientNetV2 and GRU with an attention mechanism, our endeavor strives to set new benchmarks for the performance of automatic image captioning technology. This introduction lays the groundwork for a rigorous exploration into the nuances of image comprehension and caption generation, propelling the field towards enhanced capabilities and broader applicability.

1.1 PROJECT IDEA

The project endeavors to create an advanced image captioning system, leveraging cutting-edge deep learning techniques, including EfficientNetV2 and GRU. By combining these state-of-the-art models, the system aims to produce rich and contextually relevant textual descriptions for images. This venture sits at the intersection of computer vision and natural language processing, addressing the complex challenge of bridging visual understanding and linguistic expression. With a focus on automation, the system seeks to enhance accessibility for visually impaired individuals and improve image indexing for efficient retrieval. The utilization of EfficientNetV2 ensures robust feature extraction from images, while the GRU facilitates sequential information processing for fluent caption generation. This project contributes to the evolving landscape of multimodal AI systems, fostering a deeper integration of visual and textual understanding. The exploration of EfficientNetV2 and GRU techniques aligns with the forefront of deep learning, promising innovative strides in the field of image captioning.

1.2 MOTIVATION OF THE PROJECT

To bridge the gap between computer vision and natural language understanding through the development of an advanced image captioning system, this project is driven by the recognition that seamlessly integrating visual and linguistic capabilities holds immense potential for enhancing various applications in artificial intelligence and deep learning. The emphasis on image captioning stems from its status as a highly valuable skill, presenting substantial opportunities for career growth in the dynamically evolving fields of AI and deep learning. The project's motivation extends beyond technical innovation to address the practical implications of creating systems that can interpret and articulate visual content. By advancing image captioning techniques, the project aims to contribute to the broader landscape of AI applications, empowering individuals and industries with more intuitive and comprehensive tools for image analysis and understanding.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

1. **Md. Mijanur Rahman et al.,(2022) :** This study introduces a deep neural network framework for automatic image captioning. It utilizes a Convolutional Neural Network (CNN) encoder to grasp spatial details and recognize objects in images, extracting features for creating a descriptive vocabulary. A Long Short-Term Memory (LSTM) decoder then predicts words and forms coherent sentences. The VGG-19 model serves as an image feature extractor, and the LSTM model processes sequences, producing a fixed-length output vector. The system is trained and tested on various open-source datasets like Flickr 8k, Flickr 30k, and MS COCO using Python, Keras, and TensorFlow. Performance is evaluated using the BLEU metric.[1]

2. **Kavitha P.V et al.,(2022) :** Image captioning is evolving as an interesting area of research that involves generating a caption or describing the content in the image automatically. The idea behind image captioning is to make the computer perceive a given image like a human mind leading to automatic description. Image captioning is a challenging task that involves capturing semantically correct information and expressing in a simple sentence. A large number of methods have been proposed in the recent past, and we aim to do a comprehensive survey in the different deep learning algorithms used in image captioning based on the method framework.[2]

3. **Chitrapriya Ningthoujam et al.,(2022) :** Image captioning is a process of automatically describing an image with one or more natural language sentences. In recent years, image captioning has witnessed rapid progress, from initial template-based models to the current ones, based on deep neural networks. This paper gives an overview of issues and recent image captioning research, with a particular emphasis on models that use the deep encoder-decoder architecture. We discuss the advantages and disadvantages of different approaches, along with reviewing some of the most commonly used evaluation metrics and datasets.[3]

4. **Rashid Khan et al.,(2022) :** This study develops an image captioning system that utilizes a pre-trained CNN to extract image features and integrates them with an attention mechanism. A GRU-based language model is employed to generate descriptive captions. By merging the Bahdanau attention model with GRU, the system focuses on specific image regions, enhancing performance. Experimental results on the MSCOCO dataset show competitive performance against state-of-the-art approaches.[4]

5. **Vaishali Narula et al.,(2021) :** Image captioning is one of the most recent challenges that caught the interest of the computer vision community as well as the Natural Language Processing community. Recently, the tedious task of image captioning has attained quite notable progress by using numerous techniques. The primary goal of this paper is to study existing Deep Learning techniques for Image Captioning. We have discussed a convolutional neural network-based Image Caption generation model and the salient steps involved in it. We have also discussed dataset and evaluation metrics widely used in fundamental systems.[5]

6. **Haoran Wang et al.,(2020) :** This research explores machine learning algorithms for image and natural language processing, integrating existing packages. It implements an algorithm generating comprehensive sentences from images. After requirements analysis, a bibliographic study informed model selection. A composite model, employing transfer learning in a deep convolutional neural network for feature extraction and a recurrent neural network for descriptions, was designed using Keras with TensorFlow. The result is a trained model capable of describing images in natural language.[6]

7. **Xiaoxiao Liu et al.,(2020) :** This article proposes a novel approach to improving image caption generation by integrating spatial visual attention and semantic concepts. By leveraging a semantic attention mechanism and adaptive attention, the model dynamically incorporates high-level semantic information into the caption generation process. This allows for more accurate and descriptive captions by considering both image features and salient semantic

elements. Through experimental validation, the proposed model demonstrates promising performance, producing concise yet rich descriptions of images.[7]

8. **Yurio Windiatmoko et al.,(2020) :** This research focuses on designing a model for local tourism-specific image captioning to support AI-powered assistance systems. Leveraging a visual Attention mechanism and the EfficientNet architecture, the model aims to generate captions representing both literal descriptions and human-like responses. Comparative analysis with other architectures like VGG16 and InceptionV3 demonstrates superior performance, with EfficientNetB0 achieving the best BLEU scores of 73.39 for training and 24.51 for validation. The developed model produces logical captions tailored for local tourism-related images.[8]

9. **Andrej Karpathy et al.,(2015) :** This model generates descriptions of images and their regions by learning from image-sentence datasets to understand the relationship between language and visual data. It combines Convolutional Neural Networks for image regions and bidirectional Recurrent Neural Networks for sentences, aligning them through a multimodal embedding. Using this alignment, a Multimodal Recurrent Neural Network generates descriptions for image regions. Evaluation on Flickr8K, Flickr30K, and MSCOCO datasets demonstrates its superior performance in retrieval tasks compared to baseline methods, both for full images and region-level annotations. [9]

10. **Oriol Vinyals et al.,(2015) :** This paper introduces a deep recurrent model that merges computer vision and machine translation techniques to generate natural language descriptions for images. Through extensive experiments, we demonstrate the model's high accuracy and fluency in producing descriptive sentences solely from image inputs. Our approach outperforms existing methods, achieving significantly higher BLEU scores on multiple datasets, including Pascal, Flickr30k, SBU, and COCO, approaching human-level performance in some cases.[10]

CHAPTER 3

PROBLEM DEFINITION AND SCOPE

3.1 PROBLEM STATEMENT

Implementing Deep Neural Network Based Encoder-Decoder Framework for Image Captioning using EfficientNetV2 as encoder and GRU as decoder.

3.1.1 Goals and objectives

Goal and Objectives:

- To study the deep learning techniques like CNN and RNN.
- To develop a deep-learning based image caption generator that can accurately describe the contents of an image in natural language.
- To create a user-friendly interface for interacting with the image captioning system.

3.1.2 Statement of scope

- Automatic image captioning using deep neural network encoder-decoder frameworks has extensive potential.
- It can enhance accessibility, content indexing, e-commerce, healthcare, education, and more by generating descriptive image captions.
- This technology streamlines processes, improves user experiences, and finds applications across diverse domains.

3.2 SOFTWARE CONTEXT

- The project will involve the development of software components for image captioning using EfficientNetV2 and GRU.
- The software will include modules for image feature extraction, natural language processing, and model integration.
- Additionally, the project will utilize relevant libraries, frameworks, and tools for deep learning, image processing, and natural language generation.

3.3 MAJOR CONSTRAINTS

- **Computational Resources :** Limited computational power and hardware may impact the speed and scalability of the model training and image caption generation processes, especially when dealing with large datasets like Flickr8k.
- **Data Availability :** In Flickr8k dataset the quality and diversity of these datasets may still pose challenges in terms of data preprocessing, management, and ensuring they are suitable for your specific deep learning models.
- **Time Frame :** Meeting project deadlines within the academic semester, while working with multiple datasets and conducting rigorous evaluations, is a crucial constraint, as it may affect the extent of research, development, and testing possible.

3.4 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

3.4.1 Methodologies of Problem Solving

- **Feature Extraction with EfficientNetV2 :** Utilize EfficientNetV2 as a feature extractor to represent images effectively. This methodology involves fine-tuning the pre-trained model's layers and extracting high-level features from the images.
- **Caption Generation with GRU :** Implement a GRU-based sequence-to-sequence model for generating captions. This methodology includes training the model to learn the language structure and generate coherent and contextually relevant descriptions.
- **Data Augmentation :** Apply data augmentation techniques to enhance dataset diversity and reduce overfitting. This involves techniques like image cropping, flipping, and color jittering to improve model generalization.

3.4.2 Efficiency Issues

- **Computational Resources :** Address efficiency issues by optimizing model architecture and training procedures to make the best use of available hardware. This includes considering batch sizes, model parallelization, and hardware acceleration (e.g., GPUs).

- **Memory Management :** Efficiently manage memory during training and inference to handle large datasets and avoid memory bottlenecks. Techniques such as batch loading, memory-efficient data structures, and model quantization can be explored.

- **Real-time Inference :** Optimize the caption generation process for real-time applications by improving model inference speed. This can involve quantization, model compression, and deployment on hardware suitable for real-time processing.

3.5 SCENARIO IN WHICH MULTI-CORE, EMBEDDED AND DISTRIBUTED COMPUTING USED

- **Multi-Core Utilization :** Leveraging multi-core processing is essential for parallelizing image processing tasks, enabling simultaneous feature extraction and caption generation for multiple images. This approach optimizes computational efficiency, reducing the time taken to process large volumes of images and generate captions.

- **Embedded Computing :** Implementing the image captioning system on embedded devices, such as edge computing platforms and IoT devices, allows for on-device processing without relying heavily on external computing resources. This approach ensures the availability of real-time captioning capabilities in applications where network connectivity may be limited or unstable.

- **Distributed Computing :** Utilizing distributed computing allows the system to distribute computational tasks across multiple nodes, optimizing the processing of extensive datasets and complex neural network models. By utilizing

a distributed architecture, the image captioning system can handle large-scale image processing and caption generation, catering to the demands of high-throughput applications.

3.6 OUTCOME

The outcomes of implementing image captioning using EfficientNetV2 and GRU are multifaceted:

- **Enhanced Accessibility :** It makes visual content more accessible to individuals with visual impairments, as it provides descriptions for images, enabling a richer online experience.
- **Automated Tagging :** The technology automates the process of tagging and categorizing images, which can significantly improve content organization and retrieval.
- **Content Recommendation :** It enables more intelligent content recommendation systems by understanding the visual content of images and associating them with user preferences.
- **Improved Human-Computer Interaction :** The ability to generate captions for images enhances human-computer interactions, making it easier for users to interact with and search for visual content.
- **Advancements in AI :** It exemplifies the progress in AI and deep learning, showcasing how neural networks can bridge the gap between visual and textual information.

3.7 APPLICATIONS

- **Social Media :** Image captioning is commonly used on social media platforms to automatically generate captions for user-uploaded images, making content more engaging and informative.

- **Content Recommendation :** Image captions can be used to personalize content recommendations by analyzing the textual descriptions and user preferences, improving user engagement and retention.
- **E-commerce :** Image captioning can provide product descriptions and details for e-commerce websites, enhancing the shopping experience by offering detailed information about products.
- **Education :** Image captioning can be applied in educational materials to provide additional context and information for images in textbooks, online courses, and educational websites.

3.8 HARDWARE RESOURCES REQUIRED

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	2 GHz	Required for efficient processing of image data
2	RAM	8 GB	Necessary for handling the computational load during image captioning
3	GPU	2 GB	Required for fast processing

Table 3.1: Hardware Requirements

3.9 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System : Windows 10 or Ubuntu 20.04 LTS
2. IDE : PyCharm or Jupyter Notebook
3. Programming Language : Python 3.7 or higher, with libraries such as TensorFlow, NLTK, and NumPy.

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

4.1.1 Purpose and Scope of Document

The purpose of this document is to outline the design and implementation of a Deep Neural Network-based Encoder-Decoder framework for Image Captioning. This framework combines Convolutional Neural Networks (CNNs) to encode images and Recurrent Neural Networks (RNNs) to generate descriptive captions.
The scope includes explaining the architecture, training process, and evaluation metrics for image captioning tasks. It provides a comprehensive guide for researchers and developers interested in creating image captioning systems, enabling them to understand the key components, techniques, and considerations involved in this field. The document aims to bridge the gap between theory and practical implementation in the domain of computer vision and natural language processing.

4.1.2 Overview of responsibilities of Developer

The developer's responsibilities in implementing a Deep Neural Network-based Encoder-Decoder framework for Image Captioning include

- **Data Preprocessing :** Collect, clean, and preprocess image and caption data, ensuring it's suitable for training the model.
- **Architecture Design :** Design the neural network architecture, combining CNNs and RNNs, specifying the number of layers, units, and activation functions.
- **Model Implementation :** Code the framework using deep learning libraries (e.g., TensorFlow, PyTorch) and integrate the encoder-decoder structure.
- **Hyperparameter Tuning :** Optimize hyperparameters like learning rates, batch sizes, and sequence lengths for better model performance.
- **Training :** Train the model on the prepared dataset, monitoring loss and performance metrics.

4.2 FUNCTIONAL REQUIREMENTS

4.2.1 System Feature 1 (Image Preprocessing Module)

The system must have a module for preprocessing input images to make them suitable for the image captioning model. The image preprocessing module in image captioning prepares raw images for analysis by resizing, normalizing pixel values, and potentially augmenting data. It enhances feature extraction by ensuring consistent image dimensions and colour ranges, facilitating subsequent deep learning model input. This module improves model performance and consistency in generating accurate image captions.

4.2.2 System Feature 2 (Encoder-Decoder Architecture)

The core of the system is the Encoder-Decoder architecture, where the image is encoded using a Convolutional Neural Network (CNN) and the encoded information is decoded into a caption using a Recurrent Neural Network (RNN).The Encoder-Decoder architecture is a fundamental framework used in image captioning. The encoder processes the input image, typically using convolutional neural networks (CNNs) to extract image features. The decoder, often using recurrent neural networks (RNNs) or transformer models, generates a textual caption based on the extracted features. This approach combines computer vision and natural language processing to produce coherent and contextually relevant image captions.
This feature includes:

- **Image Encoding :** The system implements the EfficientNetV2 model to effectively encode images, enabling comprehensive feature extraction and representation from input images for subsequent processing in the caption generation process.
- **Caption Generation :** The system utilizes the GRU (Gated Recurrent Unit) model for generating natural language captions from the encoded image information. This process involves leveraging the GRU's sequential processing capability to produce coherent and contextually relevant captions for the provided images.

- **Connection Between Encoder and Decoder :** The system establishes an efficient connection between the image encoder, implemented with EfficientNetV2, and the caption decoder, implemented with GRU. This integration ensures smooth information flow from the encoded image representation to the caption generation module, facilitating a seamless and accurate translation of visual features into descriptive captions.

4.2.3 System Feature 3 (Model Training and Fine-Tuning)

This feature involves the training and fine-tuning of the neural network model to ensure it can effectively generate accurate and contextually relevant image captions. It includes the following components:

- **Training Data :** The system employs diverse dataset, including Flickr8k to train the neural network model. This comprehensive training data helps the model gain a robust understanding of various visual contexts and linguistic patterns, enabling it to generate accurate and contextually relevant captions.
- **Loss function :** The system incorporates a customized loss function tailored to the specific requirements of the EfficientNetV2 and GRU-based architecture. This specialized loss function optimizes the model's training process, ensuring efficient fine-tuning and improved caption generation performance.

These components collectively contribute to the effective training and fine-tuning of the neural network model, enhancing its capability to generate accurate and contextually relevant image captions.

4.3 EXTERNAL INTERFACE REQUIREMENTS (IF ANY)

4.3.1 User Interfaces

- **Image Upload/Selection :** Allow users to upload or select images for captioning. You can include an option for capturing images from a camera.
- **Caption Display :** Display the generated image captions in a clear and readable format.

- **Caption Customization :** Provide options for users to customize the generated captions, such as adjusting length, style, or language.
- **Caption Generation Button :** Include a button or action to trigger the caption generation process.
- **Image Preview :** Show a preview of the uploaded image(s) to confirm the selection.
- **Accessibility Features :** Make the interface accessible to users with disabilities, including alt text for images and keyboard navigation.
- **Error Handling :** Implement error messages and guidance for users in case of issues with image processing or caption generation.
- **Privacy and Security :** Clearly communicate how user data and images are handled and stored, addressing privacy and security concerns.

4.3.2 Hardware Interfaces

- **Input Devices :** For non-real-time or batch processing, users can upload images from various sources, including: Computer storage (hard drive, SSD) External storage devices (USB drives, SD cards)
- **Processing Unit (CPU/GPU) :** The image captioning model requires significant computational power for image analysis and caption generation. A CPU and/or GPU is used for this purpose. Modern deep learning models benefit greatly from GPUs due to their parallel processing capabilities.
- **Memory (RAM) :** Sufficient RAM is essential for loading and processing images, especially when dealing with large datasets or multiple concurrent users.
- **Storage :** The system should have storage capacity for storing images, model parameters, and generated captions. SSDs or large-capacity hard drives are common choices.

- **Network Interface :** A network connection is necessary for accessing image databases, model updates, and potentially sharing or storing captioned images online.
- **Display :** A monitor or screen is needed to display the user interface and the captioned images. It can be a desktop monitor, laptop screen, or the screen of a mobile device.
- **Peripheral Devices :** Input devices such as keyboards, mice, and touchscreens are necessary for user interaction with the system.

4.3.3 Software Interfaces

- **Image Input :** Allow users to upload or provide images for captioning. You can use file upload widgets or device cameras (for mobile apps).
- **Image Processing :** Preprocess the input image. This may involve resizing, normalizing, and enhancing the image to improve model performance.
- **Caption Generation :** Pass the image features through the NLP model to generate captions for the image. These captions can be single sentences or multiple sentences depending on the complexity of the image.

4.4 NONFUNCTIONAL REQUIREMENTS

4.4.1 Performance Requirements

Performance requirements for an image captioning project using deep learning are critical to ensure that the system meets user expectations and operates efficiently. Here are key performance requirements to consider:

- **Accuracy :** The image captioning model should provide accurate and meaningful captions for a wide range of images. Define a minimum accuracy threshold, such as BLEU, METEOR, or CIDEr scores, to evaluate model performance.

- **Speed :** Define the expected response time for generating captions. Users generally expect quick results, so set a maximum response time to meet user experience expectations.
- **Scalability :** Ensure that the system can handle an increasing number of users and images. Define performance requirements for both horizontal (adding more servers) and vertical (scaling a single server) scalability.
- **Latency :** Define maximum acceptable response times for generating captions for different image sizes. Ensure low-latency interactions with the user interface.

4.4.2 Safety Requirements

- **Privacy Protection :** Ensure that user data and uploaded images are protected and not shared with unauthorized parties. Comply with data protection regulations such as GDPR or HIPAA, as applicable.
- **Transparency and Explainability :** Make the image captioning process as transparent and explainable as possible. Users should understand how captions are generated, and there should be a way to provide explanations for generated captions when requested.

4.4.3 Security Requirements

- **Data Encryption :** Implement data encryption in transit (using HTTPS) and at rest to protect image data and captions from unauthorized access or interception.
- **Secure Data Storage :** Ensure that user data and uploaded images are securely stored with appropriate access controls to prevent unauthorized access.

4.4.4 Software Quality Attributes

- **Accuracy :** The accuracy of generated captions is crucial. Captions should closely match the content of the images.

- **Performance :** The system should generate captions quickly, especially for real-time or interactive applications.
- **Scalability :** The ability to handle a growing number of users and images without significant degradation in performance is essential.
- **Usability :** The user interface should be intuitive and user-friendly, making it easy for users to interact with the system.
- **Security :** User data and generated captions should be protected from unauthorized access and breaches.
- **Maintainability :** The codebase should be well-structured, documented, and easy to maintain, allowing for updates and improvements.
- **Reliability :** The system should be available and responsive, with minimal downtime.

4.5 SYSTEM REQUIREMENTS

4.5.1 Database Requirements

To implement image captioning, we select a dataset flickr8k in which 8k images and 8k captions included.

4.5.2 Software Resources Required

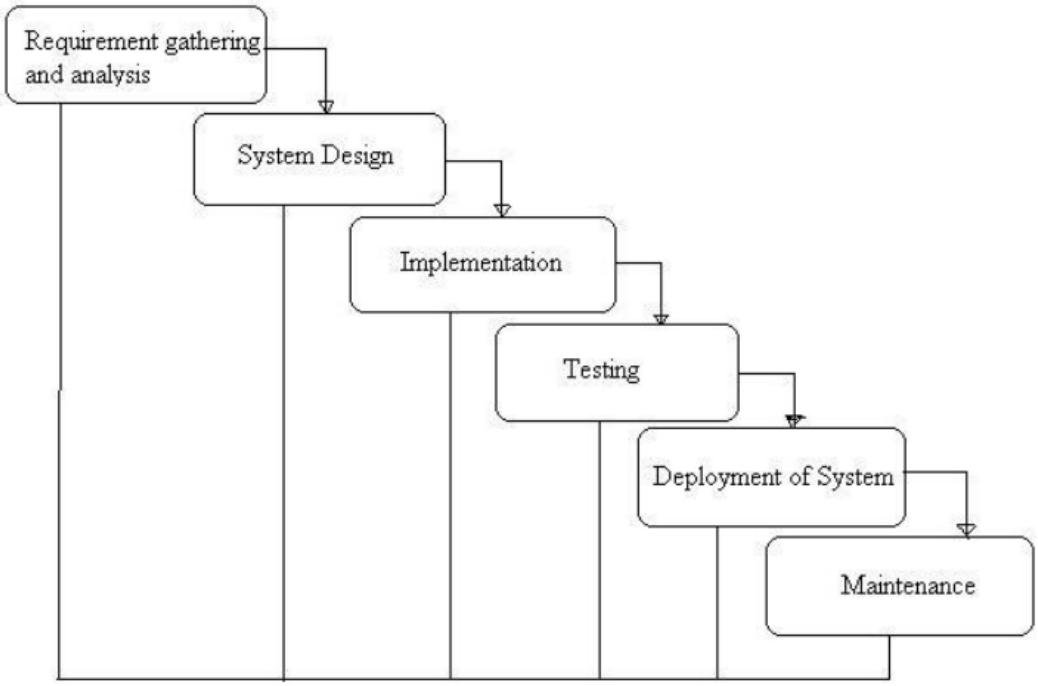
Platform :

1. Operating System : Windows 10 or Ubuntu 20.04 LTS
2. IDE : PyCharm or Jupyter Notebook
3. Programming Language : Python 3.7 or higher, with libraries such as TensorFlow, NLTK, and NumPy.

4.5.3 Hardware Resources Required

1. CPU : Speed 2 GHz Required for efficient processing of image data.
2. RAM : 8 GB Necessary for handling the computational load during image captioning.
3. GPU : Minimum 2 GB dedicated GPU is required for less training & testing time.

4.6 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED



The Waterfall Model is a traditional software development methodology that follows a linear and sequential approach, consisting of distinct phases such as requirements, design, implementation, testing, and maintenance. While the Waterfall Model is more commonly associated with traditional software development, it is not typically used for deep learning tasks like image captioning, which involve a more iterative and experimental process. Deep learning models are often developed using iterative approaches, where the model is trained, evaluated, and refined in multiple cycles.

The application of the waterfall model to image captioning using deep learning, specifically with EfficientNetV2 and Gated Recurrent Unit (GRU), provides a structured and sequential approach. The cascade of processes, from feature extraction with EfficientNetV2 to context modeling with GRU, exemplifies a systematic flow in the development lifecycle. This methodology ensures a step-by-step refinement, allowing for a thorough exploration of the interplay between visual and textual information. As we conclude this endeavor, the waterfall model's rigidity aligns with the meticulous integration of EfficientNetV2 and GRU, contributing to a comprehensive and well-defined framework for advancing image captioning capabilities within the realm of deep learning.

The process can be described as follows:

- **Requirements :** Define the requirements for the image captioning system, including input data specifications, desired captioning output, and performance metrics.
- **Design :** Design the architecture of the deep learning model, including specifying the type of neural network (e.g., CNN-RNN), deciding on model parameters, and planning data preprocessing steps.
- **Implementation :** Develop and implement the deep learning model according to the designed architecture using a framework like TensorFlow or PyTorch.
- **Testing :** Evaluate the model on a validation dataset to assess its performance. This involves measuring metrics such as accuracy, BLEU scores, or other evaluation criteria relevant to image captioning.
- **Deployment :** If the model meets the desired performance criteria, deploy it for real-world use. This may involve integrating the model into an application or system capable of accepting input images and generating captions.
- **Maintenance :** Periodically update and fine-tune the model based on new data or changing requirements. Maintenance may also involve addressing issues discovered during real-world usage.

CHAPTER 5

METHODOLOGY AND SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

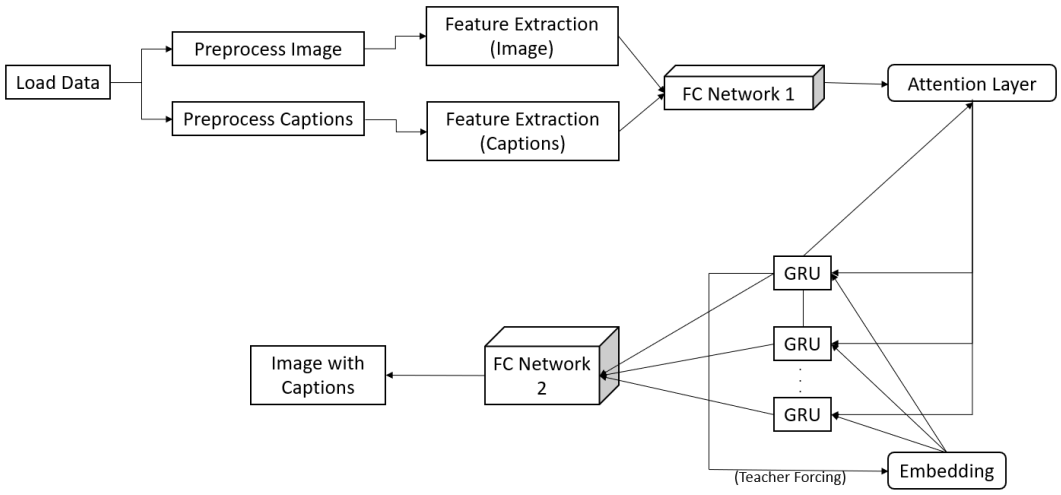


Figure 5.1: System Architecture

The image caption generation system begins with the pivotal step of loading data, encompassing a curated dataset of images and their corresponding captions. Subsequently, meticulous data preparation unfolds, involving resizing images, normalizing pixel values, and tokenizing captions for streamlined processing. The heart of the architecture lies in feature extraction, where Convolutional Neural Networks (CNNs) like EfficientNetV2 are employed to distill high-level visual features from the images. These extracted features serve as inputs for the image caption generation model, typically implemented using GRU. During the training phase, the model refines its parameters by predicting the next word in a sequence, learning the intricate associations between visual context and generated captions. The training data, consisting of image-caption pairs, fuels this iterative learning process through optimization algorithms like stochastic gradient descent. The integrated system encompasses components for user input, caption generation, and potential post-processing, culminating in a deployment-ready architecture that can generate descriptive captions for input images in real-world scenarios. This holistic architecture, combining EfficientNetV2, GRU, teacher forcing, and attention mechanisms, forms a sophisticated system for image captioning. It not only leverages the efficiency of EfficientNetV2 in visual feature extraction but also benefits from the sequential modeling capabilities of GRU, enhanced by attention mechanisms.

5.2 DATA FLOW DIAGRAMS

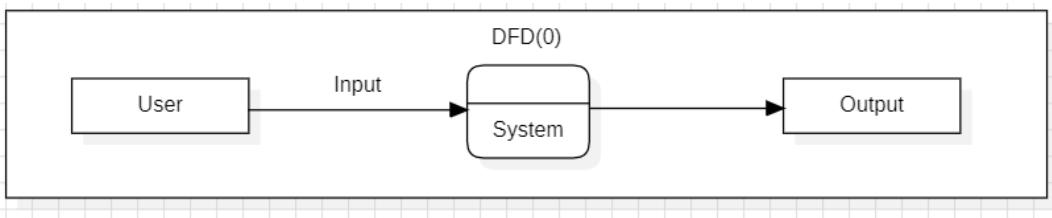


Figure 5.2: DFD0

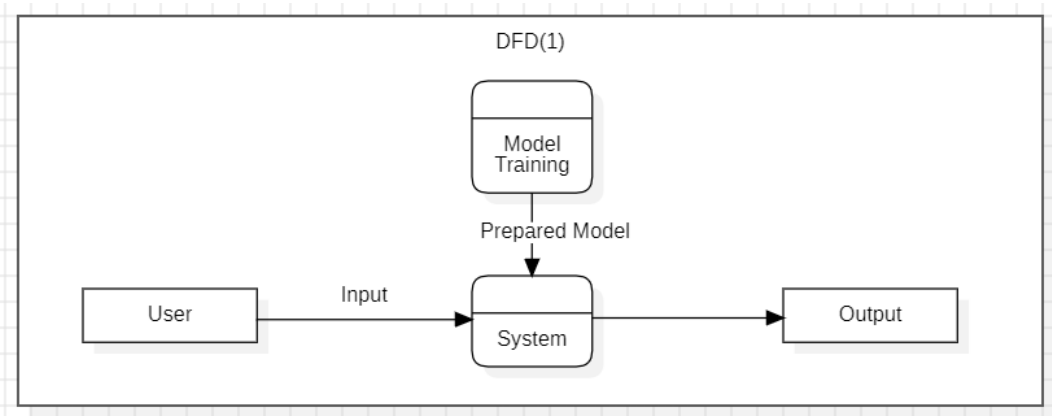


Figure 5.3: DFD1

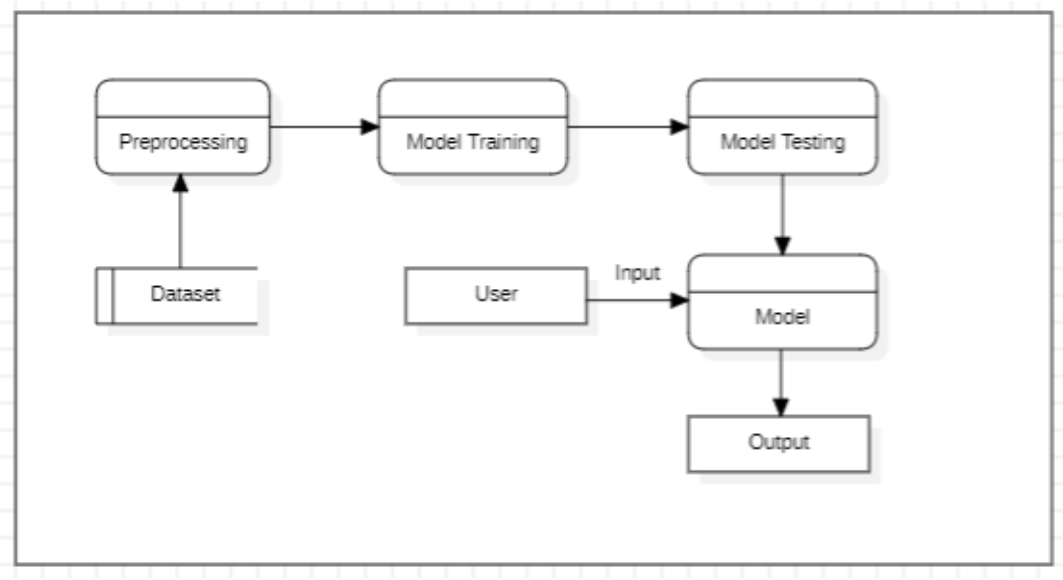


Figure 5.4: DFD2

A Data Flow Diagram (DFD) visually represents the flow of data within a system.

- External Entities : User: Initiates the image captioning process by uploading an image.
- Upload and Preprocess Image : Accepts the uploaded image from the user. Performs data preprocessing on the image (resizing, normalization, etc.).Outputs the preprocessed image data.
- Feature Extraction : Takes the preprocessed image data as input.Utilizes a feature extraction model (e.g., EfficientNetV2) to extract relevant features.Outputs the extracted features.
- Generate Caption : Takes the extracted features as input.Utilizes a caption generation model (e.g., GRU with attention) to generate captions.
- Display Image and Caption : Presents the original image and the generated caption to the user.
- Image Database : Stores information about images, including ImageID, FilePath, Timestamp.
- Caption Database : Stores generated captions, including CaptionID, CaptionText, Timestamp.This DFD provides an overview of the processes involved in image captioning, including data inputs, transformations, and outputs.

The system involves a user initiating image captioning by uploading an image, which undergoes preprocessing and is fed into a feature extraction model (e.g., EfficientNetV2). Extracted features are then used in a caption generation model (e.g., GRU with attention) to produce captions. The original image and generated caption are displayed to the user. Information about images is stored in an Image Database, and generated captions are stored in a Caption Database, both with timestamps for reference and organization. This integrated process provides a comprehensive framework for image captioning, encompassing user interaction, data processing, feature extraction, caption generation, and result presentation.

5.3 ENTITY RELATIONSHIP DIAGRAMS

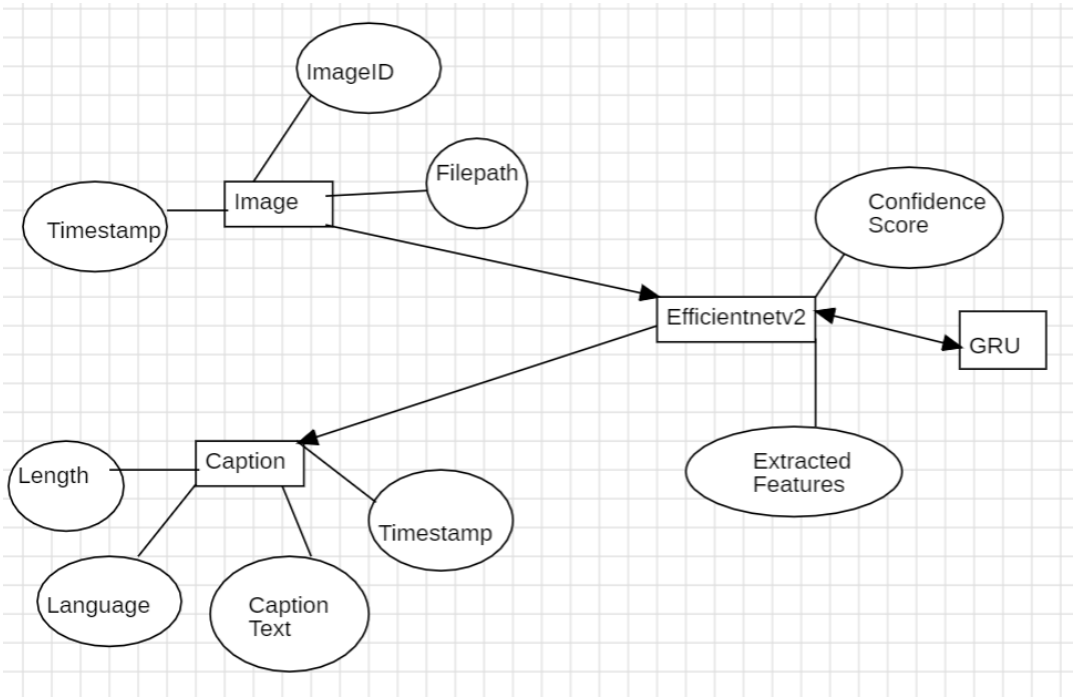


Figure 5.5: Entity Relationship Diagrams

An Entity-Relationship (ER) diagram typically represents the entities and relationships in a database. In image captioning with deep learning, the entities include caption, Image, EfficientNetV2 and GRU. The Entity-Relationship Diagram (ERD) illustrates the interconnected entities in the image captioning system. The “Image” entity encompasses attributes such as ImageID, FilePath, and Timestamp, forming a one-to-many relationship with the “Caption” entity, which consists of CaptionID, CaptionText, and Timestamp. Both entities are associated with the “EfficientNetV2” and “GRU” entities, representing many-to-one relationships, as multiple images share the same feature extraction and caption generation models. The “EfficientNetV2” entity includes ModelID, Architecture, and Parameters, while the “GRU” entity encompasses ModelID, Architecture, Attention, and Parameters. This structured ERD encapsulates the integral components, relationships, and attributes within the image captioning system, providing a comprehensive overview of its architecture and data flow.

5.4 UML DIAGRAMS

5.4.1 Use Case Diagram

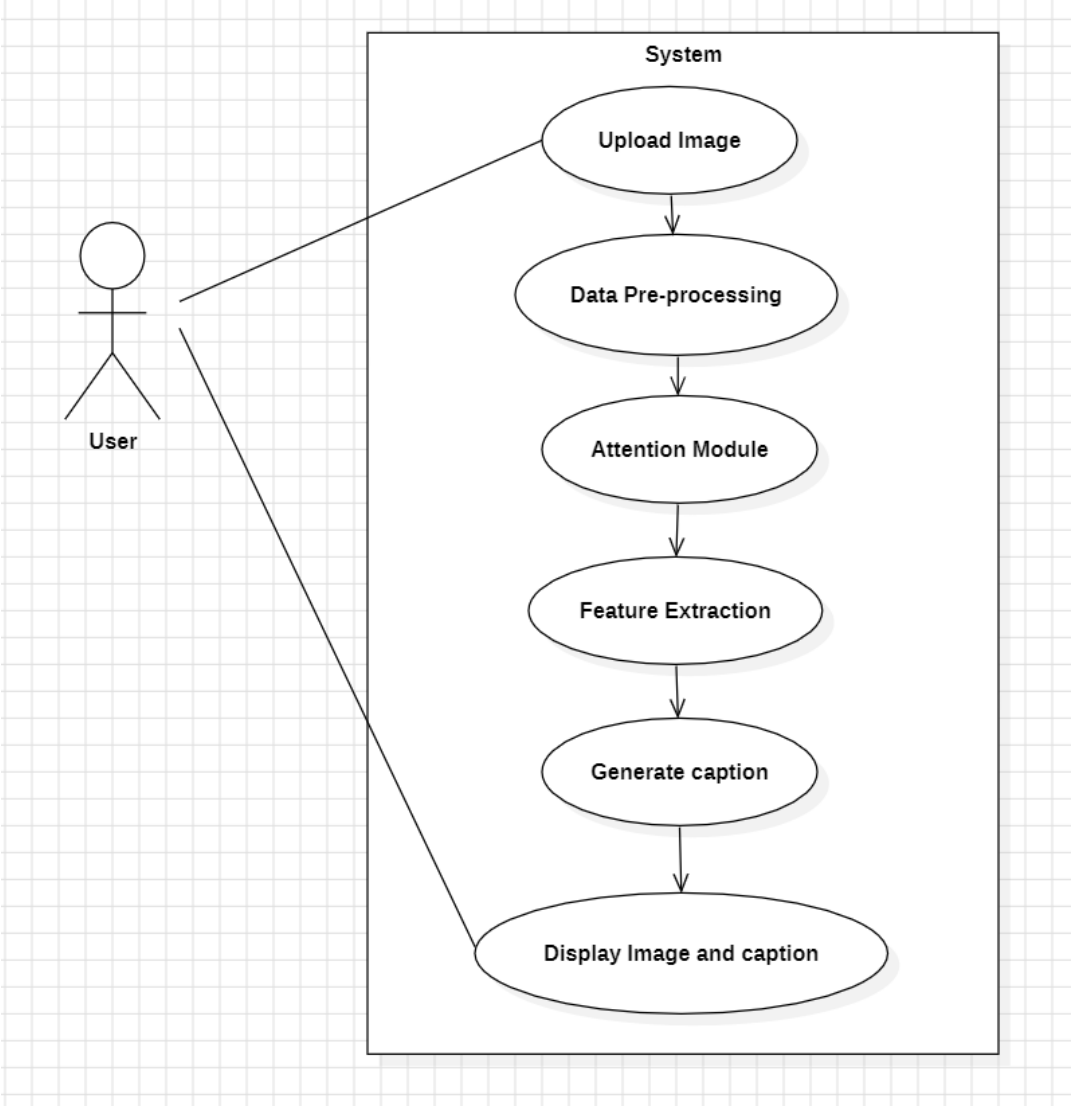


Figure 5.6: Use Case Diagram

A use case diagram for image captioning provides a high-level view of the system's functionalities and interactions with external actors. The actor represent entities that interact with the system, and use cases represent specific functionalities or features provided by the system. Here's a simplified use case diagram for an image captioning system.[11]

5.4.2 Activity Diagram

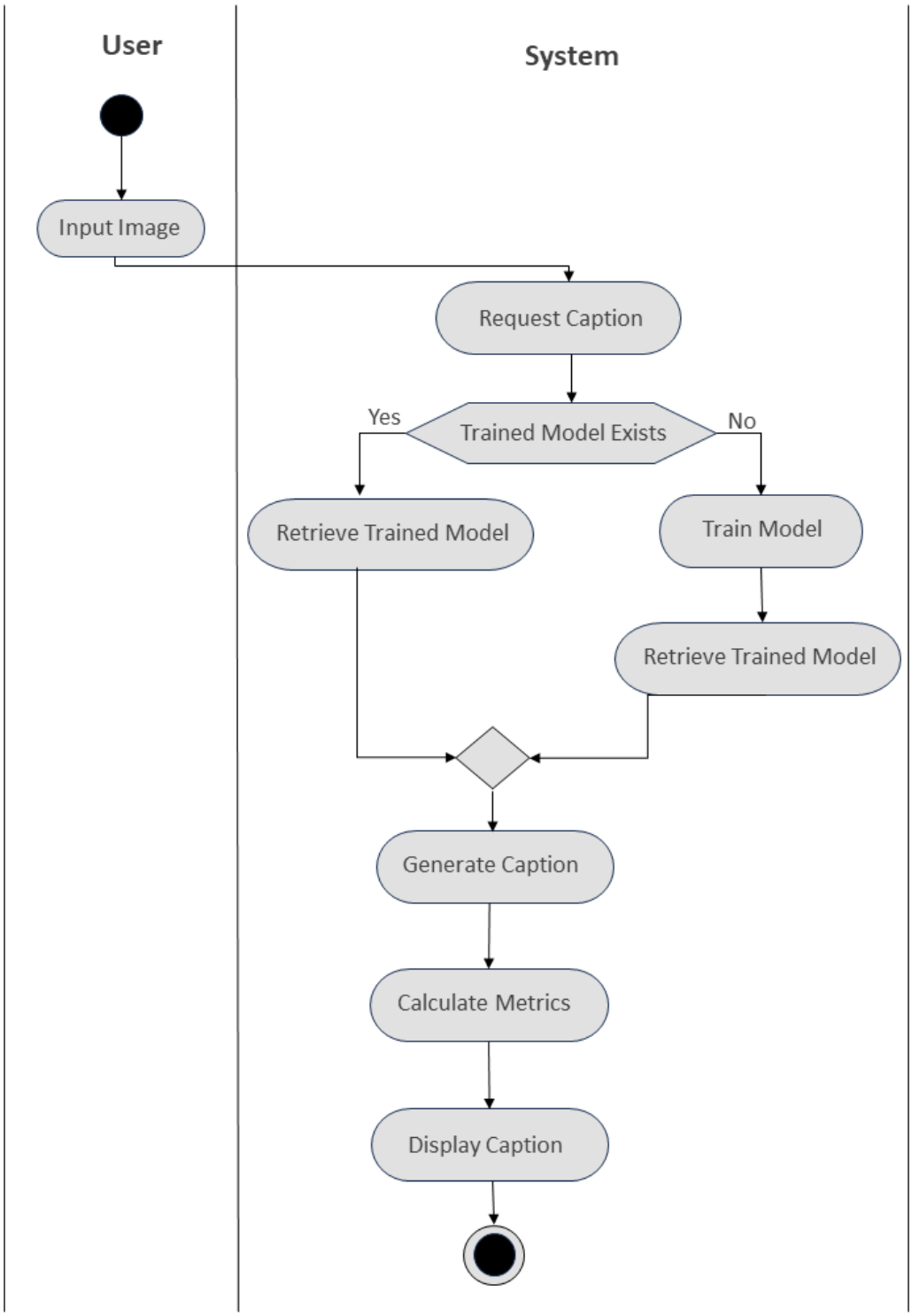


Figure 5.7: Activity Diagram

- An activity diagram for image captioning can illustrate the flow of activities involved in the process.
- “Image User” is an actor representing the user interacting with the system.
- “Upload Image” is a use case where the user uploads an image for captioning.
- “Image Captioning System” is a use case representing the overall image captioning process.
- “Receive Caption and Display” is a use case where the system receives the generated caption and displays it to the user.
- “Generate Caption for Image” is a use case responsible for actually generating the caption for the uploaded image.
- The “Image User” actor interacts with the system through “Upload Image” and receives the caption through “Receive Caption and Display.”
- This diagram illustrates the high-level interactions and functionalities involved in the image captioning The “Image User” actor starts by uploading an image.
- The “Captioning Requester” initiates the request for a caption.
- The “Image Captioning System” generates a caption for the uploaded image.
- The “Metrics Calculation System” calculates metrics for the generated caption.
- Arrows indicate the flow of activities, and each box represents an activity or a system component.
- This diagram provides a visual representation of the sequential steps involved in the image captioning process, including uploading an image, requesting a caption, generating the caption, and calculating metrics for evaluation.

5.4.3 Sequence Diagram

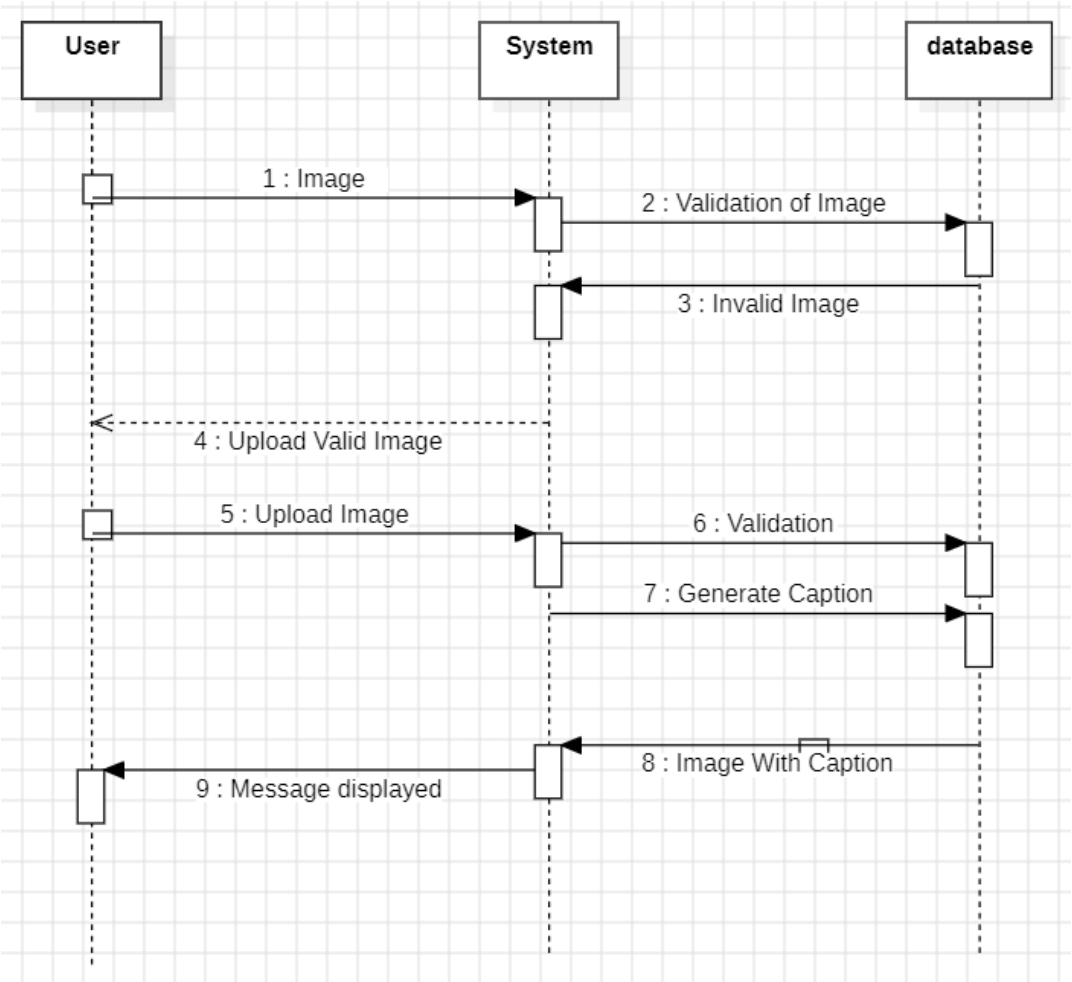


Figure 5.8: Sequence Diagram

- A sequence diagram for image captioning can help illustrate the interactions between different components or objects in a chronological order.
- In this sequence diagram:
- The “Image User” uploads an image.
- The “Image Validation” component validates the uploaded image.
- If the image is valid, it is uploaded to the “Image Captioning System.”
- The “Image Captioning System” generates a caption for the image.
- The final result is an “Image with Caption.”

5.4.4 Class Diagram

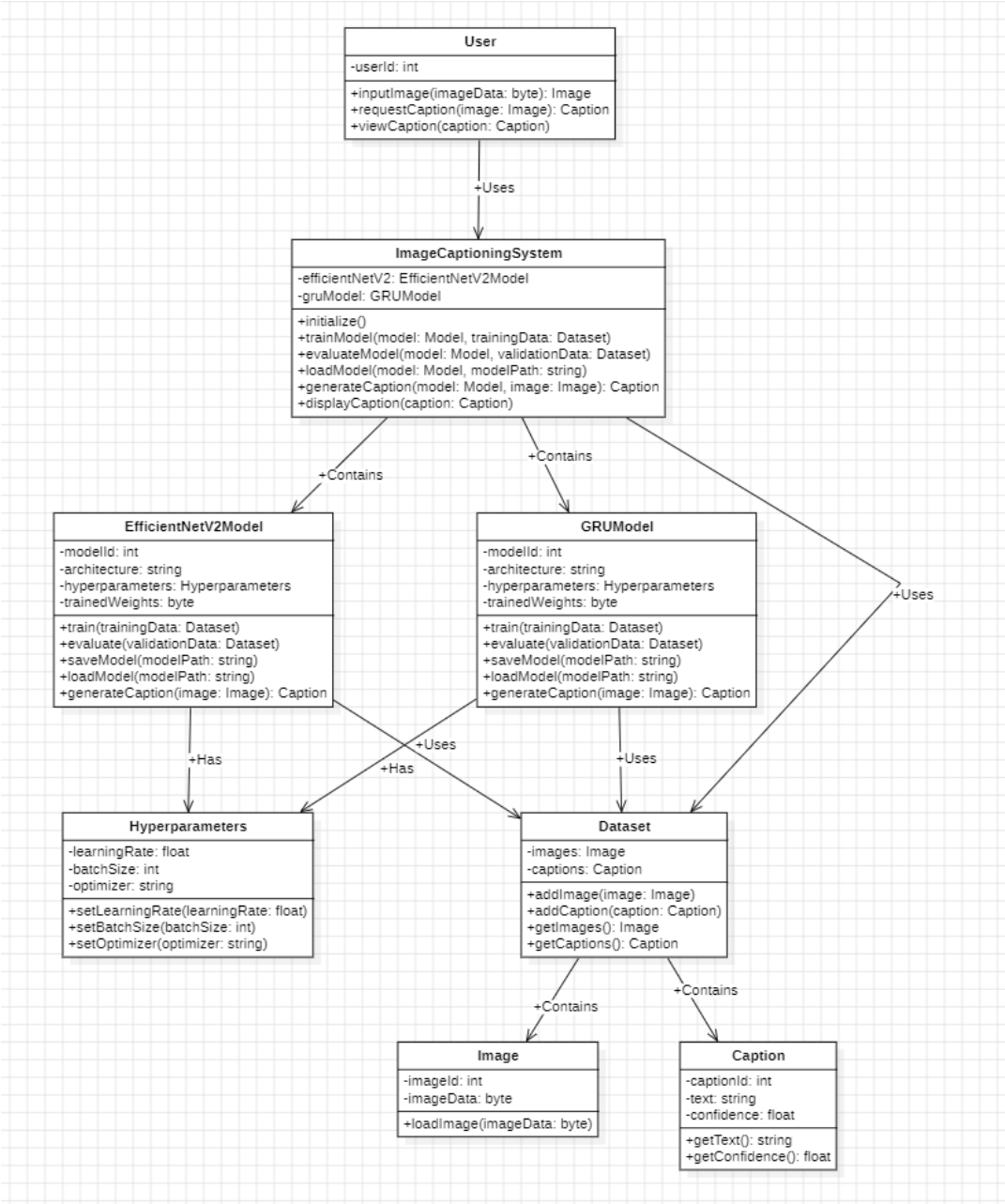


Figure 5.9: Class Diagram

In a class diagram for image captioning with deep learning, you can represent classes, their attributes, and methods. The central class in this system is the ImageCaptioningSystem, which encapsulates the entire image captioning functionality. It is associated with two main components: the ImageProcessor and the CaptionGenerator. The ImageProcessor class is responsible for handling image-related operations.[9]

5.4.5 Object Diagram

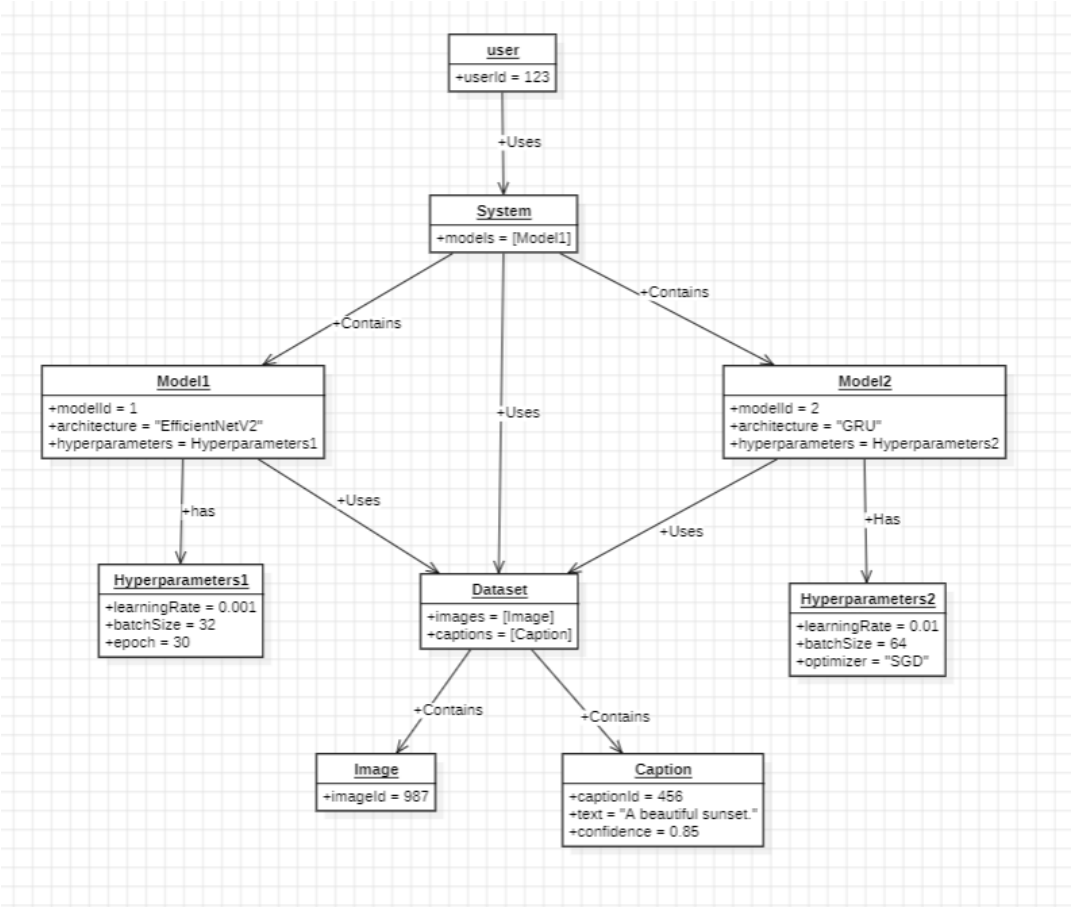


Figure 5.10: Object Diagram

An object diagram illustrates instances of classes and their relationships at a specific point in time. In the context of image captioning with deep learning, we have instances of the classes mentioned in the class diagram. imageCaptionSystemInstance contains a composition relationship with imageProcessorInstance, indicating that the ImageProcessor is a vital part of the ImageCaptioningSystem. It also has an association with captionGeneratorInstance. imageProcessorInstance is associated with imageInstance, showcasing that it processes this particular image. The instances of various classes represent the specific components involved in image captioning. The relationships and associations illustrate how these instances collaborate within the system[9]

CHAPTER 6

SOFTWARE IMPLEMENTATION

6.1 TECHNOLOGY DETAILS USED IN THE PROJECT

6.1.1 TensorFlow :

- **Definition :** TensorFlow is an open-source machine learning framework developed by Google Brain Team. It's designed to facilitate the creation and deployment of machine learning models across a variety of platforms, from desktops to mobile devices to large-scale distributed systems.

- **Key Features :**

  Flexibility : TensorFlow offers both high-level APIs (such as Keras) for ease of use and low-level APIs for greater flexibility and control over model architecture and training process.

  Scalability : It allows efficient execution of computations across different hardware accelerators, including CPUs, GPUs, and TPUs (Tensor Processing Units).

  Extensive Ecosystem : TensorFlow has a rich ecosystem of tools, libraries, and community resources, making it easy to build, train, and deploy machine learning models for various tasks.

- **Use Cases :** TensorFlow is widely used in various domains, including image classification, natural language processing, object detection, recommendation systems, and more.

6.1.2 Keras :

- **Definition :** Keras is a high-level neural networks API written in Python. It's designed to be user-friendly, modular, and extensible, allowing developers to quickly build and experiment with different deep learning architectures.

- **Key Features :**

  User-Friendly Interface : Keras offers a simple and intuitive interface for designing neural networks, making it suitable for both beginners and experienced researchers.

  Integration with TensorFlow: Keras is tightly integrated with TensorFlow as

its default backend, enabling seamless interoperability and compatibility with TensorFlow ecosystem.

Modularity : Keras allows building complex neural network architectures by stacking layers in a modular way, facilitating rapid prototyping and experimentation.

Use Cases : Keras is used for a wide range of deep learning tasks, including image classification, object detection, text generation, sentiment analysis, and more.

6.1.3 GRU (Gated Recurrent Unit):

- **Definition :** Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture, primarily used for sequence modeling tasks such as text generation, machine translation, and speech recognition.

- **Key Features :**

  Efficient Architecture : GRU is similar to the more complex LSTM (Long Short-Term Memory) cell but with fewer parameters, making it computationally more efficient.

  Gating Mechanism : It utilizes gating mechanisms to control the flow of information within the network, allowing it to capture long-range dependencies in sequential data.

  Effective for Sequence Modeling : GRU is effective for tasks involving sequential data, where preserving temporal dependencies is crucial.

  Use Cases : GRU is commonly used in various natural language processing tasks, including language modeling, sentiment analysis, named entity recognition, and more.

6.1.4 EfficientNetV2 :

- **Definition :** EfficientNetV2 is an improved version of the EfficientNet architecture, which is known for its efficiency and effectiveness in image classification tasks.

- **Key Features :**

  Compound Scaling : EfficientNetV2 incorporates advancements in architecture design, such as compound scaling, to achieve better performance with fewer parameters.

  Efficiency : It achieves state-of-the-art performance on image classification benchmarks while being computationally efficient, making it suitable for resource-constrained environments.

  Scalability : EfficientNetV2 scales the depth, width, and resolution of the network in a balanced way, optimizing performance across different input sizes.

  Use Cases : EfficientNetV2 is primarily used for image classification tasks but can also be adapted for other computer vision tasks such as object detection, image segmentation, and image captioning.

  By leveraging the capabilities of TensorFlow, Keras, GRU, and EfficientNetV2, we can develop sophisticated machine learning models for various tasks, including image captioning, with efficiency and effectiveness.

6.1.5 Streamlit :

- **Definition :** Streamlit is an open-source Python library that allows developers to create interactive web applications for machine learning and data science projects.

- **Key Features :**

  Simple and Intuitive : Streamlit provides a simple and intuitive interface for building web applications without requiring knowledge of web development.

  Fast Iteration : Rapid development and iteration of web applications with automatic reloading, allowing developers to see changes instantly. Integration with Python Libraries : Seamless integration with popular Python libraries such as Pandas, Matplotlib, and TensorFlow for building data-driven applications.

  Use Cases: Creating interactive dashboards, data visualization tools, machine learning model demos, etc.

6.2 DATASET USED IN THE PROJECT

6.2.1 Flickr8k :

- **Description :** The Flickr8k dataset consists of 8,000 images collected from the Flickr image-sharing platform. Each image is associated with five captions, providing diverse descriptions of the content.
- **Size :** 8,000 images with five captions each, totaling 40,000 captions.
- **Usage** : Flickr8k is commonly used for benchmarking image captioning models due to its moderate size and diverse range of images and captions.

6.2.2 Flickr30k :

- **Description :** The Flickr30k dataset is an extension of Flickr8k, containing 30,000 images with five captions each. It covers a wider variety of scenes, objects, and activities compared to Flickr8k.
- **Size :** 30,000 images with five captions each, totaling 150,000 captions.
- **Usage :** Flickr30k is often used for training and evaluating image captioning models, especially those aiming for better generalization and robustness.

These datasets are commonly used for training, validation, and testing image captioning models in the context of deep learning frameworks like TensorFlow and architectures like GRU and EfficientNetV2. Researchers and practitioners often fine-tune pre-trained models on these datasets or train models from scratch to generate accurate and meaningful captions for images.

CHAPTER 7

PROJECT ESTIMATION, SCHEDULE AND TEAM STRUCTURE

7.0.1 COCOMO Model

One of the most popular COCOMO models in business is that developed by Bohm [Boehm 8]. This model's initial iteration was released in 1981, and the COCOMO II, which is currently offered in the COCOMO'81, was developed after 63 software projects were reviewed that year. Boehm established the basic, intermediate, and detailed standard levels. A static model known as the simple COCOMO'81 model determines software development effort (and cost) based on program performance, which is represented as an estimated number line (LOC). The software development effort is calculated using the average model COCOMO'81 based on the size and procedure of the "cost driver," which comprises the product, repair facilities, personnel, and project features.

Every aspect of the intermediate version is available in the comprehensive COCOMO'81 model, which also assesses the cost factor's influence at each stage of the software engineering process (analysis, design, and other). Out of two equal points for the COCOMO'81 model: First, there is the development effort (MM stands for man-month, person-month, or person-work-month, which is one person's effort each month). In COCOMO'81, each person has 152 hours every month. These findings could not match standard 10 depending on the company.

MM=aKDSIb followed by Time to Effort and Development (TDEV) TDEV=cMMd KDSI represents the number of thousands of instructions sent and size. The coefficients a, b, c, and d depend on the evolution. There are three types of development. Here are the coefficients related to development modes for an intermediate model.

In the intermediate power model, the equation becomes: MM = aKDSIbC

Development Mode	Innovation	Deadline	Development Environment
Organic	little	not light	stable
Semi-detached	medium	medium	medium
Embedded large	greater	tight	complex hardware

Table 7.1: Modes of development

C is the estimated power value, it is easily calculated by the equation of the

Development Mode	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded large	2.8	1.2	2.5	0.32

Table 7.2: Modes of development

driver value , so the average model is more accurate than the basic model.

The steps to estimate using the average COCOMO'81 model are:

Identify the new project's growth pattern (organic, semi-independent, and embedded).

- Estimate the size of the item in KDSI to get a nominal estimate.

- Set 15 price drivers to influence your project. "" Calculate the effort plan using the original equation and effort setting (C).

- Use the second equation to calculate the project length. The model follows the Construction Cost Model (COCOMO), which is used to estimate the effort required to complete the project. Like all forecast models, the COCOMO model requires large datasets.

These data can be listed in the following files:

1. Object point
2. Functional Content (FP)
3. Sentences

We use big data in the form of Lines of Location Code for our project.

1. All lines of code for our project, KLOC = 6000 (approx).
2. Price per person per month, Cp = Rs. 7.415 /- (ib lub his)

7.0.2 Equation

$$E = a \times (KLOC)^b$$

Where,

$$a = 3.2, \quad b = 1.05 \quad \text{for organic program.}$$

$$E = \text{Person-months workforce}$$

$$D = a \times (E)^b$$

7.0.3 Organic projects

For medium-sized and complex projects, teams with mixed experience levels must meet strict and loose requirements (material half of the embedded types and organic types). Number of people: Using the COCOMO model, the formula for calculating the number of people needed to complete a project for is:
N = E / D
Where,

N	People needed
E	personal effort - months
D	project duration (Months)

Project Cost: The equation for calculating the project cost using the CO-COMO model:

$$C = D \times C_p$$

Here,

C	Project Cost
D	Duration in Months
C_p	Months per person Received Efforts: $E = 3.2 \times (6)^{1.05}$

$$E = 20.99 \text{ man-months.}$$

It will take a total of 20.99 man-months to complete this task.

Project duration:

$$D = 2.5 \times (M)^{0.32}$$

$$D = 6 \text{ months}$$

The project duration is approximately 6 months.

Number of people needed for the project:

$$N = \frac{20.99}{6}$$

$$N = 3.6$$

$$N = 4 \text{ people}$$

Therefore, 4 people are needed to complete the project smoothly and on time.

Cost:

$$C = 4.00 \times 5000 = 20,000$$

So the product costs Rs. 20,000 /- (approx.)

7.1 PROJECT SCHEDULE AND TEAM STRUCTURE

Developers ID	Developers Name
T1	Topic finalization
T2	Requirement specification
T3	Technology finalization
T4	System setup
T5	Concept review study
T7	Data collection and preprocessing
T8	Model architecture design
T9	Training and optimization
T10	Evaluation metrics selection
T11	Fine-tuning and model validation
T12	Documentation and reporting

Table 7.3: List of Tasks

Task No.	No. of days	Developers
T1	7	D1,D2,D3,D4
T2	4	D1,D2,D3,D4
T3	4	D1,D2,D3,D4
T4	2	D1,D2,D3,D4
T5	4	D1,D2,D3,D4
T6	7	D1,D2,D3,D4
T7	8	D1,D3,D4
T8	5	D1,D2,D3,D4
T9	5	D1,D2,D3,D4
T10	7	D1,D2,D3,D4
T11	5	D2,D3,D4
T12	10	D1,D2,D3,D4

Table 7.4: Task Organization

Developers ID	Developers Name
D1	Mr. Mayur Gadakh
D2	Mr. Gaurav Chaudhari
D3	Ms. Akanksha Gaikwad
D4	Ms. Shivanjali Dhage

Table 7.5: List of Developers

7.2 SYSTEM IMPLEMENTATION PLAN :

Sr.No.	Activity	Plan Start	Plan Duration (weeks)	Aug	Sep	Oct	Nov	Dec	Jan	Feb
1	Literature Survey	1	2							
2	Identify Objectives	14	2							
3	Feasibility study	1	2							
4	Study of Scope	15	2							
5	Requirement Analysis	1	2							
6	System Architecture	16	1							
7	UML Diagram	25	2							
8	Implementation and Testing	15	8							
9	Conclusion and Report	19	2							

Figure 7.1: System Implementation

CHAPTER 8

SOFTWARE TESTING AND VALIDATION

8.1 SOFTWARE TESTING AND VALIDATION

- **Importance of Testing :**

  Validation: Testing ensures that the image captioning system generates accurate and meaningful descriptions for a variety of images.

  Reliability : Identifies bugs and performance issues, ensuring the system works reliably across different scenarios and datasets.

  User Satisfaction : Helps in refining the system to meet user expectations and improve overall satisfaction.

- **Types of Testing :**

  Unit Testing : Involves testing individual components or modules, such as the CNN and RNN models, to ensure each part functions correctly in isolation.

  Integration Testing : Focuses on the interaction between different components (e.g., how the features extracted by the CNN are processed by the RNN).

  System Testing : Validates the complete system by testing it end-to-end, from image input to caption output.

  Acceptance Testing : Ensures the system meets the requirements and performs well in real-world scenarios, often involving user feedback.

- **Test Cases :**

  Definition : A test case is a set of conditions or variables under which a tester will determine if the system under test satisfies requirements or works correctly.

  Components : Each test case includes an ID, description, expected output, actual output, and result (pass/fail).

- **Performance Metrics :**

  BLEU Score : Used to evaluate the quality of the generated captions by comparing them to reference captions.

  Precision and Recall : Measures how many relevant captions are correctly generated (precision) and how many relevant captions are retrieved from the total available (recall).

Execution Time : Measures how quickly the system generates captions, indicating efficiency.

- **Test Data :**

  Training Data : Images and their corresponding captions used to train the model.

  Validation Data : Used to tune model parameters and ensure the model generalizes well to unseen data.

  Test Data : A separate set of images and captions used exclusively for testing the final model to evaluate its performance.

- **Evaluation Techniques :**

  Manual Evaluation : Involves human evaluators assessing the relevance and accuracy of generated captions.

  Automated Evaluation : Uses metrics like BLEU scores to automatically assess the quality of captions.

- **Challenges in Testing : Subjectivity :** Evaluating the quality of captions can be subjective, as different evaluators might have varying opinions on what constitutes a good caption.

  Complexity of Scenes : Images with complex scenes and multiple objects can be difficult for the system to caption accurately.

  Bias and Fairness : Ensuring the model does not perpetuate biases present in the training data and performs fairly across diverse image sets.

- **Continuous Testing :**

  Iterative Improvement : Regular testing and feedback loops help in continuously improving the system.

  Automated Testing Pipelines : Implementing automated testing pipelines ensures consistent and efficient testing processes.

8.1.1 Test Cases :

Testcase ID	Testcase Description	Expected Output	Actual Output	Result
TC01	Upload an image to the system	Image successfully uploaded	Image successfully uploaded	Pass
TC02	Generate caption for a clear image with one object	Accurate caption describing the object in the image	Accurate caption describing the object in the image	Pass
TC03	Generate caption for a complex image with multiple objects	Accurate caption describing multiple objects	Accurate caption describing multiple objects	Pass
TC04	Handle an empty or corrupted image file	Error message indicating invalid image	Error message indicating invalid image	Pass
TC05	Generate caption for an image with poor lighting	Caption generated with reduced accuracy due to poor lighting	Caption generated with reduced accuracy due to poor lighting	Pass
TC06	Generate caption for an image with occluded objects	Caption indicating possible occlusions or partial visibility	Caption indicating possible occlusions or partial visibility	Pass
TC07	Measure BLEU score for a generated caption	BLEU score within acceptable range	BLEU score within acceptable range	Pass
TC08	Process a batch of images for captioning	Captions generated for all images in the batch	Captions generated for all images in the batch	Pass

Table 8.1: Test Cases

8.2 RISK MANAGEMENT:

implementing an image captioning system using EfficientNetV2 and GRU involves several risks that need to be managed effectively. Here's a risk management plan tailored to this scenario:

- **Model Performance Risk:** There is a risk that the performance of the image captioning model may not meet the desired accuracy or quality standards. This could be due to limitations in the model architecture, insufficient training data, or suboptimal hyperparameters.
- **Data Quality Risk:** Poor quality or biased training data can adversely affect the performance and generalization of the model. Biases in the training data could lead to biased or inaccurate captions, especially for underrepresented groups or uncommon scenarios.
- **Computational Resource Risk:** Training and deploying deep learning models, especially large ones like EfficientNetV2, require significant computational resources in terms of processing power, memory, and storage. Inadequate resources may lead to long training times, scalability issues, or operational challenges.
- **Deployment and Integration Risk:** Integrating the image captioning system into existing software or deploying it in production environments may introduce compatibility issues, dependencies, or disruptions to existing workflows.
- **Security Risks:** Deep learning models are vulnerable to various security threats such as adversarial attacks, model inversion attacks, and data poisoning attacks. A compromised model could generate misleading or malicious captions.

CHAPTER 9

RESULT AND ANALYSIS

9.1 OUTCOMES

The proposed approach focuses on providing image captions that describe the pictures. A few anticipated results of deep learning-driven image description generator are shown in the figure, which depicts the user interfaces of the proposed system. It was observed that images featuring people or other human subjects achieved the highest accuracy, as most training images are individual photos. Local images sourced from local camera shots, university websites, and other platforms were also utilized to evaluate the suggested model, which yielded satisfactory results in captioning the images

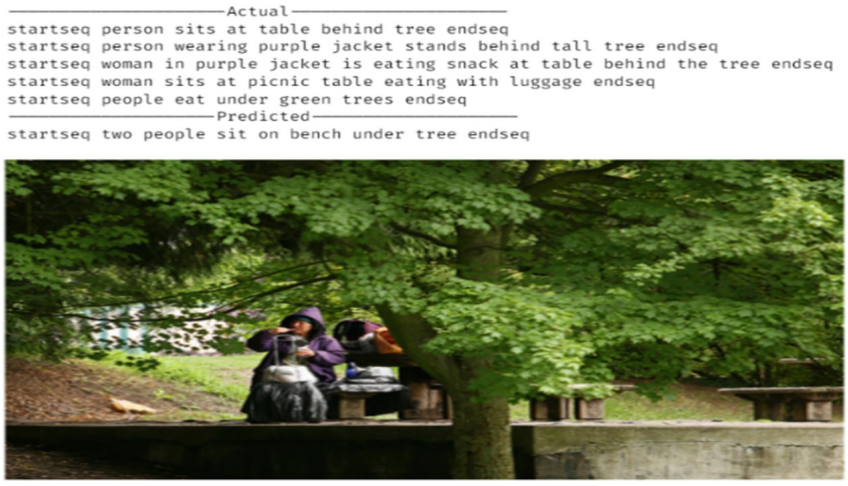


Figure 9.1: Actual and predicted captions for image from the dataset

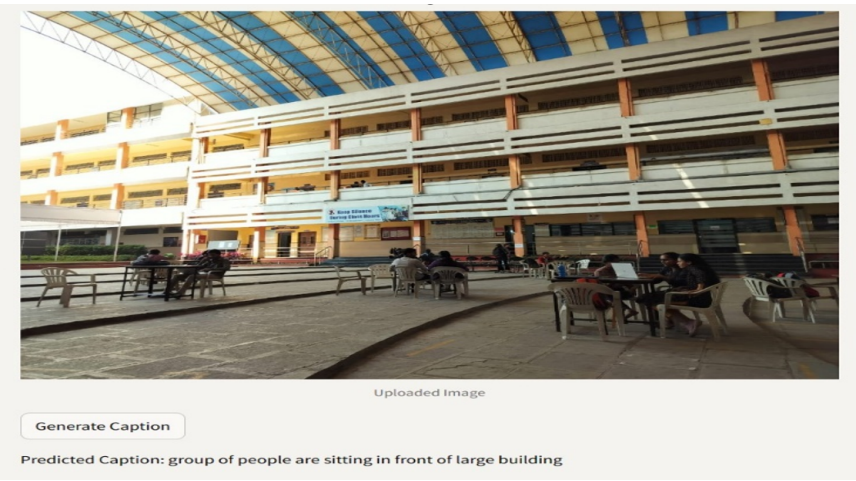


Figure 9.2: Predicted caption for custom image

9.2 PERFORMANCE EVALUATION

BLEU Metric	Flickr8k	Flickr30k
BLEU-1	0.625841	0.598135

Table 9.1: Performance evaluation

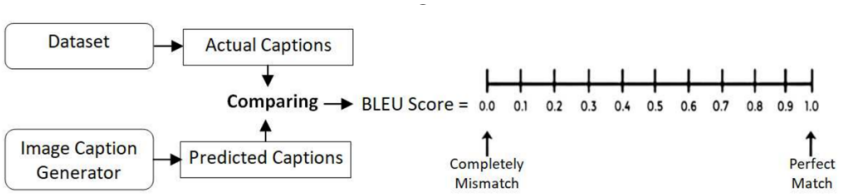


Figure 9.3: The process of generating BLEU score

The test set is used to predict picture captions, and these predictions are then evaluated using an established metric to evaluate the proposed model. BLEU ('Bilingual-Evaluation-Understudy'), a bilingual assessment metric, was used to gauge how effective the proposed photo captioning method was to calculate BLEU score, a predicted sentence is compared to a reference sentence. The corresponding BLEU score prediction and examples of relevant captions are displayed. The Python NLTK module was used to produce the BLEU score for the evaluation of the candidate text. The quality of machine-generated versions is evaluated using the BLEU(Bilingual-Evaluation-Understudy) score. Phrase BLEU score for each sentence and Corpus BLEU Rating for groups of sentences are the two levels at which it works. Comparing comparable grammes in a preset order-for example, one gramme for single words and two grammes for word pairs-determines the N-gram scores. Every N-gram matche receives a weight, usually 0 for non-matches and one for matches. Balanced geometric averages of N-gram scores over a range of orders-from 1 to n-are computed to determine the BLUE score.The collective N-gram values (BLEU-N), which are produced by calculating the weighted geometrical average of the individaul N-gram scores, are a crucial facotr in determining the overall BLEU score.

CHAPTER 10

ADVANTAGES, LIMITATIONS AND APPLICATION

10.1 ADVANTAGES

- **Automated Description :** Deep learning models can automatically generate descriptive captions for images, reducing the need for manual annotation and providing textual information that can be used for various purposes.
- **Accessibility :** Image captions can make visual content more accessible to individuals with visual impairments by providing descriptions of the content in a textual format.
- **Content Retrieval :** Image captions can improve content retrieval in image databases or search engines. Users can search for images using text-based queries, making it easier to find specific images or content.
- **Personalized Content :** Image captions can be tailored to the preferences or needs of the viewer. This personalization can enhance the user experience, especially in content recommendation systems.

10.2 LIMITATIONS

- **Accuracy and Quality :** Deep learning models for image captioning are not always perfect in generating accurate and high-quality captions. They can make mistakes, misinterpret images, or produce captions that do not accurately describe the content.
- **Overfitting :** Deep learning models may overfit to the training data, which means they perform well on the training data but struggle with new or diverse images, leading to incorrect or irrelevant captions.
- **Ambiguity Handling :** Images can be inherently ambiguous, and it can be challenging for deep learning models to handle ambiguity in image content and provide contextually appropriate captions.
- **Lack of Common Sense Understanding :** Deep learning models often lack common sense reasoning abilities, which can lead to captions that make factual errors or provide implausible interpretations of images.

10.3 APPLICATIONS

- **Social Media :** Image captioning is commonly used on social media platforms to automatically generate captions for user-uploaded images, making content more engaging and informative.
- **Content Recommendation :** Image captions can be used to personalize content recommendations by analyzing the textual descriptions and user preferences, improving user engagement and retention.
- **E-commerce :** Image captioning can provide product descriptions and details for e-commerce websites, enhancing the shopping experience by offering detailed information about products.
- **Education :** Image captioning can be applied in educational materials to provide additional context and information for images in textbooks, online courses, and educational websites.

CHAPTER 11

SUMMARY AND CONCLUSION

11.1 SUMMARY

Image captioning with EfficientNetV2 and a GRU is an advanced deep learning technique that enables automatic generation of descriptive text for images. EfficientNetV2, a convolutional neural network, extracts meaningful visual features from the input images. These features are then processed by a GRU, a type of recurrent neural network, to produce coherent and contextually relevant captions. This technology has a wide range of applications, including enhancing image accessibility for the visually impaired, automating image tagging, and improving content recommendation systems. It showcases the power of deep learning in bridging the gap between visual and textual information, ultimately facilitating more effective human-computer interactions and information retrieval.

11.2 CONCLUSION

In conclusion, image captioning using EfficientNetV2 and GRU represents a significant advancement in the field of deep learning. It demonstrates the ability of neural networks to comprehend and describe visual content with textual precision. The technology's applications in accessibility, image organization, and content recommendation underline its practical importance. As it continues to evolve, it has the potential to revolutionize how we interact with and understand the vast amount of visual data in today's digital world. Image captioning with EfficientNetV2 and GRU is a promising example of how artificial intelligence can enhance our relationship with visual media.

CHAPTER 12

REFERENCES

[1] Md. Mijanur Rahman, Ashik Uzzaman, Sadia Islam Sami, and Fatema Khatun. Developing a deep neural network-based encoder-decoder framework in automatic image captioning systems. In *IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)*, 09 2022.

[2] Kavitha P.V and Karpagam Vilvanathan. A comprehensive review on automatic image captioning using deep learning. In *Disruptive Technologies for Big Data and Cloud Applications: Proceedings of ICBDCC*, pages 167–175, 08 2022.

[3] Chitrapriya Ningthoujam and Tejbanta Singh Chingtham. *Comprehensive Comparative Study on Several Image Captioning Techniques Based on Deep Learning Algorithm*, pages 229–240. 01 2022.

[4] M. Shujah Islam Khadija Kanwal Mansoor Iqbal Md Imran Hossain Khan, Rashid and Zhongfu Ye. A deep neural framework for image caption generation using gru-based attention mechanism. 2022.

[5] Mohammad Shahnawaz Alam, Vaishali Narula, Ruchika Haldia, and Gitanjali Nikam Ganpatrao. An empirical study of image captioning using deep learning. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1039–1044, 2021.

[6] Haoran Wang, Yue Zhang, and Xiaosheng Yu. An overview of image caption generation methods. *Computational Intelligence and Neuroscience*, pages 1–13, 01 2020.

[7] Xiaoxiao Liu and Qingyang Xu. Adaptive attention-based high-level semantic introduction for image caption. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16:1–22, 12 2020.

[8] Dhomas Fudholi, Yurio Windiatmoko, Nurdi Afrianto, Prastyo Susanto, Magfirah Suyuti, Ahmad Fathan Hidayatullah, and Ridho Rahmadi. Image captioning with attention for smart local tourism using efficientnet. 09 2020.

[9] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

[10] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[11] A.A.A. Jilanic, Aamer Nadeem, Tai-hoon Kim, and Eun-suk Cho. Formal representations of the data flow diagram: A survey. pages 153 – 158, 01 2009.

ANNEXURE A

AWARDS/PARTICIPATION IN PROJECT COMPETITION/EXHIBITION

A.1 AMRUTEXPO, ORGANIZED BY AMRUTVAHINI COLLEGE OF ENGINEERING(AVCOE), SANGAMNER

We are excited to announce our active participation in the esteemed Amrut Expo, organized by the esteemed Amrutvahini College of Engineering in Sangamner. This exhibition, held on January 19th and 20th, 2024, provided us with a valuable platform to showcase our project and engage with fellow participants, industry experts, and academic leaders. The experience of participating in such a prestigious event was incredibly enriching. We received positive feedback from both judges and attendees, which encourages us to continue our journey of learning and innovation. This opportunity has left a lasting impact on our team, inspiring us to pursue excellence in our future endeavors.

A.2 INTERNARIONAL CONFERENCE, ORGANIZED BY AMRUTVAHINI COLLEGE OF ENGINEERING(AVCOE), SANGAMNER

We are delighted to announce that our research paper titled 'Best Paper Award at ICRTACT 2024' has been honored at the recent international conference on 'Recent Trends and Advancements in Computing Technologies,' held at Amrutvahini College of Engineering, Sangamner, on April 25th, 2024. Our paper, focused on 'Image Captioning Using Deep Learning with GRU and EfficientNetV2,' introduced a novel approach to image captioning. By integrating GRU and EfficientNetV2 architectures, we achieved superior performance, leveraging the strengths of both models. The recognition of our paper with the Best Paper Award underscores the significance and impact of our research in the field. We are honored to receive this prestigious award, and it motivates us to continue our efforts in advancing computing technologies.

ANNEXURE B

DETAILS OF THE PAPERS PUBLICATION (IF ANY)

B.1 PAPER PUBLICATION IN UGC CARE JOURNAL

Dr. R. S. Gaikwad, Mr. Mayur Gadakh, Mr. Gaurav Chaudhari, Ms. Akanksha Gaikwad, Ms. Shivanjali Dhage. "Image Captioning System using Deep Neural Network based on Encoder-Decoder Framework", *International Conference on Recent Trends And Advancements In Computing Technologies* (ICRTACT), AVCOE Sangamner, April 2024.

ANNEXURE C

PLAGIARISM REPORT FOR THIS REPORT

Image Captioning System using Deep Neural Network based on Encoder-Decoder Framework.

by AVCOE Central Library

Submission date: 27-May-2024 04:40AM (UTC-0400)
Submission ID: 2264595903
File name: A19_final_report_2.pdf (2.81M)
Word count: 12733
Character count: 76633

Image Captioning System using Deep Neural Network based on Encoder-Decoder Framework.

ORIGINALITY REPORT

18%	15%	6%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	4%
2	Submitted to Savitribai Phule Pune University Student Paper	2%
3	Submitted to SUNY Polytechnic Institute Student Paper	2%
4	Md. Mijanur Rahman, Ashik Uzzaman, Sadia Islam Sami. "Implementing Deep Neural Network Based Encoder-Decoder Framework for Image Captioning", 2021 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON), 2021 Publication	1%
5	docshare.tips Internet Source	1%
6	pdfcoffee.com Internet Source	1%
	core.ac.uk	