Visvesvaraya Technological University

BELAGAVI, KARNATAKA

ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ

ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

Mini Project report on

"Image Captioning Using A.I"

Submitted by

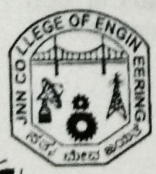
Aakash S Ganiger	**4JN22IS004**
Adarsh B K	**4JN22IS009**
Adithya S S	**4JN22IS011**
Akash A Navale	**4JN22IS016**

Under the guidance of

Mrs. Rashmi R B.E, M.Tech

Associate Professor,

Dept. of IS&E, JNNCE,
Shivamogga.



Department of Information Science & Engineering
JNN College of Engineering
Shivamogga-577204

2024-25

National Education Society ®

JAWAHARLAL NEHRU NEW COLLEGE OF ENGINEERING

SHIVAMOGGA - 577204

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Mini Project entitled

"Image Captioning Using A.I"

Submitted by

Aakash S Ganiger	**4JN22IS004**
Adarsh B K	**4JN22IS009**
Adithya S S	**4JN22IS011**
Akash A Navale	**4JN22IS016**

Students of 5th semester B.E ISE, in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Information Science and Engineering under Visvesvaraya Technological University, Belagavi during the year 2024-25.

Signature of Guide

30/12/24

Mrs. Rashmi R B.E, M.Tech
Associate Professor,
Dept. of IS&E,
JNNCE, Shivamogga

Signature of HOD

30/12/24

Dr. Raghavendra R J B.E, M.Sc., PhD
Associate Professor & Head,
Dept. of IS&E,
JNNCE, Shivamogga

ABSTRACT

Image captioning bridges the gap between visual and textual understanding by generating meaningful descriptions for images. This project presents a machine learning-based approach for automated image captioning, employing a Convolutional Neural Network (CNN) for feature extraction and a Long Short-Term Memory (LSTM) network for caption generation. The system is designed to provide accurate and contextually relevant captions for diverse images, enhancing usability across accessibility, content creation, and multimedia management.

The backbone of the project is the CNN-LSTM architecture, where the CNN extracts key features from images, and the LSTM generates textual descriptions. The dataset used for training includes diverse images paired with descriptive captions to ensure high accuracy and contextual relevance. An attention mechanism is incorporated to dynamically focus on important regions of the image during caption generation. To make the solution accessible, a Gradio-based interface is implemented, enabling users to upload images and receive real-time captions seamlessly.

This project showcases the potential of integrating deep learning techniques for automating visual content description. By delivering accurate and user-friendly solutions, it reduces dependency on manual intervention and supports applications in accessibility, content management, and AI-driven content generation. The system lays the foundation for future advancements in image captioning, contributing to innovative AI solutions across various domains

ACKNOWLEDGEMENT

We feel great to express our humble feeling of thanks to all those who have helped us directly or indirectly in the successful completion of this work.

We would like to express our special thanks to **Dr. Y. Vijaya Kumar**, Principal, JNN College of Engineering, Shivamogga, for providing an opportunity to carry out this work.

We also express our sincere gratitude to **Dr. Raghvendra R. J.**, Associate Professor and Head, Department of Information Science & Engineering, for providing adequate facilities, ways, and means by which we were able to complete this work.

We are deeply indebted to the coordinators, **Mr. Pavan M., Mrs. G. V. Sowmya,** and **Mr. Akshay M. J.,** Assistant Professors in the Department of Information Science & Engineering, for their valuable advice and guidance during the course of mini-project.

We also express our sincere gratitude to our guide **Mrs. Rashmi R,** Associate Professor of Information Science & Engineering, for providing adequate facilities, ways, and means by which we were able to complete this work.

We would like to extend our heartfelt gratitude to the teaching and non-teaching staff of the Department of Information Science & Engineering for their constructive support and cooperation at every juncture of the work.

Finally, we would also like to express our gratitude to **J.N.N College of Engineering, Shivamogga,** for providing all the required facilities, without which the completion of this work would not have been possible.

Aakash S Ganiger	**4JN22IS004**
Adarsh B K	**4JN22IS009**
Adithya S S	**4JN22IS011**
Akash A Navale	**4JN22IS016**

Table of Contents

Table of Figures

Chapter 1

Introduction

Image Captioning using Artificial Intelligence (A.I) is a rapidly evolving field that bridges the gap between computer vision and natural language processing. The primary goal of this project is to generate descriptive textual captions for images automatically, enabling machines to interpret and describe visual content in human-readable language. This capability has numerous applications, including assisting visually impaired individuals, enhancing content management systems, and improving image search engines.

At its core, image captioning involves two main tasks: analyzing the visual elements of an image and generating a coherent textual description. This process requires a combination of deep learning models, typically convolutional neural networks (CNNs) for feature extraction from images and recurrent neural networks (RNNs) or transformers for sentence generation. The integration of these components allows the system to understand the context and relationships within an image and articulate them in a structured manner.

The project also explores the use of advanced datasets like MS COCO (Microsoft Common Objects in Context) to train the AI model. These datasets provide a rich source of labeled images with detailed captions, enabling the model to learn a wide variety of visual concepts and linguistic patterns. Evaluation metrics such as BLEU (Bilingual Evaluation Understudy) and CIDEr (Consensus-based Image Description Evaluation) are commonly employed to measure the quality and relevance of generated captions against human annotations.

This project also highlights the importance of leveraging attention mechanisms to improve the contextual relevance of generated captions, making the system more robust for real-world applications. Additionally, integrating the model with interactive interfaces, like Gradio, ensures ease of use for diverse audiences. By enhancing caption accuracy and accessibility, this work has the potential to pave the way for future advancements in human-computer interaction. Through this project, we aim to develop a robust image captioning system that can process diverse image inputs and produce accurate, context-aware descriptions. The outcome of this work has the potential to contribute significantly to fields such as accessibility, automated content generation, and multimedia analysis, showcasing the power of A.I.

Chapter 1

Introduction

Image Captioning using Artificial Intelligence (A.I) is a rapidly evolving field that bridges the gap between computer vision and natural language processing. The primary goal of this project is to generate descriptive textual captions for images automatically, enabling machines to interpret and describe visual content in human-readable language. This capability has numerous applications, including assisting visually impaired individuals, enhancing content management systems, and improving image search engines.

At its core, image captioning involves two main tasks: analyzing the visual elements of an image and generating a coherent textual description. This process requires a combination of deep learning models, typically convolutional neural networks (CNNs) for feature extraction from images and recurrent neural networks (RNNs) or transformers for sentence generation. The integration of these components allows the system to understand the context and relationships within an image and articulate them in a structured manner.

The project also explores the use of advanced datasets like MS COCO (Microsoft Common Objects in Context) to train the AI model. These datasets provide a rich source of labeled images with detailed captions, enabling the model to learn a wide variety of visual concepts and linguistic patterns. Evaluation metrics such as BLEU (Bilingual Evaluation Understudy) and CIDEr (Consensus-based Image Description Evaluation) are commonly employed to measure the quality and relevance of generated captions against human annotations.

This project also highlights the importance of leveraging attention mechanisms to improve the contextual relevance of generated captions, making the system more robust for real-world applications. Additionally, integrating the model with interactive interfaces, like Gradio, ensures ease of use for diverse audiences. By enhancing caption accuracy and accessibility, this work has the potential to pave the way for future advancements in human-computer interaction. Through this project, we aim to develop a robust image captioning system that can process diverse image inputs and produce accurate, context-aware descriptions. The outcome of this work has the potential to contribute significantly to fields such as accessibility, automated content generation, and multimedia analysis, showcasing the power of A.I.

1.1 Problem Statement:

"Design and implement an automated system that takes an image as input and generates an accurate, contextually relevant caption. This involves the use of pre-trained CNNs for visual feature extraction and LSTMs to generate meaningful language descriptions based on those features."

1.1.1 Problem Description:

The project "Image Caption Using A.I" aims to develop a deep learning model capable of automatically generating descriptive captions for images. This involves integrating computer vision and natural language processing techniques to enable machines to interpret visual content and express it in human language.

Automatically generating captions for images is a complex task that necessitates a deep understanding of both visual elements and linguistic structures. The system must effectively detect and recognize various objects, comprehend scenes or locations, and interpret the interactions between objects. Subsequently, it should translate this understanding into well-formed sentences that accurately describe the image content.

To achieve this, the project employs a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks. The CNN component functions as an encoder, extracting salient features from the image, while the LSTM serves as a decoder, generating the corresponding textual description. This encoder-decoder architecture enables the model to learn the complex mappings between visual data and natural language.

The model is trained using the Flickr8k dataset, which comprises 8,092 images, each accompanied by five different captions. This dataset provides a diverse range of images and descriptive sentences, facilitating the model's ability to generalize across various scenarios and produce accurate captions for new, unseen images.

By addressing this problem, the project contributes to advancements in image understanding and language generation, with potential applications in areas such as assisting visually impaired individuals, enhancing image search engines, and improving content management systems.

1.2 Objectives

1. To collect and preprocess a diverse dataset of images representing various real-world scenarios.
2. To train machine learning models capable of generating accurate and context-aware captions for images by analyzing visual elements and their relationships.
3. To develop a user-friendly application that enables users to upload images and receive descriptive captions, enhancing accessibility and user interaction.

1.2.1 Dataset Collection and Preprocessing of Images

Dataset Collection:

The dataset for this project was sourced from multiple reliable platforms to ensure a diverse and comprehensive representation of visual scenarios. The primary dataset used is the MS COCO dataset, which contains images with associated captions describing their content. Additional datasets, such as Flickr8k and Flickr30k, were integrated to further enhance diversity and improve model generalization.

To ensure robust training and performance, images in the dataset cover various contexts, objects, and interactions. This diversity enables the model to handle a wide range of real-world scenarios, such as identifying single objects, complex interactions, and intricate relationships within images.

Expert guidance was sought to ensure that the captions accurately described the visual content, adhering to linguistic standards and contextual relevance. The combined datasets serve as a strong foundation for training a reliable image captioning model.

Image Preprocessing:

Preprocessing the collected images was a critical step in preparing the dataset for training the deep learning model. Raw images were resized to a consistent resolution to standardize input dimensions for the model. Pixel normalization was applied to scale the values within a range suitable for efficient model training, mitigating issues related to varying brightness and contrast.

To augment the dataset and improve model robustness, techniques such as rotation, flipping, cropping, and scaling were applied. These transformations helped simulate diverse conditions, enhancing the model's ability to generalize to unseen images. Noise reduction was

employed to minimize irrelevant background interference, and color normalization techniques ensured uniform feature representation.

Segmentation methods were utilized where necessary to highlight key regions of interest in images, helping the model focus on visually significant elements for caption generation. These preprocessing steps ensured the dataset's readiness for training an effective image captioning system.

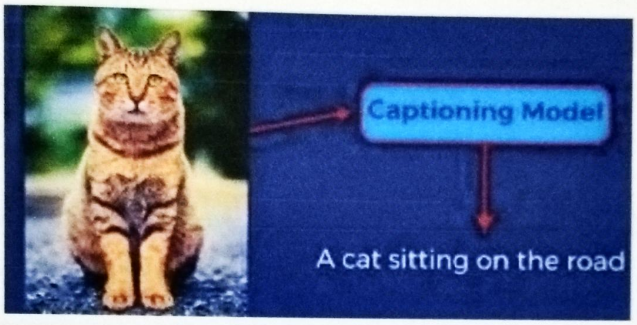


Figure 1.1: Sample image of image processing

1.2.2 Training Machine Learning Models for Caption Generation

The project employs state-of-the-art deep learning models, including a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, to generate captions for images. CNNs extract essential visual features from images, such as objects, textures, and spatial relationships. The extracted features are then processed by LSTM networks, which generate coherent and context-aware captions. Datasets such as MS COCO and Flickr8k provided a rich source of annotated images, enabling the model to learn complex mappings between visual content and textual descriptions. Evaluation metrics like BLEU and CIDEr were used to assess the quality of the generated captions, ensuring high accuracy and relevance. The system enables automated image captioning, making it a valuable tool for accessibility applications, multimedia content management, and visual search enhancements.

1.2.3 Developing a User-Friendly Captioning Tool

To enhance usability, the project utilizes Gradio to create an interactive interface where users can upload images and receive descriptive captions in real time. Gradio ensures a simple and user-friendly frontend, while the backend integrates the trained image captioning model for seamless processing. The system generates captions that describe objects and their contextual relationships, providing meaningful interpretations of visual content.

Chapter 02

Literature Survey

In this part of the report, we explore several existing research works and methodologies that form the foundation for our project. The literature survey provides insights into the techniques, models, and approaches utilized in the domain of image captioning. These references serve as a basis for understanding the state-of-the-art methods and help in benchmarking our model's performance against established frameworks.

[1]. "Image Captioning with Deep Learning Techniques" – Authors: Kelvin Xu, Jimmy Lei Ba, Ryan Kiros

This study explores an innovative approach to image captioning using deep learning models. The authors propose a combination of Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs) with attention mechanisms for generating descriptive captions. This model dynamically focuses on relevant parts of an image while generating each word in the caption.

Advantages:

- **Dynamic Attention Mechanism:** Enhances caption quality by focusing on important image regions.
- **Context-Aware Generation:** RNNs ensure captions are grammatically and contextually coherent.
- **Scalability:** The framework can generalize to diverse datasets, making it robust.
- **End-to-End Learning:** Simplifies training by integrating feature extraction and caption generation.

Disadvantages:

- **Computationally Expensive:** Requires significant resources for training models with attention mechanisms.
- **Dataset Dependency:** Performance relies heavily on the quality and diversity of training data.
- **Complexity:** Designing and tuning attention models adds complexity.

[2]. "Show and Tell: A Neural Image Caption Generator" – Authors: Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan

This paper proposes a simple yet effective image captioning framework based on an encoder-decoder architecture. The encoder uses a pre-trained CNN to extract image features, and the decoder employs an RNN to generate captions sequentially.

Advantages:

- **Pre-Trained Models:** Utilizes pre-trained CNNs, reducing training time and improving feature extraction.
- **Simplicity:** Encoder-decoder architecture is easy to implement and modify.
- **Efficient Training:** Avoids overfitting by leveraging large-scale pre-trained models.

Disadvantages:

- **Limited Context Understanding:** The lack of an attention mechanism may result in less accurate captions.
- **Vocabulary Limitations:** Generated captions are constrained by the predefined vocabulary.
- **Linear Generation:** Sequential generation can lead to slower inference times.

[3]. "Image Captioning with Transformers" – Authors: Ashish Vaswani, Noam Shazeer, Niki Parmar

This study leverages the transformer model, originally developed for natural language processing, for image captioning tasks. Transformers replace RNNs with self-attention mechanisms, enabling parallel processing of caption generation.

Advantages

- Parallelization: Significantly reduces training and inference times.
- Rich Context Modeling: Self-attention captures global relationships within captions.
- Versatility: Can be integrated with various image feature extraction methods.
- Scalability: Performs well on large datasets and adapts efficiently to complex tasks.

Disadvantages

- High Memory Usage: Self-attention mechanisms demand substantial memory resources.
- **Complex Training:** Requires fine-tuning and careful hyperparameter optimization.
- **Dataset Size:** Large datasets are necessary to prevent overfitting.

[4]. "Visual Attention Models for Image Captioning" – Authors: Jiebo Luo, Kyunghyun Cho

This paper emphasizes the role of visual attention in improving image captioning by highlighting specific image regions during caption generation. It proposes a novel approach integrating spatial and temporal attention mechanisms for detailed descriptions.

Advantages

- **Enhanced Detail:** Captions are more descriptive and contextually relevant.
- **Integration:** Supports various neural architectures for better performance.
- **Customizability:** Spatial attention adapts to different image datasets effectively.

Disadvantages

- **Resource Intensive:** Requires more computational power for attention-based architectures.
- **Annotation Dependency:** Relies on datasets with detailed annotations for optimal training.
- **Complexity:** Combining spatial and temporal attention increases implementation difficulty.

[5]. "Image Captioning with Reinforcement Learning" – Authors: Junhua Mao, Wei Xu

This paper integrates reinforcement learning to optimize image captioning models. The authors employ reward-based training to improve metrics like BLEU, ROUGE, and CIDEr.

Advantages

- **Optimized Outputs:** Improves evaluation metrics by tailoring captions to specific benchmarks.

- **Adaptability:** Adjusts to various datasets and evaluation criteria.
- **Error Correction:** Learns from mistakes during training to generate better captions.

Disadvantages

- **Metric Dependency:** Performance improvements depend on the chosen reward function.
- **Training Complexity:** Requires balancing supervised and reinforcement learning phases.
- **Stochastic Results:** Reinforcement learning introduces variability in outcomes.

Figure 2.1: Sample set of images from MS COCO dataset

Figure 2.2: Sample set of images from Flickr30k dataset

Chapter 3

System Design and Implementation

The system design for this project integrates deep learning algorithms to generate captions for images effectively. The primary model used is a Convolutional Neural Network (CNN) combined with a Recurrent Neural Network (RNN) with an attention mechanism. The CNN is responsible for extracting image features, while the RNN generates textual descriptions based on these features. The implementation was carried out using Google Colab, leveraging libraries such as TensorFlow and Keras seamless development and execution.

To ensure robust performance, the dataset was pre-processed to standardize image dimensions and enhance diversity using data augmentation techniques like rotation, flipping, and scaling. These methods improved the model's ability to generalize across varying conditions, such as lighting and perspective. The key steps in the system include preprocessing, feature extraction, caption generation, and evaluation.

In addition to the CNN for feature extraction, an attention mechanism was integrated into the RNN to dynamically focus on different parts of the image while generating each word in the caption. This significantly improved the contextual relevance of the generated descriptions. The attention-enhanced RNN was implemented using TensorFlow's attention layers, enabling accurate and coherent caption generation. The system was evaluated using standard metrics like BLEU, ROUGE, and CIDEr, ensuring high-quality captions for diverse images.

3.1 Convolutional Neural Network (VGG16)

VGG16 is a deep convolutional neural network that has been widely used in image classification tasks due to its simplicity and effectiveness. In this project, VGG16 is used as the feature extractor for the image captioning model. The architecture of VGG16 consists of 16 layers, with 13 convolutional layers and 3 fully connected layers, which help in capturing high-level image features such as textures, edges, and patterns that are crucial for accurate caption generation.

VGG16 was pre-trained on a large image dataset like ImageNet, which helps the model learn generic visual features that can be transferred to our task of image captioning. Transfer learning is used to fine-tune the VGG16 model for our specific dataset, where the pre-trained

layers of VGG16 are used as feature extractors, and the final fully connected layers are customized for the image captioning task.

The CNN extracts key image features from the input image, and these features are passed to an RNN decoder with an attention mechanism, which generates the caption. The attention mechanism enhances the system's ability to focus on specific regions of the image, improving the relevance of the generated text. [1]

Key steps in the implementation using VGG16:

- **Dataset Preparation**: Images were processed and resized to match the input size expected by the VGG16 model (typically 224x224 pixels). The corresponding captions were also pre-processed for model training.
- **Feature Extraction**: The VGG16 model was used to extract deep visual features from each image. These features represent high-level patterns in the images such as textures, colors, and object shapes.
- **Transfer Learning**: The pre-trained VGG16 model was used to reduce the amount of training data required and speed up the learning process. Only the fully connected layers were fine-tuned to adapt to the captioning task.
- **Integration with RNN and Attention Mechanism**: The extracted features were fed into an RNN decoder, which, along with the attention mechanism, dynamically focuses on different regions of the image while generating each word in the caption.

Using Google Colab, the model training and fine-tuning were optimized with GPU/TPU support, enabling faster processing and efficient model development. This setup provided a powerful environment for training the VGG16-based image captioning model and evaluating its performance. The pre-trained VGG16 model was fine-tuned to extract deep visual features from images, which were then passed to the LSTM-based decoder for caption generation. Colab's integration with TensorFlow and Keras allowed seamless implementation of the model, leveraging pre-built libraries and tools for efficient training and testing.

Additionally, the collaborative features of Google Colab facilitated teamwork, enabling multiple contributors to work on the same project in real time. Its cloud-based architecture eliminated the need for high-end local hardware, making it a cost-effective and scalable solution for model development. The use of Colab also ensured easy access to computational resources, allowing for experimentation with different hyperparameters and architectures to

achieve optimal performance in generating accurate and contextually relevant captions.

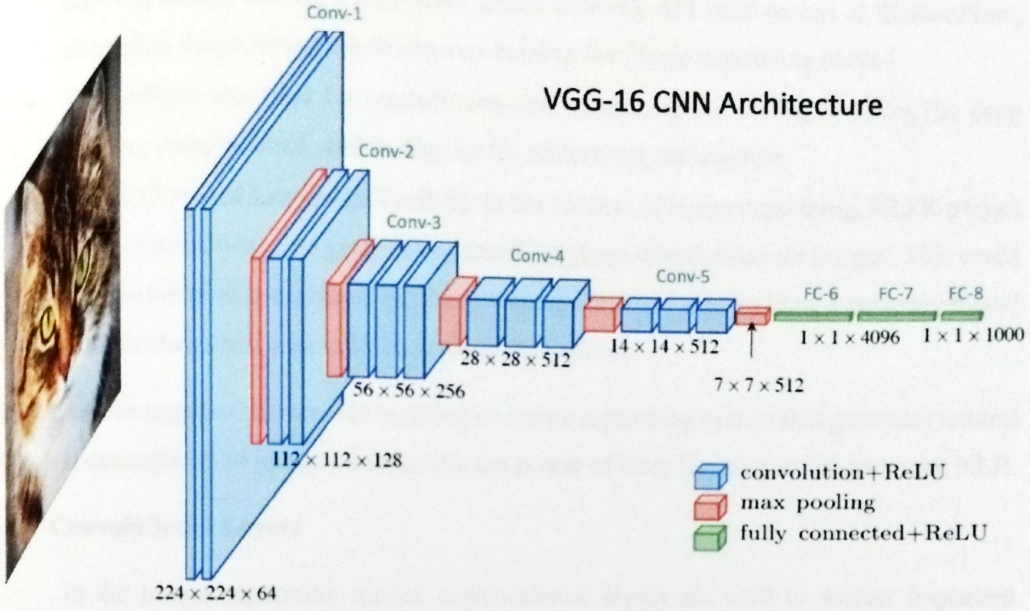


Figure 3.1: Block diagram of VGG-16 CNN Architecture

In this project, the VGG16 CNN architecture is used for feature extraction in image captioning. The architecture consists of convolutional layers that detect visual features, pooling layers that reduce dimensionality and improve efficiency, and fully connected layers that aggregate features into a comprehensive representation. These features are then passed to an RNN decoder to generate captions, providing accurate and relevant descriptions of the images based on the visual content.

3.1.1 Libraries Used

In this **Image Captioning Using AI** project, several key Python libraries were utilized for various stages of the process, from data preprocessing to model training:

- **NumPy**: This library was essential for handling large datasets and performing numerical operations. It was used for tasks like manipulating images, performing image transformations, and preparing the dataset for the machine learning models by resizing and normalizing images to appropriate formats.
- **Matplotlib**: Used for data visualization, Matplotlib helped in visualizing images and displaying results during the training process. It was also used to plot training metrics such as loss and accuracy to evaluate the performance of the image captioning model.

- **Keras** and **TensorFlow**: These were the core libraries for building and training the deep learning model. **Keras**, a high-level neural network API built on top of **TensorFlow**, simplified the process of defining and training the image captioning model.
- **TensorFlow** was used for constructing and optimizing the model, handling the deep learning computations, and managing the underlying architecture.
- **NLTK (Natural Language Toolkit)**: In the context of image captioning, NLTK played a role in processing and generating natural language descriptions for images. This could involve tokenizing captions, handling language features, and building a language model that matches the image with an appropriate caption.

These libraries together allowed for building an image captioning system that generates natural language descriptions of images, combining the power of deep learning techniques with NLP.

3.1.2 Convolutional Layers

In the image captioning model, convolutional layers are used to extract important features from images, which are then used to generate descriptive captions. The convolutional layers of the VGG16 CNN architecture process the input images using a series of filters. These filters slide over the image to perform convolution, highlighting low-level features like edges, textures, and shapes. As the image progresses through deeper layers, more complex features are captured.

The first convolutional layers of VGG16 apply filters to detect basic features. These layers use a ReLU activation function after convolution, which helps introduce non-linearity by converting negative values to zero. This allows the model to learn more complex patterns and prevents issues like vanishing gradients.

Additionally, max pooling layers are applied after the convolutional layers to down sample the feature maps, making the model more efficient. By reducing the spatial dimensions (e.g., from 3x3 to 2x2 regions), max pooling helps minimize computational complexity and overfitting. The deeper layers in the model use more filters (such as 64, 128) to capture higher-level, abstract features, enabling the model to understand and generate accurate captions based on the visual content of the images. The choice of the VGG16 architecture, with its predefined filters and weights from pre-training on large datasets like ImageNet, ensures robust feature extraction. Its sequential architecture, coupled with the regular use of ReLU activation and max-pooling, strikes a balance between computational efficiency and accuracy.

3.1.3 Flow Chart

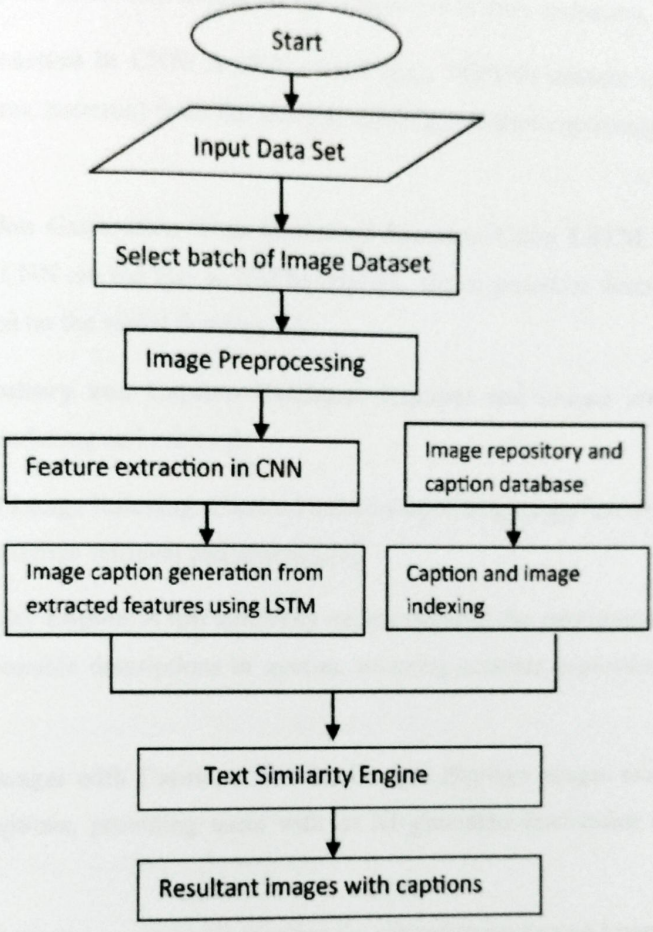


Figure 3.2: Flowchart – model workflow

The flowchart depicts the architecture and workflow of an "Image Captioning Using AI" project, where a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks is used to generate captions for images. Here's a brief description of each step:

1. **Start**: The process begins with initiating the system.
2. **Input Data Set**: The system accepts a dataset of images as input, which serves as the foundation for training and testing the model.
3. **Select Batch of Image Dataset**: Images are processed in batches to optimize memory usage and computational efficiency.

4. **Image Preprocessing**: Preprocessing steps such as resizing, normalization, and data augmentation are performed to prepare the images for feature extraction.

5. **Feature Extraction in CNN**: A CNN model (e.g., VGG16) extracts visual features (edges, textures, patterns) from the images, creating a feature representation for each image.

6. **Image Caption Generation from Extracted Features Using LSTM**: The features extracted by CNN are fed into an LSTM network, which generates descriptive textual captions based on the visual features. [2]

7. **Image Repository and Caption Database**: Captions and images are stored in a database for indexing and retrieval.

8. **Caption and Image Indexing**: Captions and corresponding image features are indexed to facilitate efficient retrieval and analysis.[4]

9. **Text Similarity Engine**: A text similarity engine matches the generated captions with the closest possible descriptions or queries, ensuring accurate captioning and search results.

10. **Resultant Images with Captions**: The final output displays images along with their generated captions, providing users with an AI-generated description of the visual content.[3]

This flowchart demonstrates a systematic pipeline for achieving automated image captioning, integrating advanced deep learning methods.

3.1.4 Image Preprocessing

Image preprocessing is a critical step in preparing the raw image data for the image captioning model. In this project, all images are resized to a uniform dimension of 299x299 pixels to match the input requirements of the CNN model (e.g., InceptionV3). Pixel values are normalized to fall within a range of 0 to 1 by dividing by 255. This normalization ensures numerical stability and faster convergence during training. Furthermore, unnecessary noise is reduced by applying filters to enhance the image quality. Images are converted to RGB format if not already, ensuring consistency in input channels. These preprocessing steps help the CNN extract high-quality visual features required for caption generation.

3.1.5 Image Processing

Image processing in this project involves extracting meaningful features from the images using a pretrained Convolutional Neural Network (CNN) such as InceptionV3 or VGG16. These models are used for feature extraction, where the final fully connected layers are removed, and the output feature maps are used as input to the captioning model. Data augmentation techniques such as random cropping, horizontal flipping, rotation, and brightness adjustments are applied to improve the dataset's variability, thereby preventing overfitting. The processed images are transformed into feature vectors that encode high-level visual information, which is later passed to the Long Short-Term Memory (LSTM) network for caption generation.

3.1.6 Model Compilation

The compiled model integrates the feature extraction CNN and the LSTM-based caption generation model. The Adam optimizer is chosen for its adaptive learning rate capabilities, ensuring efficient convergence during training. The loss function used is Categorical Cross entropy, tailored for sequence prediction tasks where the model predicts a sequence of words for the caption. The metrics include BLEU (Bilingual Evaluation Understudy) scores, which evaluate the quality of generated captions by comparing them to reference captions. The compilation ensures that the combined model can backpropagate errors effectively and update both the CNN and LSTM weights during training.

3.1.7 Model Training

The training process involves feeding image feature vectors extracted from the CNN into the LSTM network, along with their corresponding captions. The model learns to map the visual features to the sequence of words representing the caption. Training is conducted for multiple epochs, with a batch size optimized for the GPU's memory. Early stopping is used to halt training when validation accuracy no longer improves, preventing overfitting. Teacher forcing, a technique where the correct word from the sequence is provided as input during training, is used to speed up convergence. To monitor training, loss and BLEU score trends are plotted after each epoch. The final model produces captions by predicting one word at a time, using the previous word and visual features as input. Additionally, the learning rate is dynamically adjusted using a scheduler to maintain stable and efficient optimization throughout the training process.

3.2 Connecting Frontend and Backend

This project employs Python for the backend and Gradio for the frontend to create a seamless system for image captioning using AI. The backend processes input images through a CNN-LSTM model to generate captions, while Gradio provides a user-friendly interface for interacting with the system.

3.2.1 Python (Backend)

Python serves as the foundation of the backend system, handling the core functionalities of the CNN-LSTM-based model designed for image captioning. It plays a pivotal role in tasks such as extracting visual features from images using a pre-trained CNN (like VGG16) and generating descriptive captions through LSTM-based architectures.

Python's extensive ecosystem of libraries, including TensorFlow, Keras, and NumPy, empowers efficient execution of deep learning models and robust data management. These libraries provide advanced functionalities, allowing the system to process complex tasks with high precision and computational efficiency.

The backend processes user-uploaded images by leveraging pre-trained models to extract deep visual features and passing them to the caption generator. This ensures rapid processing and accurate caption outputs while maintaining compatibility with Gradio for seamless interaction between the backend model and the frontend interface.

3.2.2 Gradio (Frontend)

Gradio is used to design the frontend interface, providing an intuitive and interactive platform for users. The interface enables users to upload images and view the corresponding captions generated by the backend model.

Gradio's simplicity eliminates the need for traditional web development tools like HTML, CSS, and JavaScript while offering a responsive and visually appealing interface. It supports real-time interactions, allowing users to upload images and receive captions instantly. This feature enhances usability and makes the system accessible to a broader audience, even those with limited technical expertise.

Gradio also streamlines the deployment process by integrating directly with the Python backend, ensuring a seamless flow of data and results between the frontend and backend. Gradio simplifies the deployment process by integrating seamlessly with the Python backend.

```
[132]
import gradio as gr
from PIL import Image

def generate_caption_gradio(image):
  """Generates caption for the given image."""
  image = image.convert('RGB')
  image = image.resize((224, 224))
  image = img_to_array(image)
  image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
  image = preprocess_input(image)

  feature = vgg_model.predict(image, verbose=0)

  caption = predict_caption(model, feature, tokenizer, max_length)

  return caption

iface = gr.Interface(
    fn=generate_caption_gradio,
    inputs=gr.Image(type="pil"),
    outputs="text",
    title="Image Caption Generator",
    description="Upload an image to generate a caption."
)


iface.launch()
```

Figure 3.3: Code snippet that is used to connect front-end and back-end

3.3 Backend Images

```
epochs = 5
batch_size = 32
steps = len(train) // batch_size

for i in range(epochs):
    generator = data_generator(train, mapping, features, tokenizer, max_length, vocab_size, batch_size)
    model.fit(generator, epochs=1, steps_per_epoch=steps,verbose=1)
```

Figure 3.4: Code snippet for training the model

```
def idx_to_word(integer, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == integer:
            return word
    return None

def predict_caption(model, image, tokenizer, max_length):
    in_text = 'startseq'
    for i in range(max_length):
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        sequence = pad_sequences([sequence], max_length)
        yhat = model.predict([image, sequence], verbose=0)
        yhat = np.argmax(yhat)
        word = idx_to_word(yhat, tokenizer)
        if word is None:
            break
        in_text += " " + word
        if word == 'endseq':
            break

    final_caption = in_text.replace('startseq', '').replace('endseq', '').strip()
    return final_caption
```

Figure 3.5: Code snippet for caption prediction

Chapter 4

Results

The results of the Image Captioning Using AI project are showcased through an interactive Gradio-based web application, where users can upload images for real-time caption generation. The Gradio interface processes the input image, applies the trained CNN-LSTM model, and displays the generated captions with contextual relevance.

Snapshots of the application highlight the user-friendly interface, with options to upload images and view the resulting captions directly. The results page displays the caption generated by the model, providing a seamless and intuitive way for users to understand the content of an image.

This interactive system makes it easy for users, including those without technical expertise, to access automated image captioning capabilities, enhancing the usability and practical application of the model. The interface also allows users to experience the high accuracy and performance of the trained model, offering insight into how artificial intelligence can be leveraged for efficient visual content understanding.[1],[2]

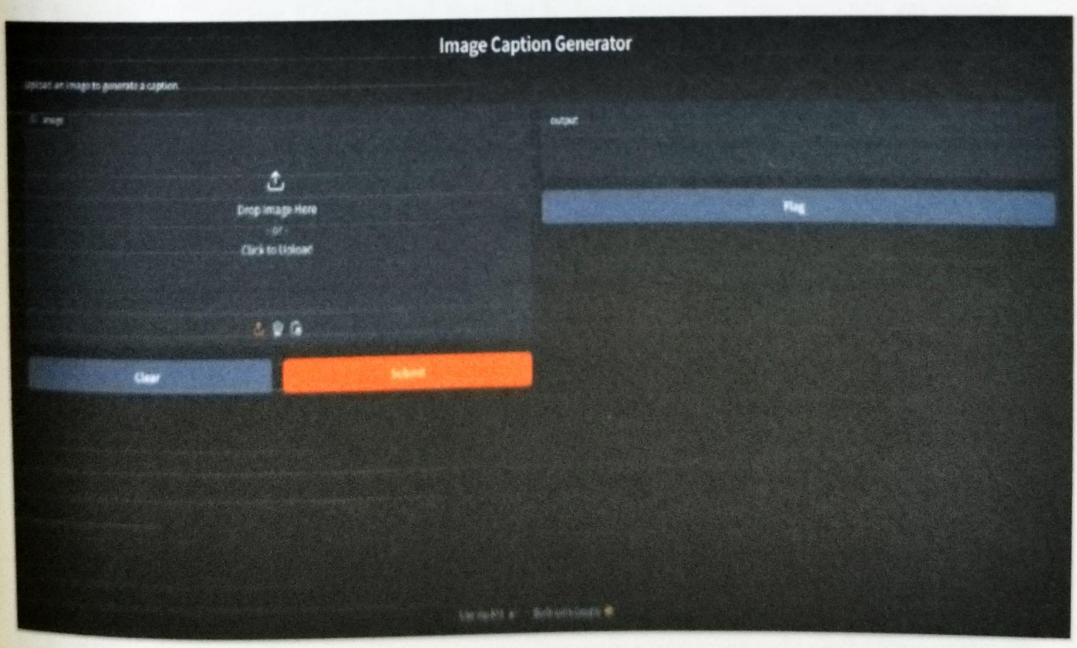


Figure 4.1: Frontend – Gradio interface

Figure 4.2: Selecting an image for generating the caption from the Flickr30k dataset

4.1 Results – Caption Generated for the below image: "two people on snow covered hill".

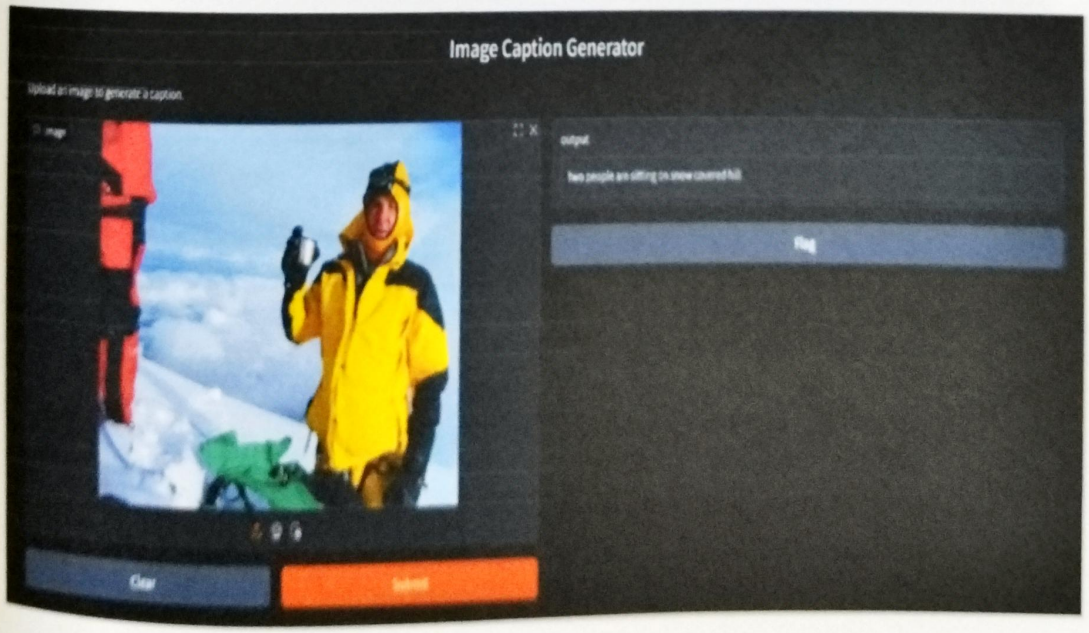


Figure 4.3: Caption generated for the sample - 1

4.2 Results – Caption generated for the below image: "dog running through snow"

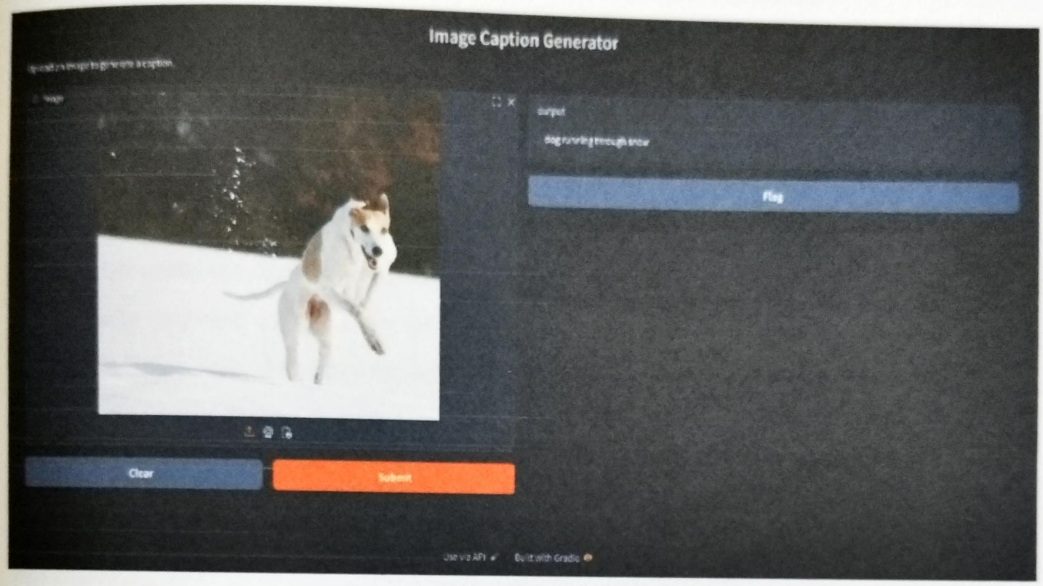


Figure 4.4: Caption generated for the sample - 2.

4.3 Results – Caption generated for the below image: "two people are riding through ditch"

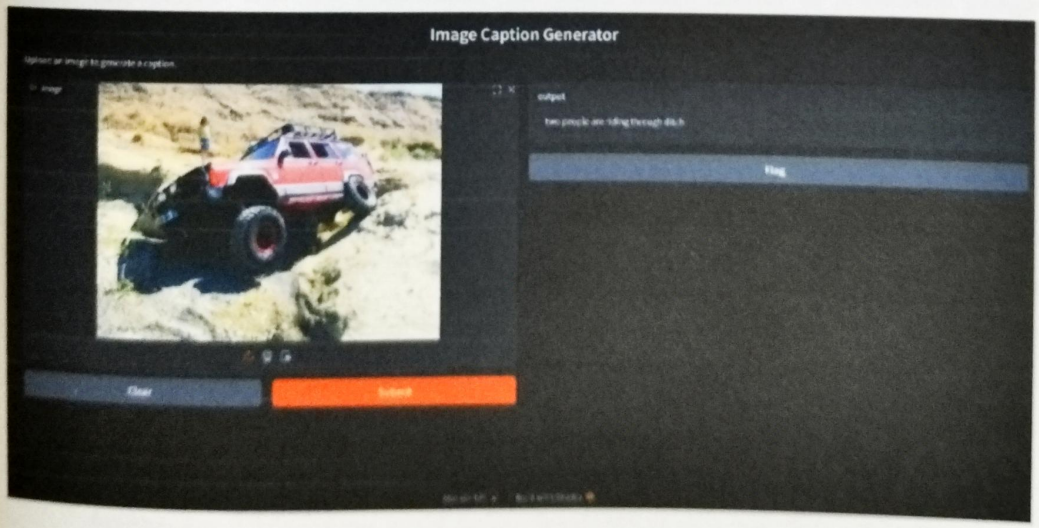


Figure 4.5: Caption generated for the sample - 3

Chapter 5

Conclusion and Future Enhancement

5.1 Conclusion

Using deep learning for image captioning is an innovative approach that bridges the gap between visual and textual understanding. The current project, which employs Convolutional Neural Networks (CNNs) for feature extraction and Long Short-Term Memory (LSTM) networks for caption generation, demonstrates the ability to generate meaningful and contextually relevant captions for images. This technology provides a cost-effective and efficient solution for automating the description of visual content, offering significant utility in fields such as accessibility, content creation, and multimedia management.

While the current model achieves promising results, further improvements can enhance its performance and adaptability. Expanding the dataset to include a diverse range of images and exploring more advanced architectures, such as transformer-based models, could improve the quality and accuracy of captions. Additionally, integrating the system into user-friendly applications, such as mobile or web-based platforms, would make the technology more accessible and practical for end users.

This project lays the groundwork for future advancements in A.I-driven image understanding. By refining the model, incorporating adaptive learning, and optimizing for real-time applications, this system has the potential to revolutionize how visual data is interpreted and utilized, contributing to advancements in AI and its applications across various domains.

5.2 Future Enhancement

Future enhancements for the "Image Captioning Using A.I" project could focus on several key areas. Firstly, expanding the dataset by incorporating a broader variety of images from diverse environments, lighting conditions, and contexts would improve the model's generalization and robustness. Increasing the dataset size and variety would also enable the model to handle more complex and ambiguous images effectively.

Another key enhancement could be the development of a real-time application using mobile or web platforms for generating captions on the go. Such an application could provide seamless access to image captioning capabilities for everyday users & content creators.

References

[1]. Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. Proceedings of the International Conference on Machine Learning (ICML), 2048–2057.

[2]. Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3156–3164.

[3]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS), 30, 5998–6008.

[4.] Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing When to Look: Adaptive Attention for Visual Captioning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 375–383.

[5]. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. L. (2015). Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). International Conference on Learning Representations (ICLR).