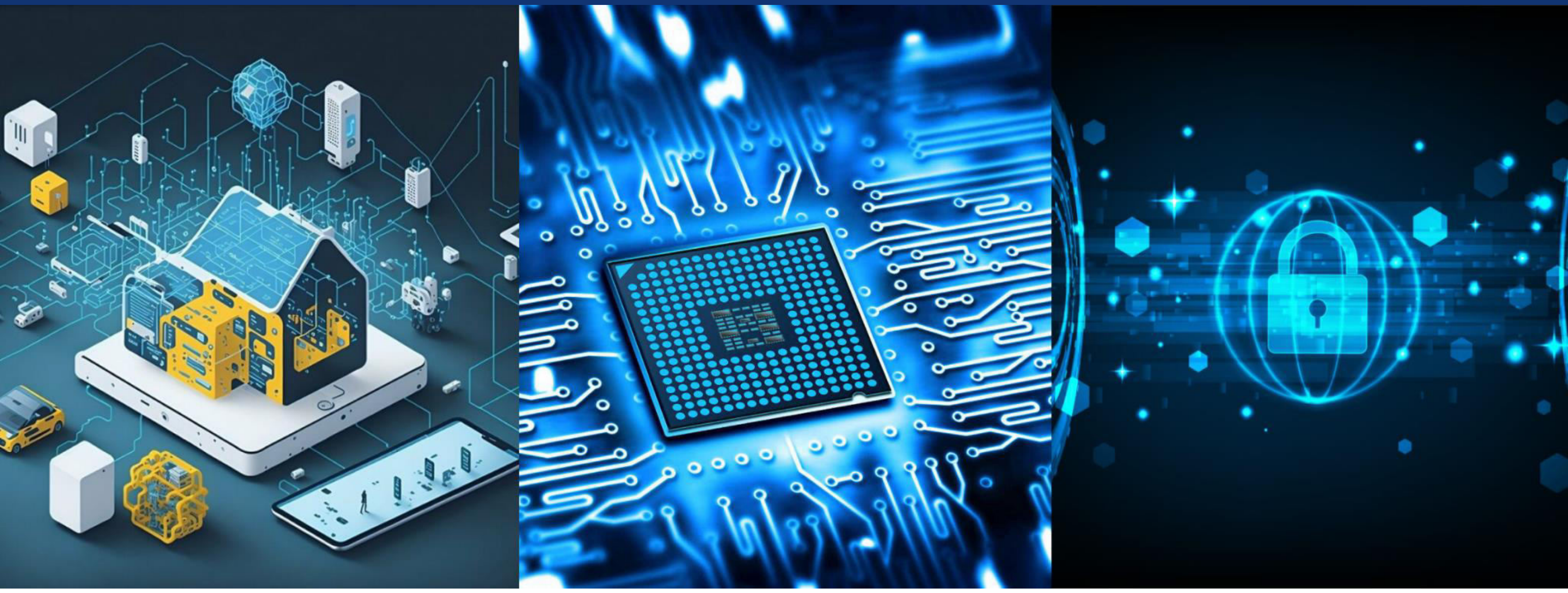
IJIRCCE

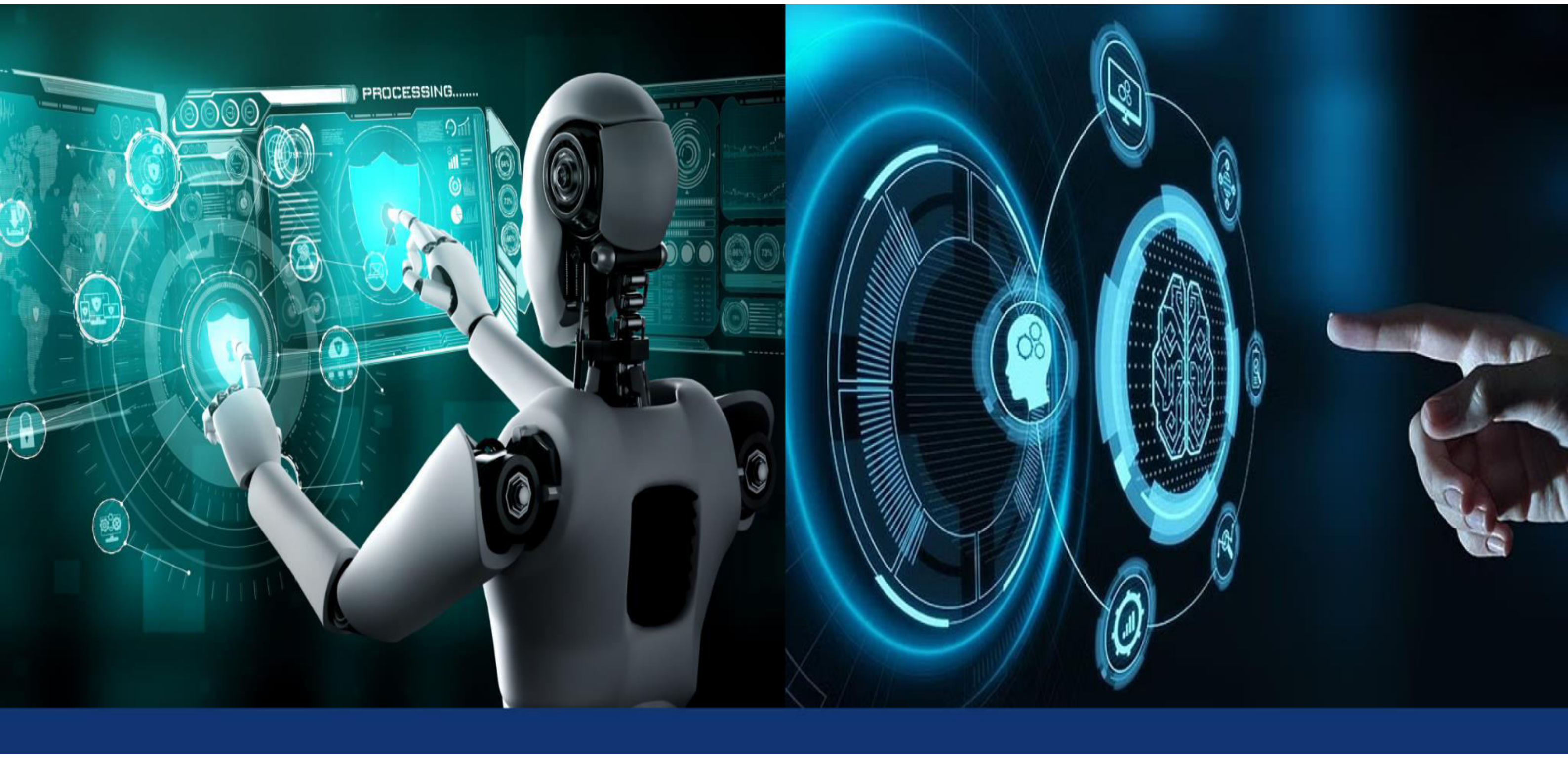
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 5, May 2025

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305158

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Image Captioning using BILSTM Based YOLO in Deep Learning

M. Umamaheshwari

Assistant Professor, Dept. of IT, The Kavery Engineering College, Salem, Tamil Nadu, India

Pavithra P, Nisha S, Nivetha S, Parameshwari S

Dept. of IT, The Kavery Engineering College, Salem, Tamil Nadu, India

ABSTRACT: Image captioning is a pivotal task in artificial intelligence that integrates computer vision and natural language processing to generate textual descriptions of visual content. This Paper presents an innovative system that combines the YOLO (You Only Look Once) object detection algorithm with a Bidirectional Long Short-Term Memory (BiLSTM) network to generate accurate and context-aware captions for images. The YOLO model serves as the feature extraction module, enabling efficient and real-time object detection within images. These detected features are then processed by a BiLSTM-based sequence generation model, which analyzes the visual context in both forward and backward directions, enhancing the quality and coherence of the generated captions. To further refine the captioning process, an attention mechanism is integrated into the system, allowing the model to focus on the most relevant regions of the image during caption generation.

I. INTRODUCTION

In recent years, the intersection of computer vision and natural language processing has led to significant advancements in artificial intelligence, particularly in the area of image captioning. Image captioning is the process of generating textual descriptions for images, a task that demands both accurate visual understanding and fluent language generation. With applications ranging from assistive technologies for the visually impaired to intelligent content indexing, the need for effective image captioning systems has grown substantially. This Paper presents a hybrid deep learning-based approach combining YOLO for object detection and BiLSTM for sequence modeling to generate more meaningful and contextually accurate captions.

II. SYSTEM STUDY

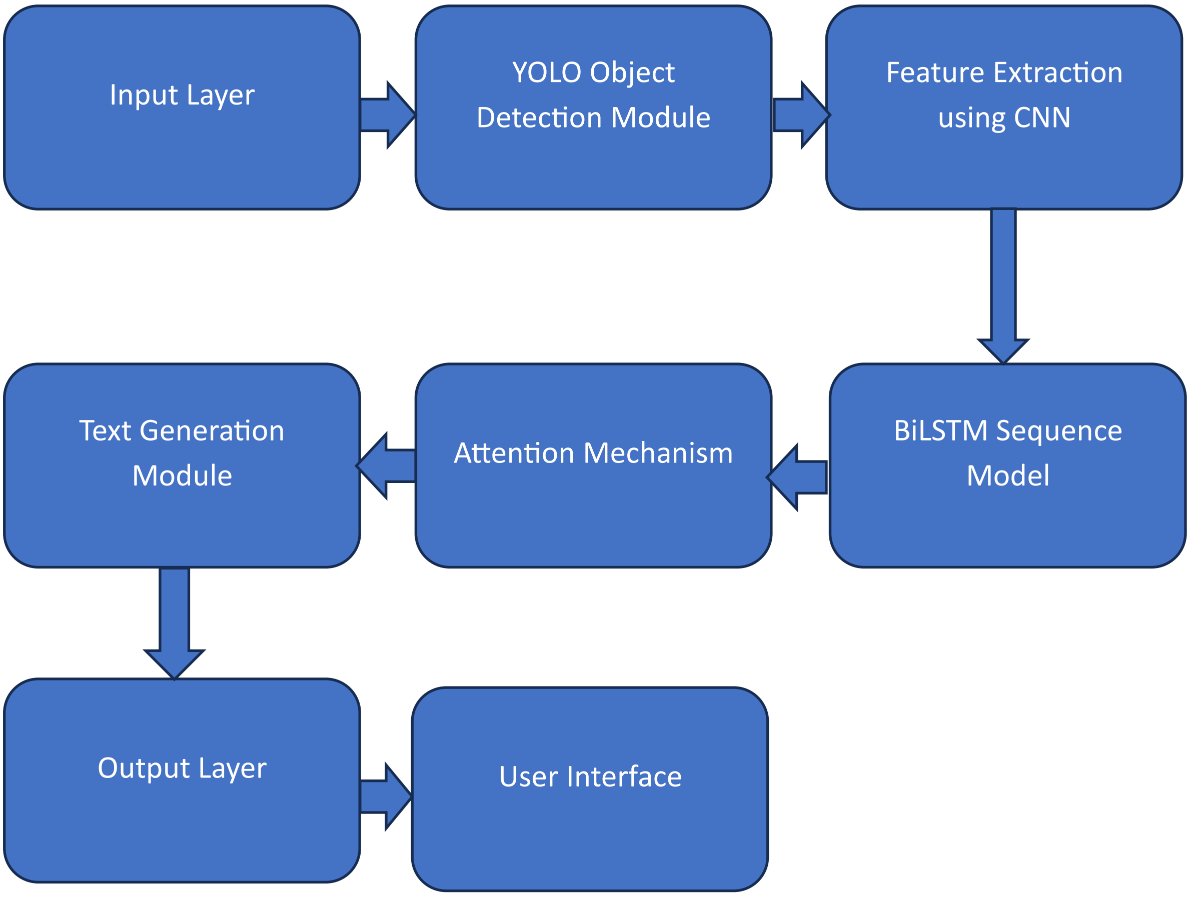
Existing System

Traditional image captioning systems typically rely on a two-stage process: first, extracting image features using Convolutional Neural Networks (CNNs), and then generating captions using Recurrent Neural Networks (RNNs) such as standard LSTMs. These systems are often trained on large datasets such as MS-COCO, and although they have shown considerable progress, several limitations persist. Conventional models often struggle with generating semantically rich captions, especially when images contain complex scenes or multiple interacting objects. Additionally, standard object detection mechanisms may miss smaller or overlapping objects, which negatively impacts the captioning accuracy.

Proposed System

The proposed system overcomes the limitations of traditional captioning models by integrating YOLO-based object detection with a BiLSTM-based language model, enhanced by an attention mechanism. YOLO (You Only Look Once) provides real-time, high-accuracy object detection by processing the image as a single regression problem. It is capable of detecting multiple objects in a single pass, making it ideal for complex image analysis. The extracted features from YOLO are then passed through a CNN backbone (such as Darknet-53) to capture detailed visual representations.

Architecture Diagram:



Architecture Diagram: Workflow

The proposed image captioning system follows a structured and modular workflow that integrates both object detection and natural language processing techniques to generate meaningful textual descriptions for input images. The architecture consists of several key components arranged in a sequential pipeline to ensure accurate and efficient caption generation. The process begins at the Input Layer, where the user uploads an image through a web or mobile interface. This image is passed to the YOLO Object Detection Module, which performs real-time object detection using a pre-trained YOLO model (e.g., YOLOv4 or YOLOv5). YOLO processes the image as a whole and detects multiple objects with their bounding boxes and class probabilities, offering high-speed and accurate feature localization. Next, the detected object regions are forwarded to the Feature Extraction Module, which uses a CNN backbone such as Darknet-53 to extract high-level semantic features from the image.

III. SYSTEM SPECIFICATION

The system specification describes the overall framework and setup required to implement and deploy the image captioning model using a BiLSTM-based YOLO architecture.

Hardware Requirements

Efficient training and execution of deep learning models require a well-defined hardware setup that balances performance and resource usage.

Component	**Specification**
CPU	Intel Core i5/i7, AMD Ryzen 5/7
RAM	8 GB minimum (16 GB recommended)
Storage	500 GB HDD or SSD (SSD preferred)
GPU (Recommended)	NVIDIA GTX 1650 / RTX 2060 or higher with CUDA support
Display	1080p monitor for clear visualization
Input Devices	Standard Keyboard and Mouse

Software Requirements

The software environment includes the operating system, programming languages, libraries, and frameworks used in the development lifecycle.

Component	**Specification**
Operating System	Windows 10 / Ubuntu 20.04 LTS
Programming Language	Python 3.8 or higher
Development Environment	Anaconda, Jupyter Notebook
Libraries & Frameworks	TensorFlow, Keras, PyTorch, OpenCV, NumPy, Pandas
NLP Tools	NLTK, SpaCy
Deployment Tools	Flask / Streamlit / Django
Visualization Tools	Matplotlib, Seaborn, Plotly
Version Control	Git & GitHub

Development Libraries and Their Role

Library	**Purpose**
TensorFlow/Keras	Model training and evaluation (BiLSTM)
OpenCV	Image preprocessing and manipulation
PyTorch	Alternative to TensorFlow (optional YOLO models)
NumPy	Numerical computation
Pandas	Data manipulation and management
Matplotlib	Visualization of training metrics and outputs
Flask/Streamlit	Web application interface
NLTK/SpaCy	Text processing for captions

Image Acquisition Module

This is the initial module of the system where the user uploads or captures an image using a device such as a camera or file browser. The module supports multiple image formats including JPG, PNG, and JPEG.

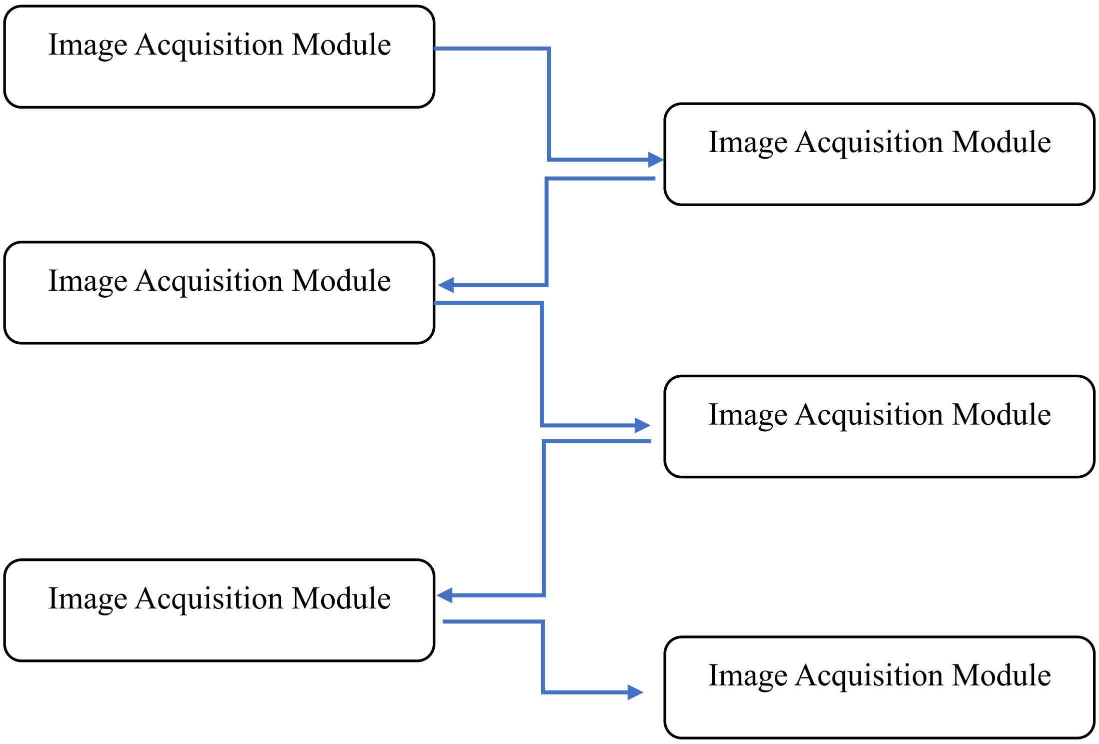
Object Detection Module (YOLO-Based)

In this module, the uploaded image is passed through a YOLO (You Only Look Once) model, which detects various objects within the image and draws bounding boxes around them.

Sequence Generation Module (BiLSTM-Based)
This is the core of the captioning model. The BiLSTM (Bidirectional Long Short-Term Memory) network receives the visual features and generates a sequence of words forming the caption. The bidirectional nature of LSTM helps understand the context from both past and future tokens.

Summary of Modules

Module Name	**Main Responsibility**
Image Acquisition	Input and preprocessing of the image
Object Detection (YOLO)	Detect and label objects within the image
Feature Extraction	Extract visual features from object regions
Caption Generation (BiLSTM)	Generate meaningful sentences from image features
Post-Processing	Clean and format the generated captions
User Interface	Interaction layer for user input/output
Evaluation	Assess the system's performance and accuracy



1. Image Input Module The process begins when a user uploads an image through the user interface. This module ensures the image is appropriately preprocessed for compatibility with the object detection system. The preprocessing involves resizing, normalization, and format conversion, preparing the image for further analysis.

2. YOLO-Based Object Detection Module Once the image is preprocessed, it is passed to the YOLO (You Only Look Once) model, a real-time object detection system. YOLO scans the image and identifies prominent objects along with their bounding boxes and class labels. This step provides semantic insight into the content of the image and highlights key regions of interest.

3. Feature Extraction using CNN Backbone The detected objects are then processed using a Convolutional Neural Network (CNN) backbone—typically Darknet-53 or ResNet—which extracts deep feature representations from the image. These features form a condensed and high-level understanding of the image content, serving as the input for the sequence model.

4.BiLSTM Sequence Modeling Module The extracted features are forwarded to a Bidirectional Long Short-Term Memory (BiLSTM) network. Unlike traditional LSTMs, BiLSTM processes input data in both forward and backward directions, allowing the model to capture full context from the image features. This dual processing improves the semantic richness and coherence of the generated captions.

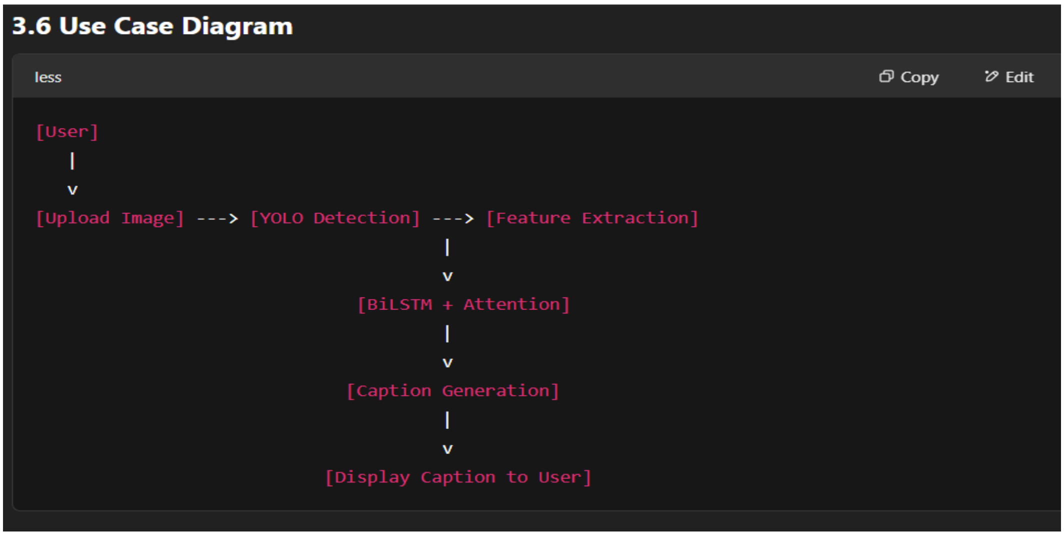
5. Attention Mechanism An attention layer is incorporated to enhance the model's focus on the most relevant image features. Instead of treating all parts of the image equally, the attention mechanism dynamically prioritizes certain regions, ensuring the generated captions are contextually accurate and relevant to the visual content.

6.Text Generation Module The context-aware features from the BiLSTM and attention layers are passed through a fully connected layer followed by a softmax classifier. This module sequentially generates words to form the final image caption. It uses the learned linguistic patterns and visual cues to ensure syntactic correctness and semantic relevance.

7. Output Interface Module Finally, the generated caption is displayed on the output interface. This user-friendly front end provides real-time feedback, enabling users to upload different images and instantly receive descriptive captions. The interface can be implemented as a web or mobile application using frameworks like Flask or Django.

IV. SYSTEM DESIGN

The system design phase serves as the architectural backbone of the Paper, transforming functional and non-functional requirements into a detailed system blueprint. In the case of "Image Captioning Using BiLSTM Based YOLO in Deep Learning," this chapter lays out the structural and logical framework that governs how the components work individually and in synergy. The system aims to identify objects within images and generate grammatically correct and semantically relevant textual descriptions. The core focus is on seamless integration of real-time object detection (YOLO) and natural language generation (BiLSTM), ensuring that the output is not only quick but contextually accurate.


3.6 Use Case Diagram

```
[User]
   |
   v
[Upload Image] ---> [YOLO Detection] ---> [Feature Extraction]
                                      |
                                      v
                              [BiLSTM + Attention]
                                      |
                                      v
                             [Caption Generation]
                                      |
                                      v
                           [Display Caption to User]
```

V. SYSTEM TESTING

The basic principle that software engineers must understand before applying methods for designing effective test cases is to guide software testing. Davis (DAV95) has proposed a series of principles that apply to the testing discipline:

Unit Testing

Unit testing verifies the smallest units of the software—individual modules such as the image preprocessing module, the BiLSTM caption generation module, and the YOLO object detection component.

Integration Testing

Integration testing was performed to validate the interaction between YOLO and BiLSTM. This involved checking:

White Box Testing

White-box testing was used to validate internal logic and data flow, especially in the following areas:

- Validation of conditional statements within the BiLSTM attention mechanism.
- Loop structures in the sequence generation logic.
- Activation and gradient flow in each layer during training.

Control structures were exhaustively examined to ensure robust model performance during training and inference.

Acceptance Testing

Acceptance testing focused on ensuring the system met all functional and user experience requirements. These tests included:

- Caption relevance as validated by end users.

Alpha Testing

Alpha testing was conducted in a controlled lab environment with select internal testers. Testing focused on:

- Testing the responsiveness and performance of the system.
- Capturing and logging any unexpected errors during image upload or caption retrieval.

Beta Testing

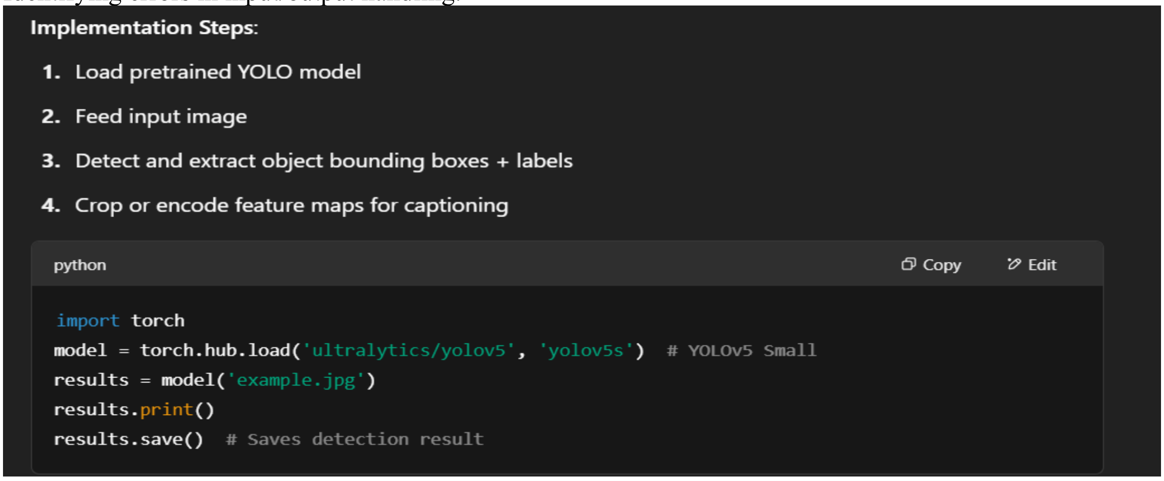
Beta testing involved real users interacting with the system via the proposed application interface. Key testing goals included:

- Identifying usability issues not caught during alpha.

Black Box Testing

Black-box testing was conducted to evaluate the system from a purely functional perspective without examining internal code. Focus areas included:

- Ensuring every image input produces a caption.
- Identifying errors in input/output handling.



Implementation Steps:

1. Load pretrained YOLO model
2. Feed input image
3. Detect and extract object bounding boxes + labels
4. Crop or encode feature maps for captioning

```python
import torch
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')  # YOLOv5 Small
results = model('example.jpg')
results.print()
results.save()  # Saves detection result
```

Test Cases

The primary objective of test case development was to identify maximum potential errors and ensure robust system behavior. Sample test cases included:

Test Case ID	Input Scenario	Expected Output	Result
TC01	Image of a dog in a park	"A dog is standing in the park"	Pass
TC02	Corrupted image file	Error message displayed	Pass
TC03	Image with multiple objects	Caption listing all objects	Pass
TC04	No object in the image	"No recognizable object found"	Pass
TC05	Image with text overlay	Caption ignores text content	Pass

VI. CONCLUSION

The proposed Paper, titled “IMAGE CAPTIONING USING BILSTM BASED YOLO IN DEEP LEARNING”, presents a comprehensive system that effectively combines object detection with natural language processing to generate meaningful captions for input images. The developed system utilizes the YOLO (You Only Look Once) model to perform real-time and high-precision object detection. The outputs of YOLO are then fed into a Bi-directional Long Short-Term Memory (BiLSTM) network to generate human-like descriptive captions based on the objects and context within the image. The BiLSTM model plays a crucial role in capturing temporal dependencies and contextual flow in both directions, thereby producing more coherent and accurate textual descriptions. The use of advanced deep learning techniques ensures that the system is robust, scalable, and capable of learning complex features and patterns from image datasets.

REFERENCES

1. Kaur, Mehzabeen, and Harpreet Kaur. "An Efficient Deep Learning based Hybrid Model for Image Caption Generation." International Journal of Advanced Computer Science and Applications 14, no. 3 (2023).
2. Wajid, Mohammad Saif, Hugo Terashima-Marin, Peyman Najafirad, and Mohd Anas Wajid. "Deep learning and knowledge graph for image/video captioning: A review of datasets, evaluation metrics, and methods." *Engineering Reports* 6, no. 1 (2024): e12785.
3. Negi, Pooja R., and Sanjay H. Buch. "Effective Image Captioning With YOLO and LSTM."
4. Patel, Mr Hardik. *Efficient Image Captioning Method using Deep Learning*. Diss. GUJARAT TECHNOLOGICAL UNIVERSITY AHMEDABAD, 2025.
5. Kaliappan, Jayakumar, Senthil Kumaran Selvaraj, and Baye Molla. "Caption Generation Based on Emotions Using CSPDenseNet and BiLSTM with Self-Attention." *Applied Computational Intelligence & Soft Computing* (2022).

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING