REAL-TIME SCENE IDENTIFICATION AND DESCRIPTION THROUGH VIDEO CAPTIONING

A PROJECT REPORT

Submitted By

DEJAH M 205001026

KRITHIKA SWAMINATHAN 205001057

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Kalavakkam - 603110

May 2024

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled **"REAL-TIME SCENE IDENTIFICATION AND DESCRIPTION THROUGH VIDEO CAPTIONING"** is the *bonafide* work of "**DEJAH M (205001026)**, and **KRITHIKA SWAMINATHAN (205001057)**" who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. T. T. MIRNALINEE
HEAD OF THE DEPARTMENT
Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Dr. K. R. SARATH CHANDRAN
SUPERVISOR
Assistant Professor,
Department of CSE,
SSN College of Engineering,
Kalavakkam - 603 110

Place:
Date:

Submitted for the examination held on. . . . . .

Internal Examiner **External Examiner**

ACKNOWLEDGEMENTS

I thank GOD, the almighty for giving me strength and knowledge to do this project.

I would like to thank and express my deep sense of gratitude to my project guide **Dr. K. R. SARATH CHANDRAN**, Assistant Professor, Department of Computer Science and Engineering, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped me to shape and refine my work.

My very sincere thanks to **Dr. T. T. MIRNALINEE**, Professor and Head of the Department of Computer Science and Engineering, for her words of advice and encouragement, and I would like to heartily thank our respected project Coordinator **Dr. B. BHARATHI**, Associate Professor, Department of Computer Science and Engineering for her valuable suggestions throughout this project.

I express my deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. I also express my appreciation to our **Dr. V. E. ANNAMALAI**, Principal, for all the help he has rendered during this course of study.

I would like to extend my sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of my project work. Finally, I would like to thank my parents and friends for their patience, cooperation and moral support throughout my life.

DEJAH M **KRITHIKA SWAMINATHAN**

ABSTRACT

Over the years, there has been a huge shift from text-based content to video-based content in all walks of life. In this era of real-time multimedia content, the demand for real-time video description and captioning is indispensable. To understand the scene and generate relevant captions, a system must first be able to identify objects in the world around us. Therefore, this research work introduces a novel approach in which features extracted from VGG16 CNN and object detection results from YOLOv8 model are concatenated, and the fused features are passed to an LSTM-based encoder-decoder architecture integrated with an attention mechanism for caption generation from live video streams. After comparison with several model variations, the proposed model is fixed upon. Trained on the MSVD dataset after refining the original captions for improved accuracy, the model showcases promising performance.

TABLE OF CONTENTS

ABSTRACT iii

LIST OF TABLES vii

LIST OF FIGURES viii

1 INTRODUCTION 1

1.1 MOTIVATION 1

1.2 BACKGROUND 3

1.3 PROBLEM DEFINITION 5

1.4 ORGANIZATION OF THE REPORT 6

2 LITERATURE SURVEY 7

2.1 RELATED WORK 7

2.1.1 Related Work on Surveying Deep Learning Approaches for Video Captioning 7

2.1.2 Related Work on Unique Approaches for Video Captioning 8

2.1.3 Related Work on Image Captioning Techniques that are Adaptable to Video Captioning 13

2.2 RESEARCH GAPS IDENTIFIED 16

2.3 RESEARCH OBJECTIVES 16

3 PROPOSED METHODOLOGY 17

3.1 OVERVIEW 17

3.2 DESCRIPTION OF COMPONENTS 20
3.2.1 Video Stream Processing 20
3.2.2 Visual Model 20
3.2.3 Language Model 21
3.3 MODEL SELECTION 22
3.4 FEATURE EXTRACTION 24
3.5 BUILDING THE MODEL 28
3.5.1 Attention Layer 28
3.5.2 LSTM Encoder 30
3.5.3 LSTM Decoder 31

4 EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS 33
4.1 DATASET 33
4.1.1 Dataset Description 33
4.1.2 Caption Refinement 33
4.1.3 Caption Preprocessing 34
4.2 ECOSYSTEM 36
4.3 EXPERIMENTS CONDUCTED 36
4.3.1 Evaluation of Model Variations 36
4.3.2 Testing of the Proposed Model 41
4.3.3 Testing Real-Time Video Captioning 43
4.4 PERFORMANCE ANALYSIS 44
4.4.1 Performance Metrics 44
4.4.2 Comparison with State-of-the-art (SOTA) Models 46
4.4.3 Sample Output 48
4.4.4 Qualitative Analysis 51

5 SOCIAL IMPACT AND SUSTAINABILITY **53**

5.1 SOCIETAL IMPACT 53

5.2 HEALTH AND SAFETY 53

5.3 LEGAL CONSIDERATIONS 54

5.4 ENVIRONMENTAL IMPACT 54

5.5 CULTURAL SENSITIVITY 54

6 CONCLUSIONS AND FUTURE WORK **55**

6.1 CONCLUSION 55

6.2 FUTURE SCOPE 55

A GLOSSARY **56**

A.1 DATASETS 56

A.2 PRE-TRAINED MODELS 56

A.3 STATE-OF-THE-ART (SOTA) Models 56

LIST OF TABLES

2.1 Literature Survey 1 - Deep Learning Approaches for Video Captioning 7
2.2 Literature Survey 2 - Summary of existing approaches for scene description 10
2.3 Literature Survey 3 - Summary of existing approaches for scene description 12
2.4 Literature Survey 4 - Summary of existing approaches for scene description 15

4.1 Examples for Replacing Synonyms 34
4.2 Evaluation of Model Variations. T-A:Training Accuracy(%), V-A:Validation Accuracy(%), B:BLEU-4(%), M-METEOR(%), R:ROUGE-L(%), C:CIDER(%), T:Average time taken for caption generation(in seconds) 37
4.3 Performance of Proposed Model on MSVD dataset. T-A:Training Accuracy(%), V-A:Validation Accuracy(%), B:BLEU-4(%), M-METEOR(%), R:ROUGE-L(%), C:CIDER(%), T:Average time taken for caption generation(in seconds) 42
4.4 Performance comparison with state-of-the-art approaches. BLEU-4, METEOR, ROUG-L and CIDER scores are represented as percentages. 46

LIST OF FIGURES

3.1 Architecture of the Proposed System 18
3.2 VGG16 CNN Architecture. Features are extracted from the final fully connected layer. [42] 23
3.3 A single LSTM cell [39] 24
3.4 Outline of the implemented Feature Extraction 25
3.5 YOLOv8 result format for an image. (x1,y1) and (x2, y2) represent the coordinates of the top-left and bottom-right corners of the bounding box of the detected object. The confidence score measures how confident the model is of its prediction of that object. The class_id is one of 80 classes (0-79). 26
3.6 A screenshot of some of the extracted feature files 27
3.7 Encoder-Decoder Architecture - TRAINING 31
3.8 A snapshot of the model getting trained 32
3.9 Trained model 32

4.1 Steps in Caption Preprocessing 35
4.2 Variations in Accuracy on Varying Vocabulary Size 38
4.3 Variations in Performance on Varying Vocabulary Size 38
4.4 Variations in Performance on Adding Custom Vocabulary 39
4.5 Variations in Performance on Adding Attention Layer 40
4.6 Variations in Latency 41
4.7 Performance of the two variations of the proposed model on MSVD dataset 42
4.8 Encoder-Decoder Architecture - Caption Generation 43
4.9 Comparing CYF-KAL and CYF-CAL with Current State-of-the-art Models 48
4.10 Baseline model caption: A man is riding a bike Proposed model caption: a person is riding a motorcycle 49
4.11 MSVD video file: g36ho6UrBz0_5_20.avi CYF-KAL: a person is playing a guitar, Time taken: 0.57s CYF-CAL: a person is playing a guitar, Time taken: 0.55s 50
4.12 MSVD video file: fnpp8v9NbmY_181_188.avi CYF-KAL: a person is putting ingredients in a bowl, Time taken: 0.71s CYF-CAL: a person is mixing ingredients, Time taken: 0.50s . . . 50

4.13 MSVD video file: j2Dhf-xFUxU_13_20.avi CYF-KAL: a person is slicing a carrot, Time taken: 0.60s CYF-CAL: a person is slicing an, Time taken: 0.55s 51

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The transformation from text-based content to video-based content has been one of the most significant shifts in how we consume and communicate information in the digital age. The Cisco Annual Internet Report (2023) [35] predicted that video content would make up 82% of all internet traffic by 2024. Moreover, visuals are not only processed faster than text, but are also more engaging and impactful for the human brain, making it more efficient to communicate through video-based content [40] [41].

Unsurprisingly, this shift has been very noticeable in education where students are now more satisfied with videos for learning rather than the traditional textbook approach [12]. Faced with these facts, it is not at all unreasonable to assume that the amount of video-based content on the internet is only going to increase further hereon.

Real-time scene identification and description through video captioning has many advantages.

1. Content Discoverability: In an age of vast video content, efficient content indexing is critical. Real-time scene identification and description enable more accurate video indexing, making it easier to search for specific

content within videos, thus improving content discoverability and search engine optimization.

2. Accessibility and Inclusivity: There is a growing need for technology that enhances accessibility for individuals with visual impairments. By providing real-time textual descriptions of scenes and objects in videos, this project contributes to a more inclusive and accessible digital environment, enabling visually impaired individuals to better engage with video content.

3. Live Commentary: The project has the potential to result in satisfyingly accurate scene descriptions suitable for live commentary. This can be extremely useful while telecasting sports matches or in video games.

4. Human-Computer Interaction: Real-time video captioning has the potential to enhance human-computer interaction in virtual assistants and augmented reality systems as providing contextually relevant captions can improve the user experience. It can even broaden the reach of video content, and open up new avenues for industrial and market expansion.

5. Education: This technology can also benefit educational and informational video content. By generating accurate and timely descriptions, it facilitates a better understanding of the content, aiding in quicker learning.

In summary, real-time scene identification and description through video captioning addresses several pressing needs, including accessibility, content indexing, improved human-computer interaction, educational applications and commercial opportunities. It has the potential to benefit a wide range of users and industries while pushing the boundaries of technology and research.

1.2 BACKGROUND

Video captioning refers to the process of generating descriptions for videos in a readable text format. This requires identification of objects in the video and a general understanding of the scenes observed. Scene understanding refers to the ability of a computer to interpret the content of a visual scene. This involves tasks like scene segmentation (partitioning the scene into meaningful regions), object detection (identifying objects in the scene) and activity recognition (understanding the actions taking place). Hand-in-hand with scene understanding is the method of video captioning, which is the process of generating natural language descriptions of video content. Certain deep learning models are highly effective at extracting features from images and videos, and can therefore be used for scene understanding, while other architectures are extremely useful for caption generation.

Successful real-time video captioning can aid in various applications from improved speed and efficacy of crime detection and surveillance to enhancing gaming experience with dynamic context-aware narrations during gameplay or reality-television.

It is evident that with the sheer volume of video-based content in today's world, attempting to manually process and monitor videos on a large scale for any application is impractical. This highlights the necessity for an automated approach to keep track of the information communicated through videos. Thus, automatic scene understanding, and the generation of suitable captions, is found to be a crucial challenge in the world at present.

One key challenge is capturing the context and relationships between objects in a scene [10] as capturing the dynamics of a scene is crucial for a complete understanding [11]. This is made all the more difficult by partial occlusions and ambiguous lighting conditions that can make it almost impossible to accurately identify and segment objects [25]. Moreover, real-world scenes are often cluttered and obscure, requiring models to make sense of incomplete or conflicting information [19]. Extracting meaningful concepts from such complex visual data remains a challenge. Thus, despite significant progress, the task of scene understanding and video captioning faces several hurdles related to data complexity and model limitations. Furthermore, the challenge is intensified as with the growing prevalence of real-time multimedia content, scene identification and captioning must also be successfully executed in real-time.

Current deep learning approaches for video captioning face limitations beyond general scene understanding challenges. Existing methods often prioritize static object recognition, neglecting the crucial task of capturing the flow and sequence of actions within a video. Furthermore, pre-trained networks designed to extract object interactions struggle to capture the unique features that emerge from interactions between multiple objects. Similarly, models trained for human actions miss temporal information about non-human objects, hindering a comprehensive understanding of the scene. These limitations highlight the need for advancements in capturing action sequences, handling multi-object interactions, and effectively incorporating subtle visual details into video captioning models.

1.3 PROBLEM DEFINITION

The project aims to address the challenge of automatically and accurately identifying scenes in real-time video streams and generating descriptive captions.

This involves integrating computer vision techniques for scene recognition, with natural language processing methods, to provide real-time, contextually relevant textual descriptions for enhancing human-computer interaction in various applications.

The focus of the project shall lie in developing a working real-time video captioning model using deep learning techniques. This involves establishing a baseline video captioning model and improving upon various aspects to enhance the performance of the model with respect to accuracy and speed. Thus, the resultant model should be an integration of the best object detection, activity detection and sentence generation frameworks available as open-source. It also aims to reduce latency of the caption generation process without compromising accuracy, so as to enable its usage in real-time applications.

The expected outcomes are as follows:

- The development of a machine learning model that can generate descriptive captions for videos in real-time.
- The proposed object-event buffering mechanism should ensure prompt and accurate description of each scene.

- Improved accuracy and reduced latency in the caption generation process, as compared to the existing baseline model, which facilitates its real-time usage.

1.4 ORGANIZATION OF THE REPORT

The rest of the report is organised as follows. Chapter 2 presents a comprehensive literature survey and related work done in this domain. Chapter 3 provides a detailed description of the proposed system and outlines each step of the model development. Chapter 4 discusses specific implementation details such as the dataset used and working ecosystem. It also examines the results obtained and presents an extensive evaluation against the performance of currently existing state-of-the-art models, followed by a thorough qualitative analysis. Chapter 5 deliberates about the impact of this project work on our community, while Chapter 6 proceeds to conclude this report with suggestions on further improvements.

CHAPTER 2

LITERATURE SURVEY

In recent years, the field of video captioning has witnessed significant advancements, driven largely by the application of deep learning techniques. This survey seeks to provide an overview of key works in this domain.

2.1 RELATED WORK

2.1.1 Related Work on Surveying Deep Learning Approaches for Video Captioning

Rafiq et al. [23] presents an extensive examination of deep learning approaches in the context of video description. The study delves into several aspects, including benchmark datasets, evaluation metrics, architectural choices, and underlying

TABLE 2.1: Literature Survey 1 - Deep Learning Approaches for Video Captioning

Paper Title	**Discussion**	**Inference**
Video description: A comprehensive survey of deep learning approaches (2023) [23]	Benchmark datasets, Evaluation metrics, Architectures, Mechanisms, etc.	Various methodologies were studied
A Review of Deep Learning for Video Captioning (2023) [1]	Multiple Deep Learning approaches for various applications of Video Captioning	Research gaps and future scope were analysed

mechanisms. The work synthesizes findings from various methodologies, offering a comprehensive view of the current state of the field.

Meanwhile, Abdar et al. [1] explores a multitude of deep learning techniques employed in video captioning across various applications. The review encompasses a wide range of applications, highlighting the diversity of deep learning methods employed for video captioning. In addition to summarizing existing work, the paper identifies research gaps and outlines future research directions, providing valuable insights for researchers seeking to advance this area.

The aforementioned research works collectively offer a comprehensive overview of the application of deep learning in video captioning and is documented in Table 2.1. The advantages and disadvantages of various approaches are outlined, thereby helping researchers address the existing research gaps and work towards the future scope in this domain.

2.1.2 Related Work on Unique Approaches for Video Captioning

On top of this, work related to scene description using video captioning has been explored in existing literature. These studies demonstrate the wide-ranging applications of deep learning and multi-modal approaches in generating textual descriptions for video content, with varying degrees of success, and a summary of these approaches is provided in Tables 2.2 and 2.3. For instance, the framework in Saldanha et al. [24] involved keyframe extraction of the most relevant frames

in the given video. This is fed as input to the image captioning algorithm that generates a caption for every keyframe. Following this, a summarisation method is used to obtain a description of every scene in the given video. The system yielded METEOR (Metric for Evaluation of Translation with Explicit Ordering) scores of 0.46 and 0.134 on the MSVD (Microsoft Research Video Description Corpus) and MPII (Multi-Person Pose Estimation) datasets respectively. Note that this approach performs summarisation of an existing input video for generating captions and hence, this approach does not work for real-time scene description.

The research done by Shahir et al. [27] aimed at generating Bengali captions that could plausibly summarize a given video as well as identifying the best performing model for Bengali video captioning. The approach achieved impressive metrics, with BLEU4 (Best Linear Unbiased Estimator) scores of 0.434, METEOR scores of 0.391, and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores of 0.472, indicating its robustness in generating captions for Bengali video content. Although this approach does not work for real-time captioning, the captions generated are significantly accurate and thus, the use of an LSTM model can be considered for our own proposed work.

Exploring other possibilities, Iashin et al. [14] employs temporal event localisation using the Bidirectional Single-Stream Temporal action proposals network (Bi-SST). BiSST applies 3D Convolution network (C3D) to video frames and extracts features that are passed to a subsequent bi-directional LSTM network. The system demonstrated substantial success, achieving BLUE4 scores of 1.81 and METEOR scores of 10.09. Amirian et al. [3] details a proposed architecture for detecting physical activity. However, it did not provide for multi-object identification in each scene. While the specific results are not

TABLE 2.2: Literature Survey 2 - Summary of existing approaches for scene description

Paper Title	**Dataset**	**Methodology**	**Results**
Scene Description Using Keyframe Extraction and Image Captioning (2023) [24]	MSVD, MPII Movie Description Dataset	Keyframe extraction, image captioning; performed caption generation after summarisation	METEOR: 0.46, 0.134; summarisation not useful real-time
An Attention-based Hybrid Deep Learning Approach for Bengali Video Captioning (2023) [27]	Microsoft Research Video Description Corpus (MSVD) dataset	Attention, CNN: VGG19, RNN: LSTM	BLUE4: 0.434, METEOR: 0.391, ROUGE: O.472
Multi-modal Dense Video Captioning (2020) [14]	ActivityNet Captions dataset	ASR to detect human speech; Input subtitles + audio track + video frames	BLUE4: 1.81, METEOR: 10.09; does not provide full description
The Use of Video Captioning for Fostering Physical Activity (2020) [3]	*not mentioned*	Human activity detection using STEP framework. Object detection - YOLO. NLP module for correlating captions	Accurate for short videos; focused on human physical activity

detailed, the system was found to be accurate for short videos. Note that the system processes events, but does not sample the input video in real-time continuously.

There are several other deep learning approaches to video captioning that focus on improving a specific component of the captioning process. For instance, Xu et

al. [33] proposes the use of a modality-specific LSTM (MA-LSTM) to capture the important information in different modalities and was rewarded with a BLEU-4 score of 52.3 and a CIDER-D score of 70.4 on the MSVD dataset. Chen et al. [7] incorporates reinforcement learning to pick the informative frames from an input video and uses those frames alone for caption generation. This technique resulted in the METEOR score of 33.1 and CIDER score of 76.0 on the MSVD dataset. Further, an end-to-end training approach for video captioning is explored by Olivastri et al. [20]. This involves optimizing both the encoding and decoding parts of the system simultaneously in an end-to-end manner. The approach yielded good results with a ROUGE-L score of 60.6 and a CIDER score of 47.6. However, we notice that none of these techniques significantly improve the feature selection process for training the model.

On a slightly different tangent, some studies explore the importance of focusing on syntax and semantics for video captioning. Zheng et al. [34] introduces a syntax-aware module that observes the relationship among video objects and models it to syntax components for caption generation. The use of this technique boosted the CIDER score of the model to 81.0 on the MSVD dataset. Similarly, the domain-specific semantics identification approach used by Hemalatha and Sekhar [28] produced a model that achieved high performance on semantics-based video captioning metrics with a CIDEr score of 76.0 and ROUGE-L score of 73.1 on the MSVD dataset. Having said that, the proposed work here follows a lexical approach and therefore, more importance is given to the BLEU and METEOR metrics.

TABLE 2.3: Literature Survey 3 - Summary of existing approaches for scene description

Paper Title	**Dataset**	**Methodology**	**Results**
Learning Multimodal Attention LSTM Networks for Video Captioning [33]	MSVD, MSR-VTT	Modality-specific LSTM for different modalities	BLEU-4: 52.3, CIDER-D: 70.4; special treatment for different modalities slightly improves results
Less Is More: Picking Informative Frames for Video Captioning [7]	MSVD, MSR-VTT	Picking out informative frames alone using reinforcement learning	METEOR: 33.1, CIDER-D: 76.0; good performance despite selecting only few frames as input
End-to-End Video Captioning [20]	MSVD, MSR-VTT	End-to-end simultaneous training	CIDER: 86.6, ROUGE-L: 70.2; identified discriminate features required for caption generation
Syntax-Aware Action Targeting for Video Captioning [34]	MSVD, MSR-VTT	Modeling of caption syntax based on relationships of video objects	ROUGE-L: 69.4, CIDER: 81.0; syntax-aware caption generation yields higher semantic-based scores
Domain-Specific Semantics Guided Approach to Video Captioning [28]	MSVD, MSR-VTT	Differentiating semantics of videos on the basis of domains	METEOR: 34.7, CIDER: 76.0; does not perform as well as syntax-based approach

2.1.3 Related Work on Image Captioning Techniques that are Adaptable to Video Captioning

We also extend the literature survey in Table 2.4 to works that explore the development of custom image captioning models, anticipating that the procedure can be extrapolated to video captioning. One such work includes Al-Malla et al. [2] which presents an attention-based, encoder-decoder deep architecture that makes use of convolutional features extracted from a CNN (Convolutional Neural Network) model pre-trained on ImageNet (Xception), along with object features extracted from the YOLOv4 (You Only Look Once) model pre-trained on the MS COCO dataset. It is interesting that the outputs of these models are combined by concatenation to give rise to a feature map that can be used to generate more accurate descriptions. This feature extraction technique led to an improvement of 15% in the CIDER score on the MS COCO and Flickr30k datasets. The work also introduces a novel importance factor, which is a positional encoding scheme to help describe the most relevant events in an image. However, this can now be easily implemented with YOLOv8 which provides a cut-off based on a threshold confidence score.

Indeed, the use of YOLO for image captioning is being explored in current research. For example, Karve et al. [16] performed image captioning with an improvisation on the standard CNN, taking inspiration from the YOLO algorithm. The proposed YOLOv1-LSTM model was able to deliver faster output with good accuracy. Similarly, Vo-Ho et al. [32] developed an image captioning system that extracts object features from YOLO9000 and Faster R-CNN. From a given image, they extracted a list of tags using YOLO9000, created embeddings and used an

attention module to produce local features. The features are then together given as input to the LSTM model to generate probabilities of words and a beam search strategy is used to choose the best candidate for the caption. Another instance includes the work of Saroja and Mary [26] in image captioning using an improved YOLOv5 and Xception V3 on the Flickr 8k, Flicr30k and MS COCO (Microsoft Common Objects in Context) datasets. The method achieved 99.5% accuracy, 99.1% precision, 99.3% recall and 99.4% F1-score on the MS COCO dataset.

In a similar vein, a significant amount of work was done by Preethi and Dhanalakshmi [22] in video captioning using a pre-trained CNN and LSTM, i.e., following the two stages of feature extraction and caption generation. Inception V3 and VGG16 CNN models were used for extracting the features from the video, and caption generation is then done using LSTM with the help of the extracted features. However, the work still faces some limitations on the speed of output generation and inaccurate detection of objects in the frame.

Thus, the survey provides us with a comprehensive overview of current research in this field and helps us identify any scope for improvement. As is evident from this review, techniques for optimizing the two parts of the captioning process separately, and later combining them to incorporate the best of both tasks, is not yet popular for video captioning. This research direction is the core focus of our proposed work.

The research gaps identified from the literature survey are discussed in detail in the next section.

TABLE 2.4: Literature Survey 4 - Summary of existing approaches for scene description

Paper Title	**Dataset**	**Methodology**	**Results**
Image captioning model using attention and object features to mimic human image understanding (2022) [2]	MS COCO and Flickr30k datasets	Concatenate CNN and YOLOv4 output features + Positional encoding	CIDEr score increase by 15.04%
Conversational Image Captioning Using LSTM and YOLO for Visually Impaired (2022) [16]	*not mentioned*	Improvised CNN with inspiration from YOLO for detection + LSTM for caption generation	Faster output, good accuracy compared to other state-of-the-art
A Smart System for Text-Lifelog Generation from Wearable Cameras in Smart Environment Using Concept-Augmented Image Captioning with Modified Beam Search Strategy (2019) [32]	MS COCO dataset	Object feature extraction with YOLO9000 and Faster R-CNN + Beam search strategy LSTM	0.11 increase in CIDER score; 9000 categories identified
Image Captioning Using Improved YOLO V5 Model and Xception V3 Model (2023) [26]	Flickr8k, Flicr30k and MS COCO datasets	Improved YOLOv5 + Xception V3 models	99.4% F1-score on MS COCO dataset
Video Captioning using Pre-Trained CNN and LSTM (2023) [22]	*not mentioned*	Inception V3 + VGG16 CNN for feature extraction + LSTM	Satisfactory caption generation; can be enhanced

2.2 RESEARCH GAPS IDENTIFIED

From the literature survey, we identify that video captioning methods are still not effective at capturing sequences of actions (as opposed to object recognition). Further, pre-trained networks for extracting object interactions cannot derive discriminant multi-object features. Models trained for recognizing human actions fail to capture temporal information in other non-human objects in the scene. Moreover, object occlusions and video boundaries make vision-to-language translation difficult. Training with subtle fine-grained visual attributes is difficult.

2.3 RESEARCH OBJECTIVES

The research objectives of our project are thus as follows:

- To adapt and employ successful image captioning techniques to the task of video captioning.
- To explore several variations of a video captioning model after introducing different techniques.
- To extensively compare the performance of varied approaches to video captioning.
- To improve the performance of video captioning models significantly using open-source resources alone.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 OVERVIEW

Our proposed system for real-time scene identification and description is depicted in Figure 3.1. It combines computer vision and NLP (Natural Language Processing) techniques to enable real-time scene identification and description through video captioning. It is an encoder-decoder architecture with the visual characteristics from the video being encoded as features that are used by the LSTM (Long Short-Term Memory) to generate captions. The novelty proposed aims to address the research gaps of describing both objects and activities in each scene, as well as provides for recognising and describing multiple objects and events in each scene. This includes:

- Real-time description for each scene through our object-event buffering structure. This is achieved by video segmentation and consequent buffering of frames to compile context vectors in each segment, that ensures an activity is detected and identified as such.
- Feature fusion of the standard image features with the object detection features retrieved for every set of frames in a given input video stream. The combined features are then used to generate the captions. This is believed to add efficiency and accuracy in recognising multiple objects in the scene and also derive better temporal inferences between those objects.

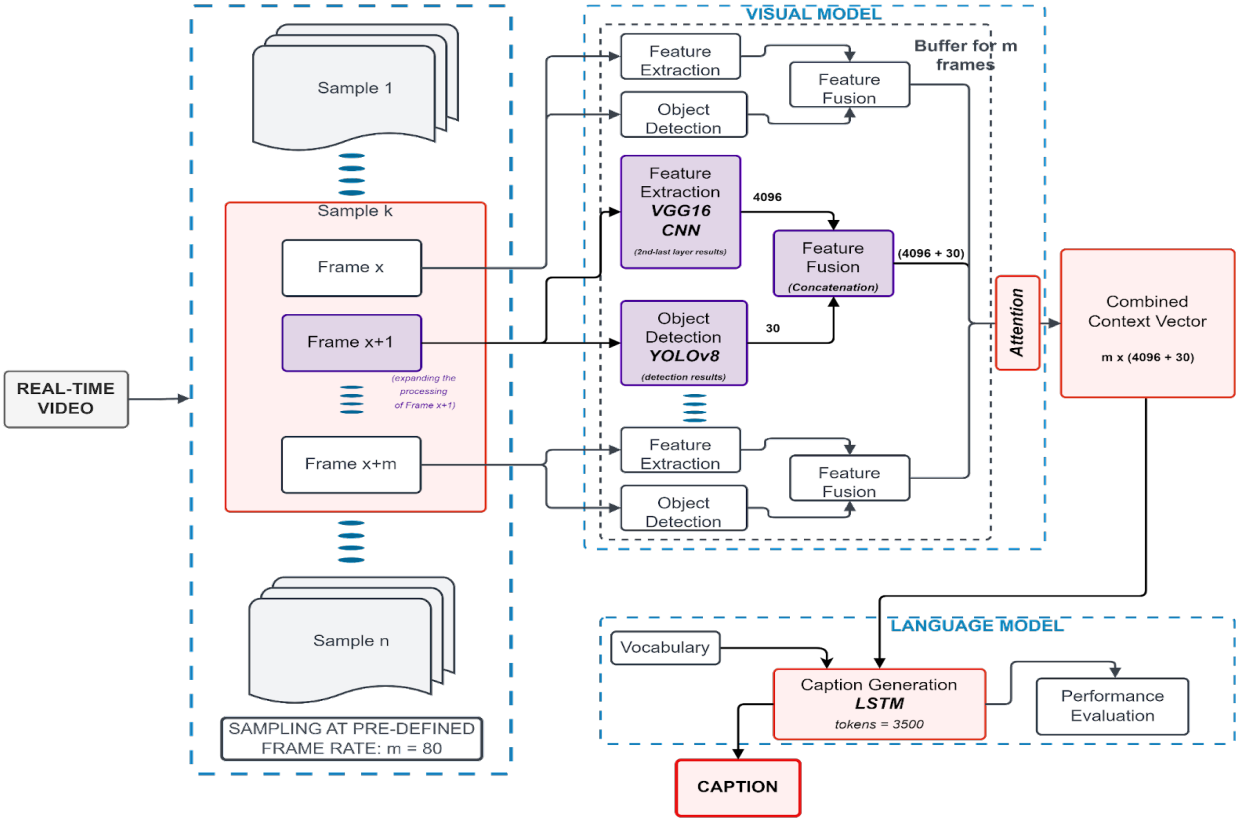

FIGURE 3.1: Architecture of the Proposed System

- A custom attention mechanism that helps the encoder module in processing different sections of the feature set with a relevant level of focus, and thereby enhance its visual understanding of the scene. This is elaborated under section 4.4.

We theorise that extracting features like shapes, textures, corners, contours, etc., as well as information of objects detected in each frame will improve comprehension. Therefore, to efficiently extract features for real-time processing, we leverage two pre-trained models: VGG16 (Visual Geometry Group) CNN [29] (on ImageNet [9]) for spatial features due to its reduced overfitting with a smaller convolutional filter, and YOLOv8 [15] (on MS-COCO [17]) for object detection due to its speed and accuracy. We utilize the penultimate (fully connected) layer of VGG16 and

the final detection output of YOLOv8 to create a comprehensive context vector for activity recognition in each video segment.

Meanwhile, for text generation, we adopt a sequence-to-sequence architecture with LSTM (Long Short-Term Memory) [13] cells. The encoder architecture imposes the already detected object information on the extracted spatial features, and uses them to derive better temporal inferences across the multiple frames in a video segment. To support the feature fusion, a trainable attention layer is also introduced. The model is trained on the context vector from the encoder as well as a generated vocabulary set from the training labels. Thus, a suitable caption is generated for each video segment in real-time.

The algorithm for the working of this structure is presented in Algorithm 1.

Algorithm 1 Algorithm for Processing and Captioning each Video Segment

Input: Predefined number of m frames from each video segment *video*, vocabulary *vocab* for the caption generation model, predefined number of decoder tokens l.

Output: Caption $caption$ for the video segment.

$video\_features \leftarrow \{\}$ ▷ Empty array of size m.
for each frame $frame$ in *video* **do**
    $cnn\_f \leftarrow$ extract_cnn_features(frame) ▷ Where cnn_f holds the CNN features extracted from $frame$ and $len(cnn\_f) = c$ (predefined).
    $yolo\_f \leftarrow$ extract_yolo_features(frame) ▷ Where $yolo\_f$ holds the YOLOv8 features extracted from $frame$ and $len(yolo\_f) = y$ (predefined).
    $combined\_f \leftarrow$ concatenate_features(cnn_f, yolo_f) ▷ Where $combined\_f$ holds the concatenated features extracted from $frame$ and $n = len(combined\_f) = c + y$.
    $video\_features$.append($combined\_f$).
end for
$caption \leftarrow$ lstm(attention, video_features, vocab, time_steps_encoder=m, num_encoder_tokens=n, num_decoder_tokens=l) .

return $caption$.

3.2 DESCRIPTION OF COMPONENTS

3.2.1 Video Stream Processing

1. **Sampling Rate:** The system utilizes a predefined sampling rate to extract frames from the video stream. This rate ensures that we capture representative frames for analysis.

2. **Video Segmentation:** The input video stream is split into smaller sample videos that are processed individually. This is done to ensure reduction of delay in real-time video captioning.

3. **Frame Extraction:** Each sample is further broken down into frames. A predefined 'm' number of frames are retrieved from each sample and given as input to the feature extraction unit.

3.2.2 Visual Model

1. **Feature Extraction:** Features such as pixels, shapes, textures, corners, contours, etc. are extracted from each frame using a Convolutional Neural Network (CNN) model. These features reveal hidden patterns in the images that can be used to identify objects and activities in the frames.

2. **Object Identification:** A computer vision technique for object detection called YOLOv8 is additionally employed to identify and categorize objects within the frames. Being notoriously famous for its unique detection head, the YOLOv8 offers the best speed and accuracy today when it comes to classifying objects in the scene.

3. **Activity Identification:** To detect an activity within the current sample, the system enters a buffering mode. It continues to observe subsequent frames up to the end of the sample duration, after which the features from all the samples are together collected and placed into the feature space.

4. **Feature Fusion:** The visual encoding information from the above steps are fused together to form a customised combined feature set. This is fed as input into the encoder-decoder architecture with the intention of allowing the LSTM encoder to impose the already detected object information on the extracted spatial features, and use them to derive better temporal inferences across the multiple frames in a video segment.

5. **Attention Mechanism**: To support the feature fusion in the visual model, a trainable attention layer is also introduced before the encoder LSTM layer to better capture the significance of the different kinds of features.

3.2.3 Language Model

1. **Live Captioning:** Leveraging Natural Language Processing (NLP) techniques, a suitable caption is generated for each video segment in real-time. The language model is trained on the context vector from the encoder as well as a generated vocabulary set from the training labels.

In summary, our proposed system combines computer vision and NLP techniques to enable real-time scene identification and description through video captioning. By carefully avoiding potential pitfalls, we aim to deliver a robust and effective solution for enhancing the understanding and accessibility of video content.

3.3 MODEL SELECTION

We require the use of two pre-trained models for our visual model. The first is a feature extraction model to identify activities in the video based on the image features extracted from each frame. The second is an object detection model that identifies objects in each frame.

For feature extraction, we implemented the VGG16 (Visual Geometry Group) CNN [29] model, pre-trained on the ImageNet Dataset [9], a dataset containing more than 14 million training images across 1000 object classes. It is a convolution neural network (CNN) model supporting 16 layers. The VGG16 model can achieve a test accuracy of 92.7% in ImageNet. It is one of the top models from the ILSVRC-2014 competition. VGG uses a smaller convolutional filter, which reduces the network's tendency to over-fit during training exercises. A 3×3 filter is the optimal size because a smaller size cannot capture left-right and up-down information. Thus, VGG is the smallest possible model to understand an image's spatial features. Consistent 3×3 convolutions make the network easy to manage. The architecture of VGG16 is outlined in Figure 3.2. To obtain only the image features from the model, we make use of the output of the second-last layer,i.e., the fully connected layer.

For object detection, we implemented the YOLOv8 [15] model which is pre-trained on the MS-COCO dataset [17]. It contains 80 class labels for object recognition. The model is well-known for its fast execution and excellent accuracy. In our implementation, we utilise the final detection output of the complete YOLO model to get the details of the objects identified in each image.

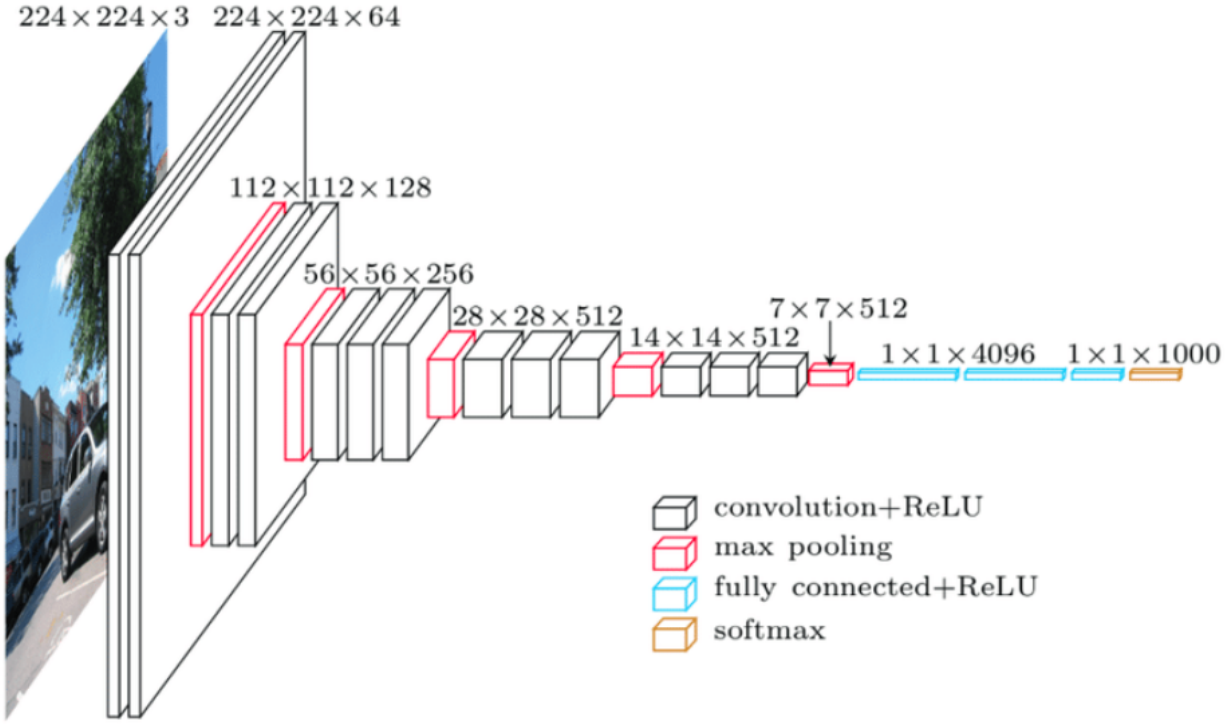

FIGURE 3.2: VGG16 CNN Architecture. Features are extracted from the final fully connected layer. [42]

Since our problem statement also requires text generation, we have adopted a sequence-to-sequence architecture, with LSTM cells for both the encoder and the decoder components as the language model. A single LSTM cell is depicted in Figure 3.3. An LSTM cell is used for each frame in the video segment and together, the mechanism is capable of processing an entire video segment. The encoder architecture imposes the already detected object information on the extracted spatial features, and uses them to derive better temporal inferences across the multiple frames in a video segment. To support the feature fusion, two variations of a trainable attention layer are also introduced. The model is trained on the context vector from the encoder as well as a generated vocabulary set from the training labels. Through this process, a suitable caption is generated for each video segment in real-time.

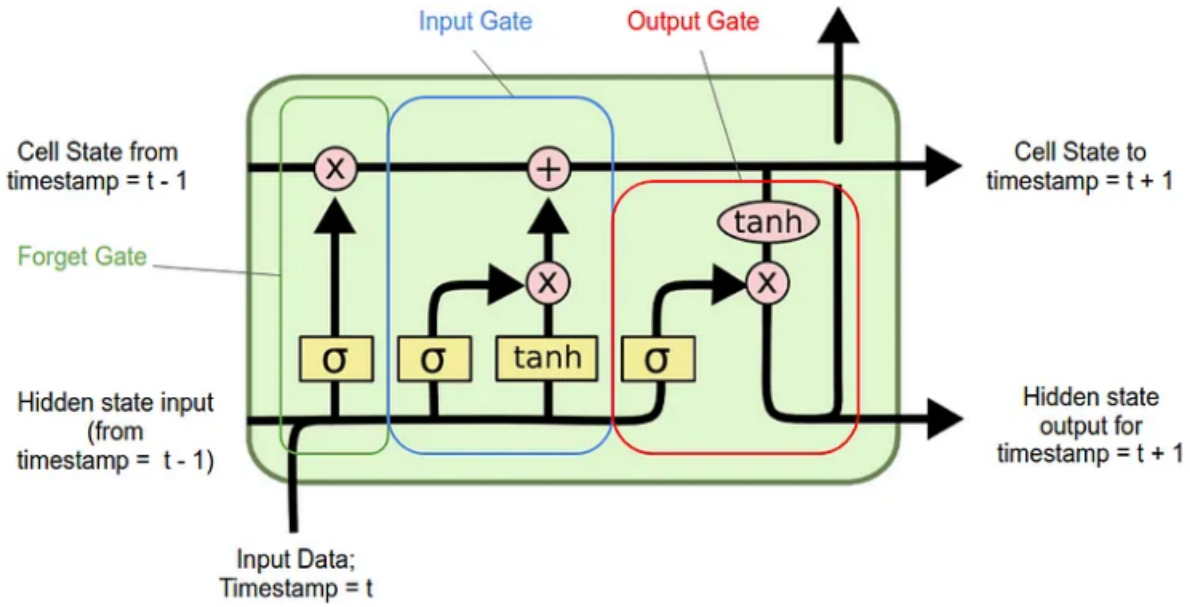

FIGURE 3.3: A single LSTM cell [39]

Thus, VGG16 CNN and YOLOv8 are chosen for extracting features from each frame in the video segment while an LSTM is chosen for caption generation.

3.4 FEATURE EXTRACTION

The complete feature extraction process is outlined in Figure 3.4.

Frame Extraction: The videos from the corpus are processed one by one - each video is split into frames, out of which 80 frames are sampled per video and stored in a temporary folder.

VGG16 CNN: The frames in this temporary folder are then passed to the VGG16 CNN model. The output of the final fully connected layer, a feature array of size (1,4096) is retrieved for each frame. Thus, for the sample of 80 frames, a NumPy

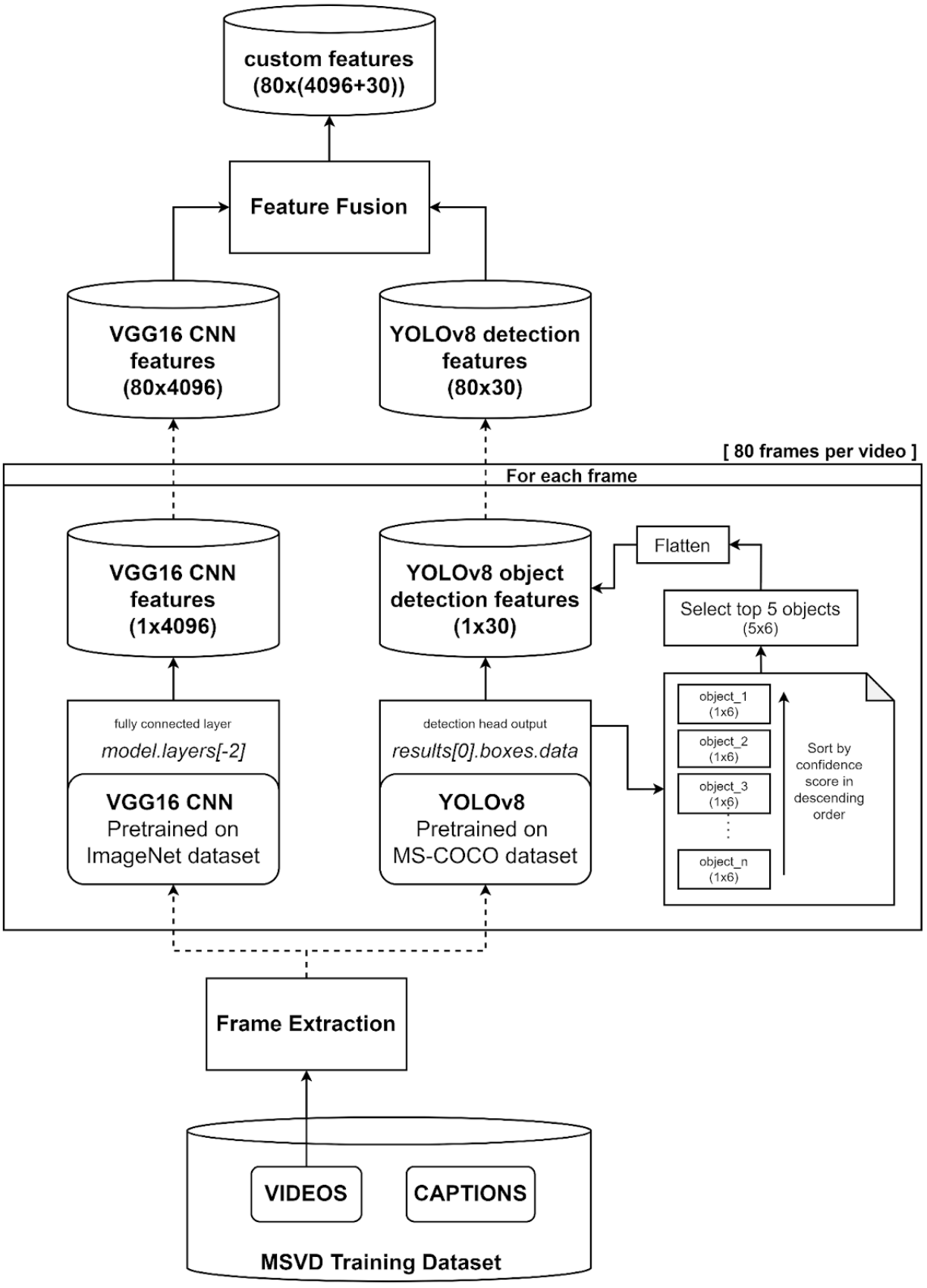

FIGURE 3.4: Outline of the implemented Feature Extraction

feature array of size (80,4096) is generated. Here, 80 is the number of features (frames) per video, and 4096 is the number of features per frame.

YOLOv8: The frames in the temporary folder are also simultaneously passed as input to the YOLOv8 model. The detection output of the model is automatically stored in its results object. For each frame, the detection data is as shown in Figure 3.5.

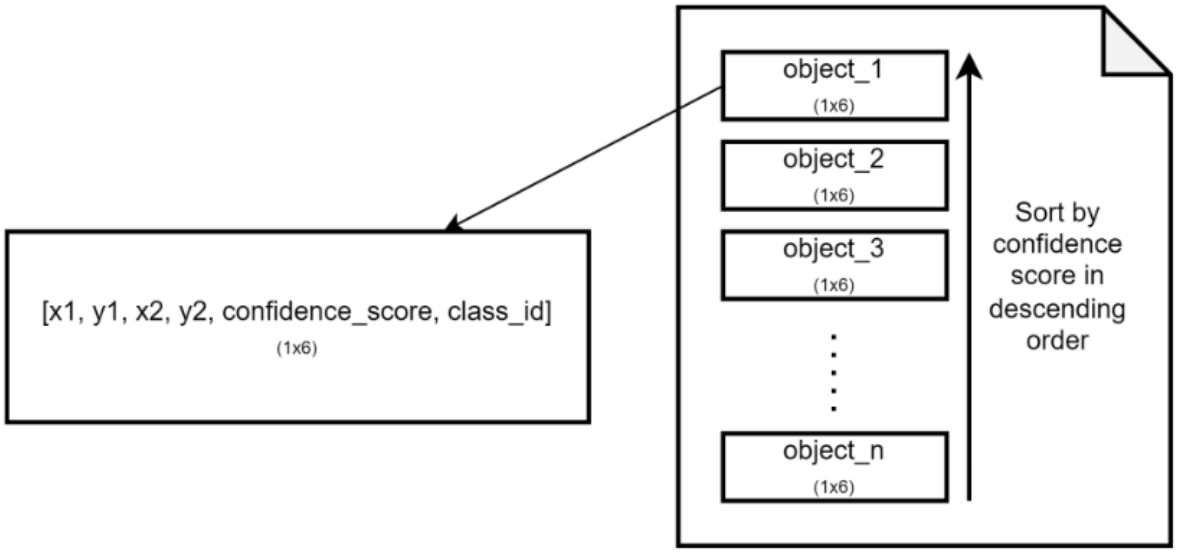

FIGURE 3.5: YOLOv8 result format for an image. (x1,y1) and (x2, y2) represent the coordinates of the top-left and bottom-right corners of the bounding box of the detected object. The confidence score measures how confident the model is of its prediction of that object. The class_id is one of 80 classes (0-79).

Each object has a corresponding feature array of size (1,6). If less than 5 objects are detected in a frame, then the array is padded with zeros (1,6) for normalisation. To avoid correlating the null zeros with the detected class_id 0 (represents ”person” in the MS-COCO dataset), all instances of class_id 0 are rewritten as class_id 80. Thus, the classes are now given by the range 1-80 instead of 0-79.

The detected objects are sorted according to their confidence scores in descending order and the top 5 objects in each frame are selected to form a feature array of size

(5,6). This is then flattened to form a feature array of size (1,30) for the current frame. Thus, for the sample of 80 frames, a NumPy feature array of size (80,30) is generated using YOLOv8. We have stored all the features in Google Drive as shown in Figure 3.6.

Feature Fusion: The VGG16 CNN and YOLOv8 features are now concatenated to form a feature array of (80,4096+30) for each sample. We believe that the LSTM will be able to read this concatenated features, and be able to distinguish between the CNN output and YOLO output based on the differences in the numerical scale and distribution. To further help the LSTM use that object information to derive a better understanding, we have employed attention layers to weigh the importance of different parts of the input sequence dynamically, and enhance its processing.

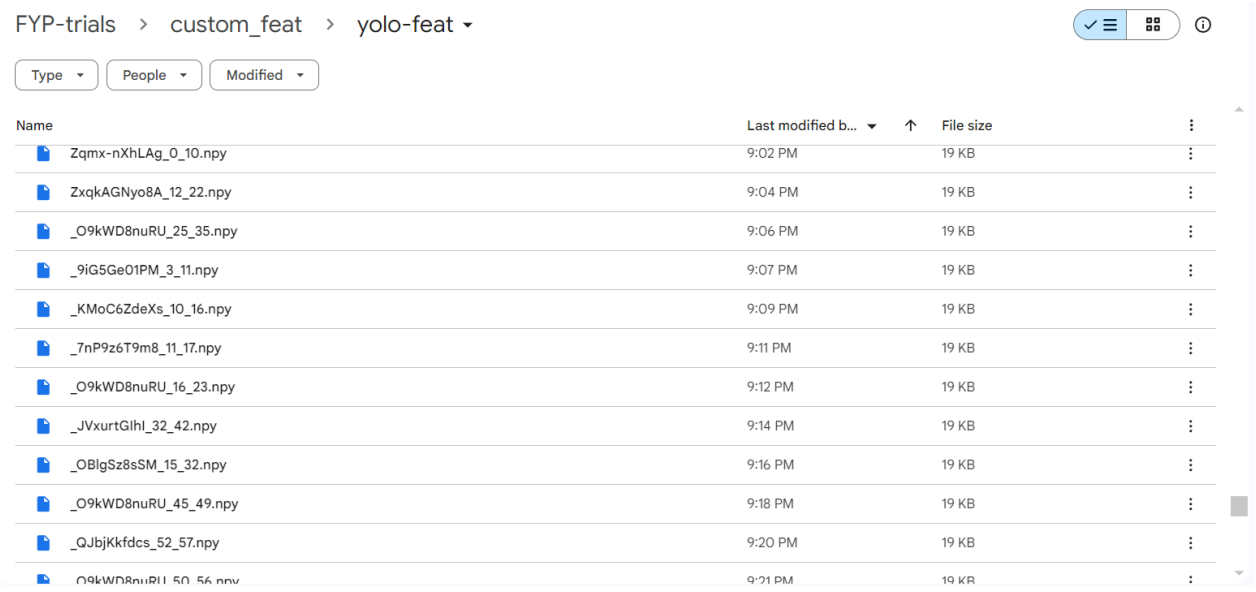

FIGURE 3.6: A screenshot of some of the extracted feature files

Thus, feature extraction is completed for all the videos in the given dataset.

3.5 BUILDING THE MODEL

The model is trained on the concatenated features and the refined captions. The basic structure consists of an encoder and decoder, upon which variations are introduced and experimented with, to arrive at the model giving the best performance on the validation dataset. All training is done on the free version of Tesla T4 GPU provided by Google Colab.

3.5.1 Attention Layer

Despite taking in the concatenated feature set as input, the LSTM cells lack capability to inherently differentiate between the features extracted from the VGG16 model and the object detection information as outputted by YOLOv8. To encourage the LSTM to not only distinguish, but also correlate between the two types of information and use it to its advantage, an attention layer is employed before feeding the feature set to the LSTM encoder.

Two types of attention layers are experimented with to reach the above subgoal:

1. **Pre-existing Attention Layer from Ternsorflow's Keras API**: A Luong-style Attention layer [18] (Keras [37]) is employed to capitalize on the trainable weights created by the layers. The CNN feature matrix of shape (80,4096) and the YOLOv8 output of shape (80,30) form the query tensors, while the combined feature matrix of shape (1,80,4126) is passed in as the value tensor. Attention weights are computed separately for each query tensor and combined to create a unified attention distribution. The

output of the Attention() layer is a tensor of the same shape as the query, and represents its attended version that highlights it in the feature set.

2. **Custom Attention Layer**: A custom attention mechanism is developed for our specific use case - highlighting different sections of the input feature so as to process them with different levels of focus as per their underlying meaning.

   (a) **Initialisation:** The layer is initialized with the number of categories and the number of time steps considered in the input feature set. Here, we group the input sequence into four static categories. Category 1 represents the 2D CNN features as extracted by the VGG16 layer. Category 2 represents the coordinates of the detected objects as outputted by the YOLOv8 model. Category 3 represents the class labels of these detected objects with values in the range of 1-80, where each integer represents an object class, and Category 4 represents the confidence scores of the detected objects. Here, a value of 80 is adopted for the number of timesteps, to align with the number of frames considered in each video segment.

   (b) **Building the layer:** Inside the build method, weights are initialized for each category as described above. These weights are learned during training to determine the importance of each category in generating the output.

   (c) **Forward Pass:** The call method takes in the input tensor and performs the following operations:

      - Splitting of the input sequence into four separate categories as described in the initialisation phase.

- In each category, the corresponding weight that is learned during training, is applied to the inputs. These weights are concatenated to form a unified representation.
- Attention scores are calculated by summing the weighted inputs across categories and a softmax activation is applied to obtain attention weights, determining the importance of each token.
- Finally, the attention weights are applied to the input tensor, and this helps to aid the model in processing different input sequences with a relevant level of focus.

3.5.2 LSTM Encoder

The input layer of the LSTM is customized to (1,80,4126) in order to pass the combined set of attended features as input to the LSTM encoder. Our Encoder architecture currently consists of 80 LSTM cells, with each cell taking in the fused feature set of one frame. The combined feature vector of the first frame are passed into the 1st cell and so on up until the 80th frame, as described in Figure 3.7. The Encoder learns this representation and generates a context vector of uniform length and sequence, that can be fed into the Decoder architecture. Thus, the final state of the Encoder LSTM acts as the initial state for the Decoder LSTM, and all the intermediary outputs are discarded.

3.5.3 LSTM Decoder

The decoder takes as input, the final state of the encoder (the context vector), along with the caption on which it is to get trained. In the first decoder cell ⟨bos⟩acts as input to start the sentence. This is followed by feeding in each and every word of the caption from the training data, until ⟨eos⟩. is reached.

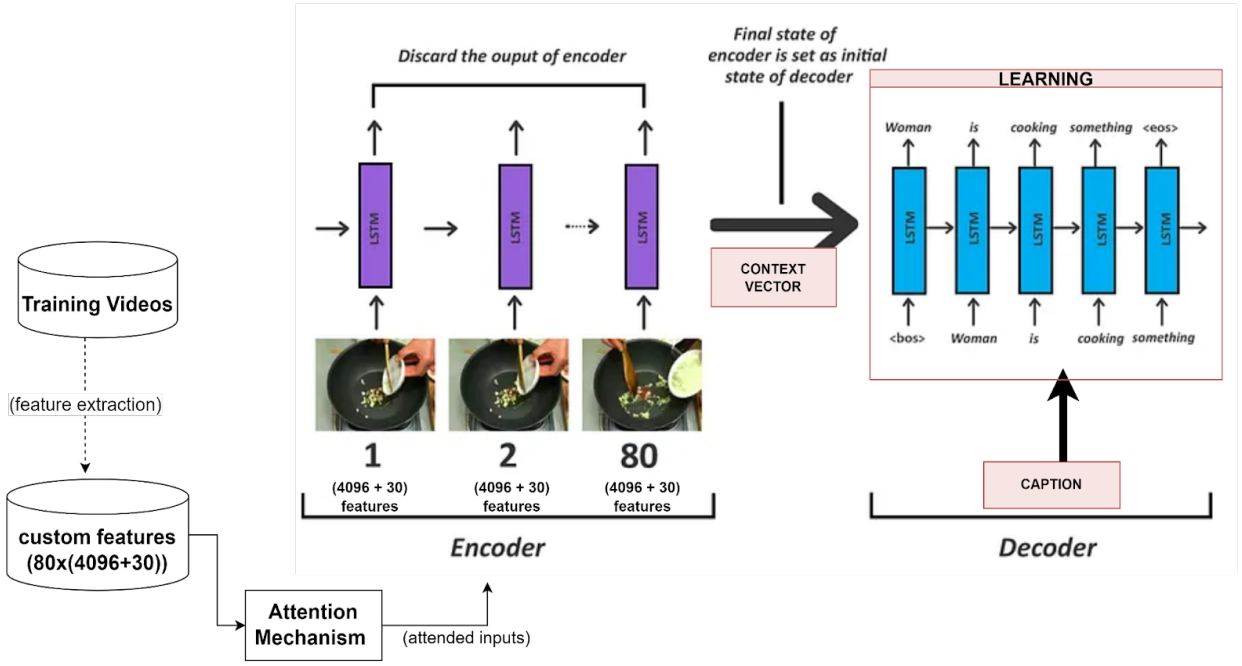

FIGURE 3.7: Encoder-Decoder Architecture - TRAINING

The model is trained for 120 epochs on the colab free-version of Tesla T4 GPU [5], over a period of 2 days. 120 epochs is chosen as it is the minimum number of epochs necessary to properly train the model. Early stopping is also employed so that the training is stopped automatically when the validation loss does not improve over the epochs. A screenshot of the model during training is shown in Figure 3.8.

```
Epoch 91/120
23/23 [==============================] - 26s 1s/step - loss: 0.6475 - accuracy: 0.8058 - val_loss: 1.1811 - val_accuracy: 0.7404
Epoch 92/120
23/23 [==============================] - 26s 1s/step - loss: 0.6415 - accuracy: 0.8081 - val_loss: 1.1721 - val_accuracy: 0.7407
Epoch 93/120
23/23 [==============================] - 26s 1s/step - loss: 0.6360 - accuracy: 0.8090 - val_loss: 1.1658 - val_accuracy: 0.7403
Epoch 94/120
23/23 [==============================] - 26s 1s/step - loss: 0.6315 - accuracy: 0.8107 - val_loss: 1.1782 - val_accuracy: 0.7394
Epoch 95/120
23/23 [==============================] - 26s 1s/step - loss: 0.6277 - accuracy: 0.8103 - val_loss: 1.1811 - val_accuracy: 0.7393
Epoch 96/120
23/23 [==============================] - 26s 1s/step - loss: 0.6238 - accuracy: 0.8107 - val_loss: 1.1677 - val_accuracy: 0.7409
Epoch 97/120
23/23 [==============================] - 26s 1s/step - loss: 0.6199 - accuracy: 0.8114 - val_loss: 1.1690 - val_accuracy: 0.7372
Epoch 97: early stopping
```

FIGURE 3.8: A snapshot of the model getting trained

Figure 3.9 shows the output after training of the model is complete.

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 80, 4126)]        0

 endcoder_lstm (LSTM)        [(None, 80, 512),         9500672
                              (None, 512),
                              (None, 512)]

=================================================================
Total params: 9500672 (36.24 MB)
Trainable params: 9500672 (36.24 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Model: "model_2"
_____________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=============================================================================================
 decoder_inputs (InputLayer  [(None, 10, 1500)]        0         []
 )

 input_3 (InputLayer)        [(None, 512)]             0         []

 input_4 (InputLayer)        [(None, 512)]             0         []

 decoder_lstm (LSTM)         [(None, 10, 512),         4122624   ['decoder_inputs[0][0]',
                              (None, 512),                        'input_3[0][0]',
                              (None, 512)]                        'input_4[0][0]']

 decoder_relu (Dense)        (None, 10, 1500)          769500    ['decoder_lstm[2][0]']

=============================================================================================
Total params: 4892124 (18.66 MB)
Trainable params: 4892124 (18.66 MB)
Non-trainable params: 0 (0.00 Byte)
```

FIGURE 3.9: Trained model

The trained model is then evaluated as described in the next section.

CHAPTER 4

EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

4.1 DATASET

4.1.1 Dataset Description

The Microsoft Research Video Description Corpus (MSVD) [6] is a collection of 1970 videos, each video depicting a single activity that lasts for about 6 to 25 seconds. For example, the video may be a small clip of someone cooking, of a man riding a bike, or of a baby playing. Each video clip has multiple associated captions across multiple languages featuring over 120,000 sentences with an average of 15 English captions per clip. Following the evaluation setup of prior work [31], we split the dataset into three parts: 1,200 videos for training, 100 for validation, and 670 clips for testing.

4.1.2 Caption Refinement

On observing the captions in the dataset, we noticed that the frequency of important words in the captions were being reduced due in part to the following reasons:

- Several synonyms are present for a word in the original captions. These contribute to splitting the preference for words that have the same meaning. Therefore, we chose some important recurring words and replaced all instances of their synonyms with them. Some examples are shown in Table 4.1.

- The captions are in sentence case, which means that ”Person”, ”person” and ”person.” would be considered different words. As it is not necessary for us to display the captions in a sentence format, we have decided to convert the captions to lower case and remove the full stops.

- Important words such as articles (a, an, the), verbs (”ing”-words), popular nouns (cross-verified with the MS-COCO class labels) and number words (one, two, three, ...) are picked out from the compiled list of unique words in the captions to form the customised initial vocabulary which will later be used for training the model.

TABLE 4.1: Examples for Replacing Synonyms

Synonyms	**Replaced by**
infant, newborn	baby
huge, large	big
individual, lad, man/woman, male/female, lady	person
many, a number of	several

4.1.3 Caption Preprocessing

The steps are outlined in Figure 4.1. Only the refined captions of sentences of length 6-10 words are retained. They are tokenized and padded to a uniform

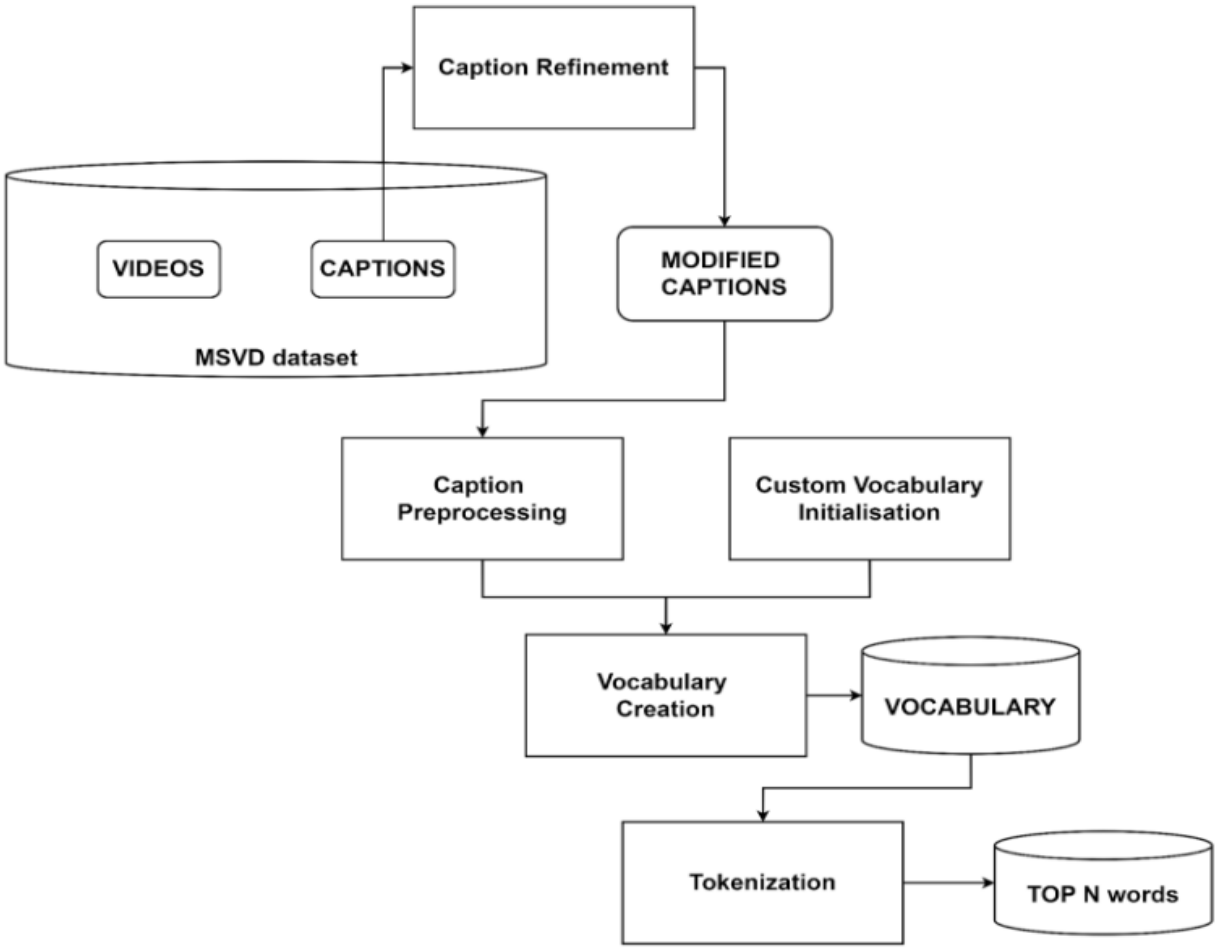

FIGURE 4.1: Steps in Caption Preprocessing

length of 10 words. Each caption phrase is wrapped within the delimiters ⟨bos ⟩and ⟨eos ⟩. A Keras Tokenizer [38] is used to create a vocabulary comprising a unique set of most frequently used words in the captions. The custom vocabulary list created during caption refinement is optionally added as a prefix to the new vocabulary. Combined, the top N (a predefined number of decoder tokens decided after experimentation) words are identified from the vocabulary list. This is later fed into the decoder for caption generation. Finally, the captions are also tokenized and padded to arrive at a uniform length of 10 words per caption.

4.2 ECOSYSTEM

All models were initially trained on a subset of the MSVD Dataset for internal comparison, which was carried out on the Tesla T4 GPU offered by Google Colaboratory. The top 2 models were handpicked from these based on their performance and trained on the complete MSVD Dataset, for which the TPU v2 hardware accelerator offered by Google Colaboratory was taken advantage of.

4.3 EXPERIMENTS CONDUCTED

4.3.1 Evaluation of Model Variations

The original baseline model [36] for video captioning incorporates caption generation based on only the CNN features extracted from the video. Hypothesizing that the CNN features are not sufficient information for good video captioning, we experiment with variations of the model involving feature fusion with YOLOv8 detection results, prefixing a custom vocabulary and the addition of an attention layer before the LSTM encoder. We also vary the size of the vocabulary generated from the training labels, to arrive at the optimal decoder for the LSTM.

Note that due to the heavy computational requirements of training, these experiments are all carried out on a subset of the MSVD dataset with 755 videos used for training and 416 videos used for testing (following a similar train:test ratio as in the MSVD dataset - 1200:670::755:416).

The results of our experimentation are presented in Table 4.2.

TABLE 4.2: Evaluation of Model Variations. T-A:Training Accuracy(%), V-A:Validation Accuracy(%), B:BLEU-4(%), M-METEOR(%), R:ROUGE-L(%), C:CIDER(%), T:Average time taken for caption generation(in seconds)

MODEL	**EPOCHS = 120**	**T-A**	**V-A**	**B**	**M**	**R**	**C**	**T**
Baseline								
CNN	early stopping at 70	75.87	70.79	53.1	33.4	50.2	27.6	1.21
Yolo+CNN (without custom vocab)								
1500 tokens	early stopping at 91	80.88	73.49	70.9	51.6	60.8	31.4	0.66
2000 tokens	early stopping at 73	77.83	73.02	70.6	51.5	60.5	31.7	1.18
2500 tokens	early stopping at 66	75.78	71.70	70.8	51.2	59.6	31.9	0.84
Yolo+CNN (with custom vocab)								
1500 tokens	early stopping at 74	78.78	73.52	66.5	49.3	60.6	31.6	1.23
Yolo+CNN - 1500 tokens + attention layer								
Keras Attention layer	early stopping at 97	81.14	73.72	72.6	52.9	61.2	31.4	0.75
Custom Attention layer	early stopping at 78	79.05	74.26	71.8	51.8	60.3	31.3	0.65

1. **Variation in the Vocabulary Size:** The vocabulary size is varied from 1500 to 2500 in steps of 500, and the outcome is depicted in Figures 4.2 and 4.3. It is clear that increasing the number of tokens does not improve the

performance of the model. Moreover, the model trained with 1500 tokens also shows higher training accuracy and validation accuracy, and is therefore adopted as the base model for future experimentation.

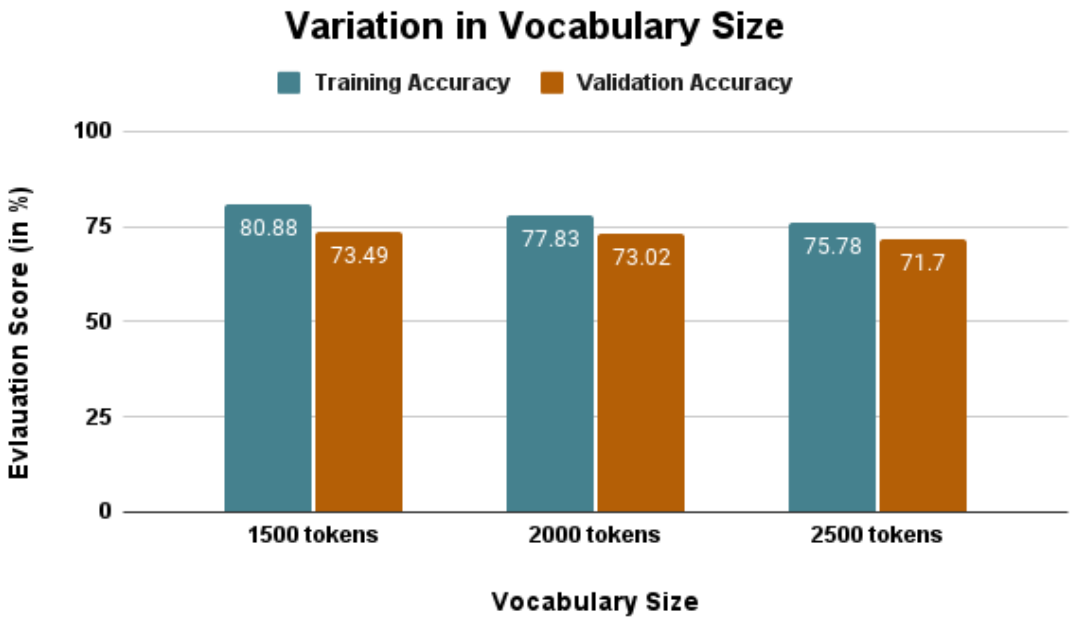

FIGURE 4.2: Variations in Accuracy on Varying Vocabulary Size

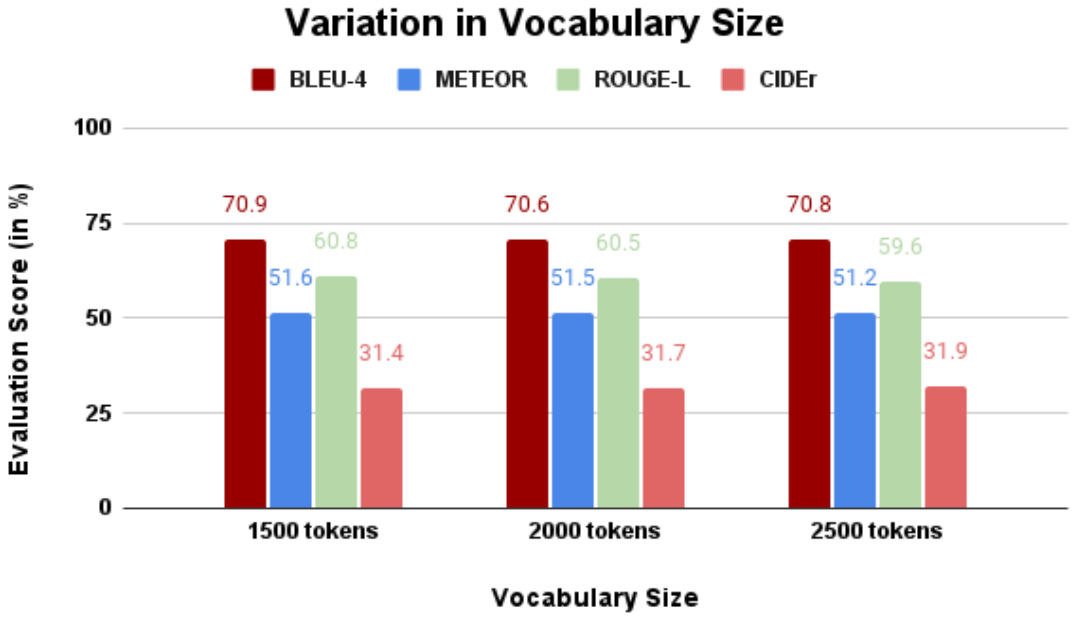

FIGURE 4.3: Variations in Performance on Varying Vocabulary Size

2. **Initialisation with a Custom Vocabulary:** The custom vocabulary mentioned in Section 4.1.2 is prefixed to the vocabulary derived from tokenizing the training captions. However, this customization does not prove useful. Rather, it is evident from Figure 4.4 that it slightly worsens the performance of the model. Thus, this technique is abandoned.

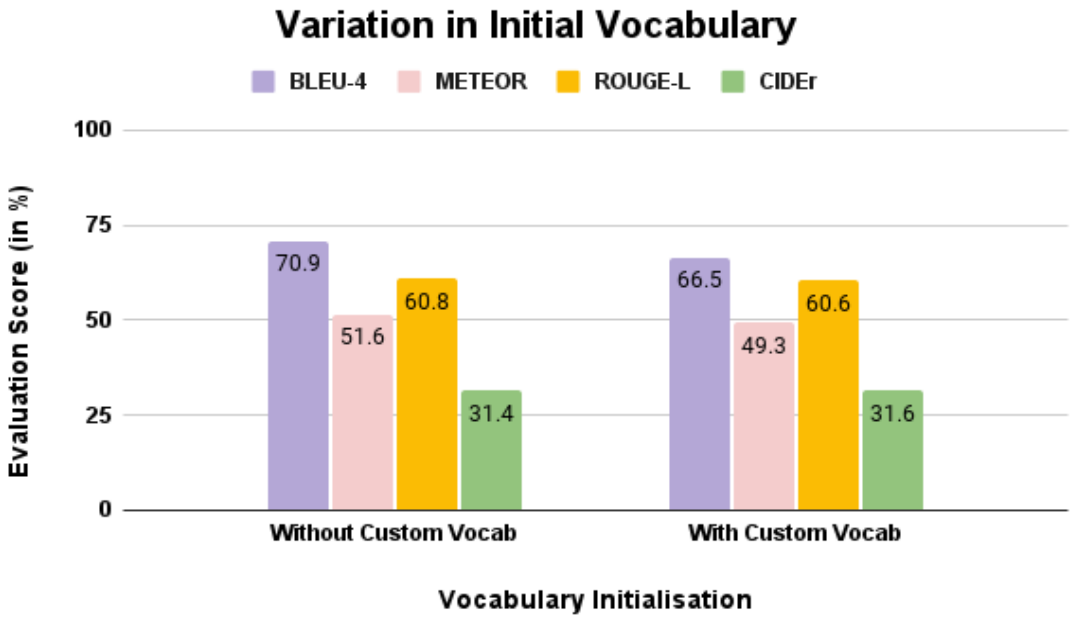

FIGURE 4.4: Variations in Performance on Adding Custom Vocabulary

3. **Addition of an Attention Layer:** Considering the model after incorporating feature fusion with 1500 tokens as the new base model, an attention layer is added in each of the two ways outlined in Section 3.5.1. The scores suggest that while both techniques for attention perform similarly, with the in-built attention layer from Keras API only slightly outperforming our custom-weighted attention layer, the addition of an attention mechanism itself significantly boosts the performance of the original model. This is depicted in Figure 4.5.

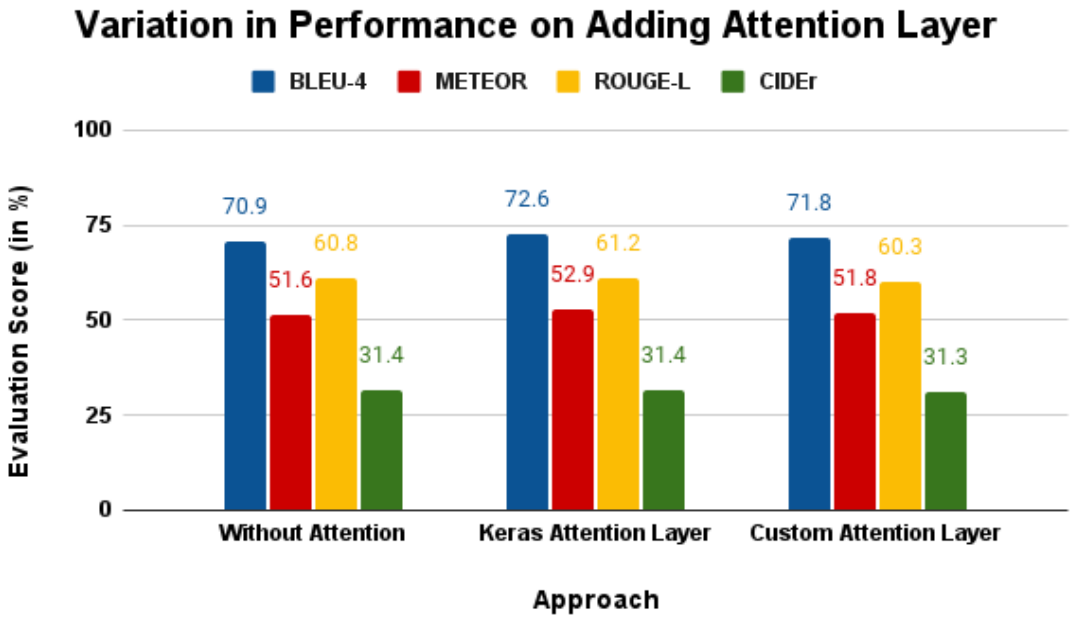

FIGURE 4.5: Variations in Performance on Adding Attention Layer

4. **Latency:** From Table 4.2, we must notice that the average time taken for generation of each caption has halved from the baseline model using only CNN features to the proposed model using both CNN and YOLOv8 features. This is a huge advantage that must be taken into account when attempting video captioning in real-time scenarios. This comparison is illustrated in Figure 4.6. We also notice that while increasing the vocabulary size or introducing the custom vocabulary increased the latency in caption generation, the integration of an attention mechanism significantly decreased the average time taken for caption generation. Thus, we can confidently claim that either of the models comprising the attention layer can be easily used for video captioning in real-time.

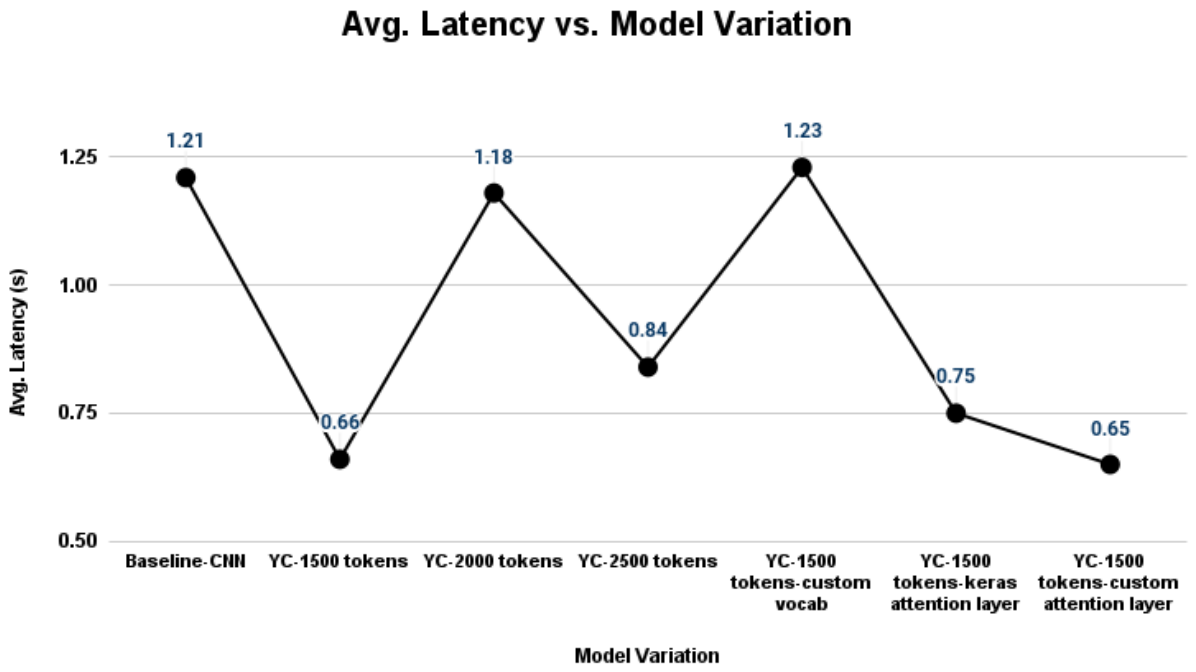

FIGURE 4.6: Variations in Latency

After our evaluation of the model variations, the two variations of the best model are chosen for generating video captions in real-time.

4.3.2 Testing of the Proposed Model

Thus, the proposed model is derived after incorporating feature fusion and an attention mechanism. The entire MSVD dataset has been used to build this model, with training conducted on the 1200 training videos, validation on the 100 validation videos, and testing on the remaining 670 test videos to evaluate its performance. The results achieved are depicted in the table below and Figure 4.7.

TABLE 4.3: Performance of Proposed Model on MSVD dataset. T-A:Training Accuracy(%), V-A:Validation Accuracy(%), B:BLEU-4(%), M-METEOR(%), R:ROUGE-L(%), C:CIDER(%), T:Average time taken for caption generation(in seconds)

MODEL	**EPOCHS = 120**	**T-A**	**V-A**	**B**	**M**	**R**	**C**	**T**
CNN-YOLOv8 Fusion Keras Attention Layer (CYF-KAL)	early stopping at 40	69.25	62.44	74.4	54.3	63.4	33.3	0.73
CNN-YOLOv8 Fusion Custom Attention Layer (CYF-CAL)	early stopping at 43	69.54	62.55	73.5	53.4	62.4	32.7	0.71

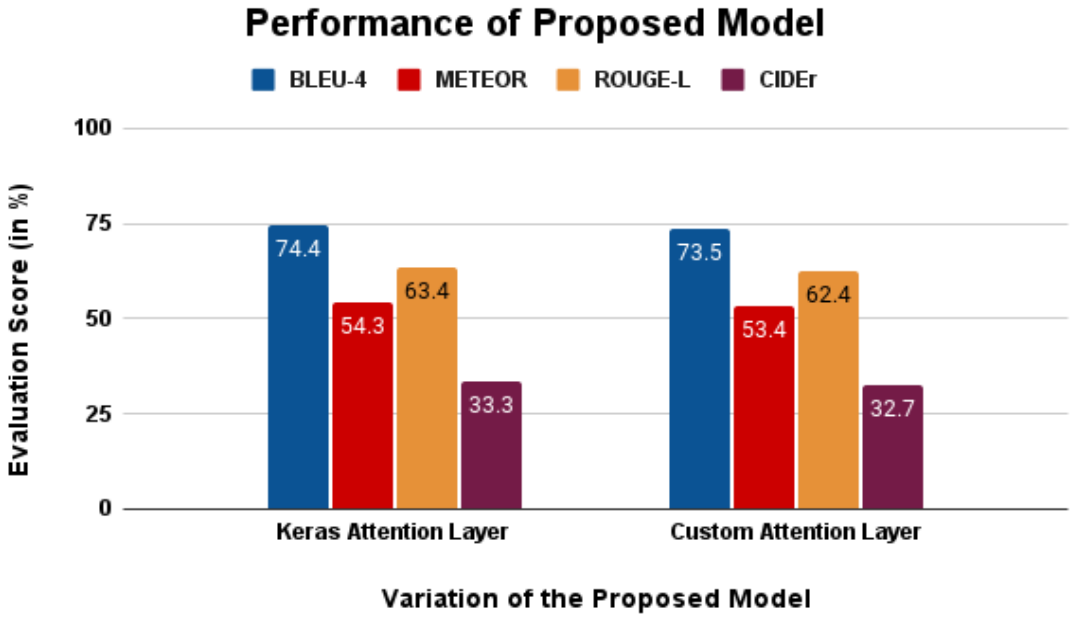

FIGURE 4.7: Performance of the two variations of the proposed model on MSVD dataset

4.3.3 Testing Real-Time Video Captioning

Any new input video undergoes the feature extraction process described in Section 3.4. Features from all 80 frames are extracted, fed to an attention mechanism, and then passed into the trained encoder model as shown in Figure 4.8. The final state of the encoder represents the combined uniform context vector representation that is fed into the decoder as its initial state, along with the ⟨bos⟩token, so that the decoder starts its prediction of the first word.

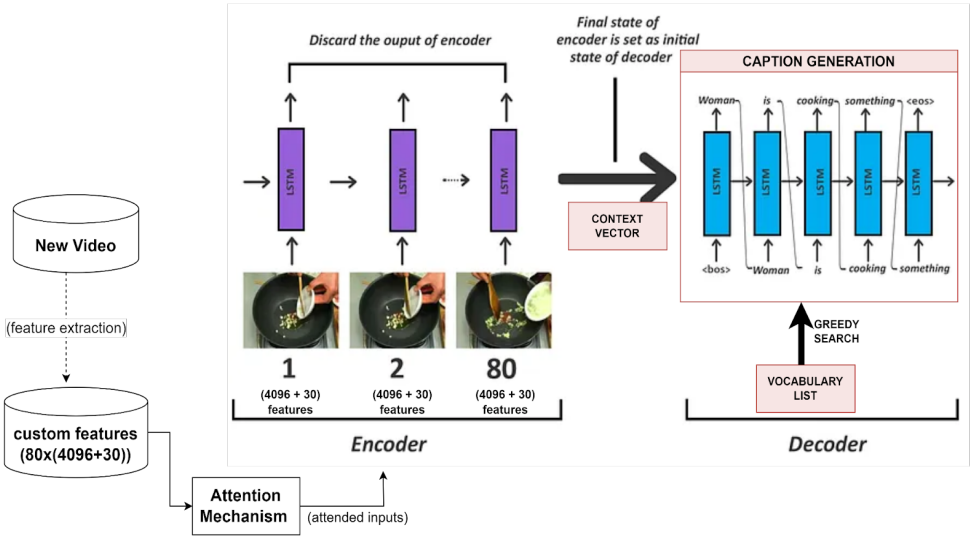

FIGURE 4.8: Encoder-Decoder Architecture - Caption Generation

The decoder also takes in the tokenized custom vocabulary list as part of its input. It then generates a probability distribution across this vocabulary list of a preset number of unique words, and a greedy search algorithm is used for sampling the probability distributions to generate the most likely sequence of words as the caption. Alternatively, a beam search strategy can also be used, but through experimentation, we conclude that the greedy approach works better for a real-time input video stream.

4.4 PERFORMANCE ANALYSIS

4.4.1 Performance Metrics

Various automatic evaluation techniques are available for video captioning and typically, multiple metrics are employed simultaneously as they can complement each other to provide a comprehensive performance summary. For caption generation, the main criteria for evaluation usually include accuracy with respect to the video content, fluency of the sentence and similarity to human-provided ground truth captions. The latter is the most common criterion used by some widely used metrics today. The four major performance metrics used for comparison in this research work are explained below:

4.4.1.1 BLEU-4

The Bilingual Evaluation Understudy Score (BLEU) [21] is a metric that compares a candidate sentence with a reference sentence. It has been widely adopted since this score is easy to compute and is highly consistent with human evaluation.

The evaluation works by counting matching n-grams in the candidate sentence and in the reference sentence, without regard to the word order in the sentences. A perfectly matching sentence is given a score of 1.0 while a perfectly mismatched sentence is given a score of 0.0. The BLEU-4 is the cumulative 4-gram BLEU score and is computed with even weights of (0.25,0.25,0.25,0.25).

4.4.1.2 METEOR

Metric for Evaluation of Translation with Explicit Ordering (METEOR) [4] score is a metric that measures the quality of a generated candidate sentence based on the alignment between the candidate text and the reference text. It lies in the range of 0 to 1 with higher values indicating better scores.

4.4.1.3 ROUGE-L

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [8] is an evaluation metric that measures the number of common n-grams in a set of candidate sentences and reference sentences. It uses both precision and recall to compare a set of candidate sentences against a set of reference sentences. However, as precision and recall can be complementary, to achieve a balance, the ROUGE is calculated as the F1-score.

ROUGE-L, the metric used for comparison here, is a type of ROUGE that compares the length of the Longest Common Subsequence (LCS) instead of counting the matching n-grams. The value ranges from 0 to 1, with the score increasing proportionally to the quality of the predicted candidate.

4.4.1.4 CIDEr

Consensus-based Image Description Evaluation (CIDEr) [30] measures how similar a generated candidate sentence is to a reference sentence based on the concept of consensus. It evaluates the similarity of the two not only in terms of vocabulary and fluency, but also semantics. It is computed using the TF-IDF

scores of each n-gram of the candidate and reference sentences. The CIDEr score typically has a range of 0-10 which is scaled to 0-1.

4.4.1.5 Latency

When processing videos in real-time, one cannot afford to have huge latency in caption generation. Therefore, the average time taken to generate each caption is an important measure of how well the model will perform in a real-time scenario.

4.4.2 Comparison with State-of-the-art (SOTA) Models

Table 4.4 presents a performance comparison between our proposed model variations (CYF-KAL and CYF-CAL) and state-of-the-art models on the MSVD dataset. We find that while achieving superior results in BLEU-4 and METEOR, our models exhibit lower performance in ROUGE-L and CIDEr.

TABLE 4.4: Performance comparison with state-of-the-art approaches. BLEU-4, METEOR, ROUG-L and CIDER scores are represented as percentages.

APPROACH	**BLEU-4**	**METEOR**	**ROUGE-L**	**CIDEr**
MA-LSTM [33]	36.5	26.5	41	59.8
PickNet [7]	41.3	27.7	44.1	59.8
DSD-3 DS-SEM [28]	50.1	34.7	**73.1**	76
SAAT [34]	46.5	33.5	69.4	81
EtENet-IRv2step1 [20]	49.1	33.6	69.5	**83.5**
EtENet-IRv2step2 [20]	50	34.3	70.2	60.6
CYF-KAL (proposed approach)	**74.4**	**54.3**	63.4	33.3
CYF-CAL (proposed approach)	73.5	53.4	62.4	32.7

Our analysis of Table 4.4 reveals several key findings -

- The multimodal attention mechanism suggested by MA-LSTM [33] is rewarded with fair results on the MSVD dataset, but the proposed method still achieves better scores with respect to all the evaluation metrics except CIDEr.

- The PickNet [7] approach utilises reinforcement learning to demonstrate that by picking out important frames alone video captioning can provide good results, but again, the proposed model scores better in all evaluation metrics except CIDEr.

- The two-stage approach employed here in PickNet appears superior to end-to-end training (EtENet models [20]) in terms of BLEU-4 and METEOR.

- DSD-3 DS-SEM [28] and SAAT [34] prioritize domain-specific semantics and sentence syntax, respectively, potentially explaining their dominance in ROUGE-L and CIDEr scores.

- Conversely, our lexical approach might be hindered by a limited vocabulary, leading to less diverse and descriptive captions, thereby impacting CIDEr performance. Encouragingly, our proposed model achieves the highest BLEU-4 and METEOR scores, and trails only four models in ROUGE-L, demonstrating its overall effectiveness.

A summary of the comparison is presented in Figure 4.9.

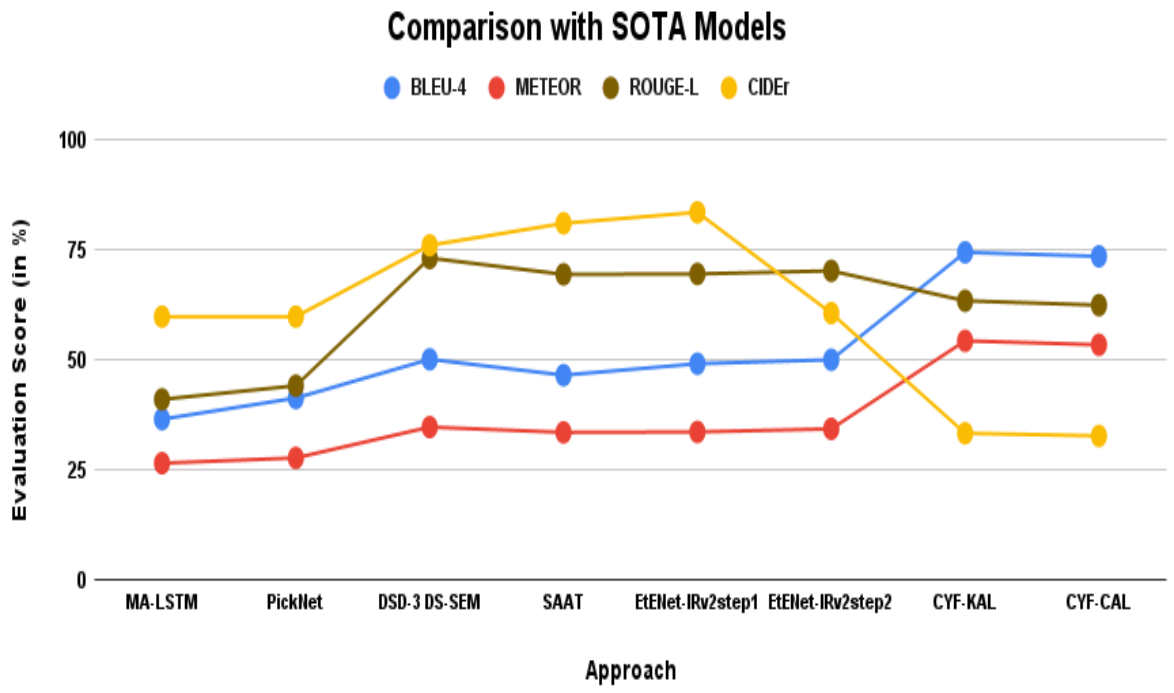

FIGURE 4.9: Comparing CYF-KAL and CYF-CAL with Current State-of-the-art Models

Thus, the proposed model is compared with existing state-of-the-art models.

4.4.3 Sample Output

Figure 4.10 shows a frame from a test video along with the corresponding caption generated by the baseline model and the proposed model. From this, we observe that the captions generated by the proposed model are significantly better than those generated by the baseline model due largely to the refined vocabulary and attention mechanism.

FIGURE 4.10: Baseline model caption: A man is riding a bike
Proposed model caption: a person is riding a motorcycle

Both variations of the proposed model generate good captions with minor differences. Figure 4.11 and Figure 4.12 show the captions generated by the two models for the same video sample. The names of the video files are also provided in the description of the figures for easy comparison with the set of ground-truth captions provided in the MSVD dataset. We notice that in Figure 4.11, both models generate the same caption. However, similar to how we observed in the performance comparison of model variations, we find that CYF-CAL generates the caption slightly faster than CYF-KAL. For Figure 4.12, both variations generate different captions, but both are meaningful and accurate.

Thus, sample output from the proposed model is demonstrated.

FIGURE 4.11: MSVD video file: g36ho6UrBz0_5_20.avi
CYF-KAL: a person is playing a guitar, Time taken: 0.57s
CYF-CAL: a person is playing a guitar, Time taken: 0.55s

FIGURE 4.12: MSVD video file: fnpp8v9NbmY_181_188.avi
CYF-KAL: a person is putting ingredients in a bowl, Time taken: 0.71s
CYF-CAL: a person is mixing ingredients, Time taken: 0.50s

4.4.4 Qualitative Analysis

The proposed model demonstrates qualitative improvements in caption generation over the baseline, mainly due to incorporation of the discussed feature fusion procedure, the attention mechanism and the refined vocabulary. However, we must acknowledge that while the refined captions provide the advantages of being less complicated and gender-neutral, the technique is not completely effective in improving scene understanding as the model learns from a limited vocabulary. Moreover, when the model is trained with a finite number of top words in the vocabulary chosen as tokens, there is a chance that important parts of a sentence generated as the caption could be missed. For example, after training the model on the entire MSVD training dataset, we found that some captions were generated without the object part of the sentence as shown in Figure 4.13. This is also why some activities in the videos are misidentified.

FIGURE 4.13: MSVD video file: j2Dhf-xFUxU_13_20.avi
CYF-KAL: a person is slicing a carrot, Time taken: 0.60s
CYF-CAL: a person is slicing an, Time taken: 0.55s

Further, some objects are misidentified due to large similarities in their features. There are many instances in which the model described a baby as a person and vice-versa, although, after observing the original annotations in the dataset, we must allow that these errors could arise from the description of the videos in the ground-truth references itself.

With respect to real-time video captioning, the model performs admirably due to segmentation of the input video stream and reduced latency in caption generation. Still, we realize that a good amount of computational resources and time are wasted if the caption is regenerated in the next video segment when there is no significant change in the scene. We believe that this proposed method for video captioning would become even more effective if integrated with frame comparison and boundary detection techniques. This would mean that a suitable caption would be generated for the changing scenes in the video only as and when required.

Thus, the performance of the proposed model is analysed in all aspects to understand its overall benefits and limitations.

CHAPTER 5

SOCIAL IMPACT AND SUSTAINABILITY

As always, the benefits and issues pertaining to this project must be considered. The main points are highlighted below:

5.1 SOCIETAL IMPACT

- The widespread adoption of automatic video captioning can enhance accessibility for individuals with hearing impairments, making multimedia content more inclusive.
- However, there may be concerns regarding the potential impact on traditional manual captioning jobs, potentially leading to job displacement or changes in employment opportunities.

5.2 HEALTH AND SAFETY

- From a health perspective, prolonged exposure to multimedia content, especially through video streaming platforms, can contribute to issues such as eye strain and digital fatigue.
- Safety concerns may arise if automatic video captioning is used in critical applications, such as medical or emergency response scenarios, where inaccurate captions could lead to misunderstandings or errors.

5.3 LEGAL CONSIDERATIONS

- Legal frameworks surrounding privacy and data protection must be considered, particularly when processing video content that may include identifiable individuals or sensitive information.
- Compliance with accessibility regulations is essential to ensure equal access to video content for individuals with disabilities.

5.4 ENVIRONMENTAL IMPACT

- The computational resources required for training and deploying deep learning models for automatic video captioning can have environmental implications, contributing to increased energy consumption and carbon emissions.
- Strategies for energy-efficient algorithms and infrastructure optimization are crucial.

5.5 CULTURAL SENSITIVITY

- Automatic video captioning systems must be culturally sensitive and capable of accurately transcribing and interpreting diverse languages, accents, and dialects.
- Careful consideration should be given to cultural nuances and context when generating captions to avoid misinterpretations or offensive content.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSION

This research work proposes a novel real-time video captioning approach using feature fusion and an attention mechanism. The method involves extracting features from video frames using a VGG16 CNN and detecting objects using YOLOv8. These features are combined and fed into an LSTM with attention for caption generation. Trained on a refined MSVD dataset, the model achieves promising performance.

6.2 FUTURE SCOPE

It can be said that the proposed method offers advantages over existing models due to its incorporation of additional object detection mechanism, and subsequent attention mechanisms for distinguishing the features. Future work includes exploring 3D CNNs for better identification of temporal features, background understanding, and selective frame processing for improved captioning accuracy and reduced latency.

Appendix A

GLOSSARY

The glossary provides a comprehensive list of all the abbreviations used in this report along with their full forms.

A.1 DATASETS

- **MSVD**: Microsoft Research Video Description Corpus

A.2 PRE-TRAINED MODELS

- **CNN**: Convolutional Neural Network
- **LSTM**: Long short-term memory
- **VGGNet**: Visual Geometry Group Network
- **YOLO**: You Only Look Once

A.3 STATE-OF-THE-ART (SOTA) Models

- **MA-LSTM**: Multi-Attention Based LSTM
- **DSD-3 DS-SEM**: Domain Specific Decoder - TOP 3 -Domain-Specific Semantics
- **SAAT**: Syntax-Aware Action Targeting
- **EtENet-IRv2** End-To-End Inception Residual Network v2

REFERENCES

1. Abdar, M., Kollati, M., Kuraparthi, S., Pourpanah, F., McDuff, D., Ghavamzadeh, M., Yan, S., Mohamed, A., Khosravi, A., Cambria, E., and Porikli, F. (2023, April 22). 'A Review of Deep Learning for Video Captioning'. arXiv preprint arXiv:2204.08691, pp. 1-42.

2. Al-Malla, M.A., Jafar, A., and Ghneim, N. (2022). 'Image captioning model using attention and object features to mimic human image understanding', J Big Data 9, Article 20, pp. 1-16.

3. Amirian, S., Farahani, A., Arabnia, H., Rasheed, K., and Taha, T. (2020, December). 'The Use of Video Captioning for Fostering Physical Activity'. In 2020 International Conference on Computational Science and Computational Intelligence (CSCI) IEEE, pp. 611-614.

4. Banerjee, S., and Lavie, A. (2005, June). 'METEOR: An automatic metric for MT evaluation with improved correlation with human judgments'. In Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp. 65-72.

5. Bisong, E., and Bisong, E. (2019) 'Google colaboratory', Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners, pp. 59-64.

6. Chen, D., and Dolan, W. B. (2011). 'Collecting highly parallel data for paraphrase evaluation.' In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies Vol. 1, pp. 190-200.

7. Chen, Y., Wang, S., Zhang, W., and Huang, Q. (2018). 'Less is more: Picking informative frames for video captioning'. In Proceedings of the European conference on computer vision (ECCV), pp. 358-373.

8. Chin-Yew Lin, 2004, 'ROUGE: A Package for Automatic Evaluation of Summaries', In Text Summarization Branches Out, Barcelona, Spain. Association for Computational Linguistics, pp. 74-81.

9. Deng, J., Dong, W., Socher, R., Li, L. J., Li, Kai, and Li, Fei-Fei. (2009). 'ImageNet: A large-scale hierarchical image database'. 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248-255.

10. Dey, A. U., Ghosh, S. K., Valveny, E., and Harit, G. (2021). 'Beyond visual semantics: Exploring the role of scene text in image understanding.' Pattern Recognition Letters, 149(1), pp. 164-171.

11. Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. (2022, October). 'Make-a-scene: Scene-based text-to-image generation with human priors'. In European Conference on Computer Vision, Springer Nature Switzerland, pp. 89-106.

12. Granitz, N., Kohli, C., and Lancellotti, M. P. (2021). 'Textbooks for the YouTube generation? A case study on the shift from text to video'. Journal of Education for Business, 96(5), pp. 299–307.

13. Hochreiter, S., and Schmidhuber, J. (1997). 'Long short-term memory'. Neural computation, 9(8), pp. 1735-1780.

14. Iashin, V., and Rahtu, E. (2020, June). 'Multi-modal Dense Video Captioning'. In the proceedings of CVPR Workshops, pp. 213-220.

15. Jocher, G., Chaurasia, A., and Qiu, J. (2023). 'Ultralytics YOLO (Version 8.0.0)' [Computer software]. Available: https://github.com/ultralytics/ultralytics

16. Karve, P., Thorat, S., Mistary, P., and Belote, O. (2022). 'Conversational Image Captioning Using LSTM and YOLO for Visually Impaired'. In Proceedings of Third International Conference on Communication, Computing and Electronics Systems, Springer, Singapore, pp. 237-244.

17. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, C. L. (2014). 'Microsoft coco: Common objects in context'. In Computer Vision–ECCV 2014, Springer International Publishing, pp. 740-755.

18. Luong, M. T., Pham, H., and Manning, C. D. (2015). 'Effective approaches to attention-based neural machine translation'. arXiv preprint arXiv:1508.04025, pp. 1-11.

19. Nadeem, U., Shah, S. A., Sohel, F., Togneri, R., and Bennamoun, M. (2019). 'Deep learning for scene understanding'. In Handbook of deep learning applications, pp. 21-51.

20. Olivastri, S., Singh, G., and Cuzzolin, F. (2019, June). 'End-to-end video captioning'. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 1-9.

21. Papieni, K., Roukos, S., Ward, T., and Zhu, W.J. (2002). 'Bleu: a method for automatic evaluation of machine translation', In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311-318.

22. Preethi, A., and Dhanalakshmi, P. (2023). 'Video Captioning using Pre-Trained CNN and LSTM.' In 2023 International Conference on Signal

Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, pp. 1-7.

23. Rafiq, G., Rafiq, M., and Choi, G. S. (2023). 'Video description: A comprehensive survey of deep learning approaches'. Artif Intell Rev, 56, pp. 13293–13372.

24. Saldanha, D.L., Kesari, R.T., Srinivas, K.R. and Natarajan, S. 'Scene Description Using Keyframe Extraction and Image Captioning', 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2023, pp. 0662-0668.

25. Salehi, A., and Balasubramaniam, M. (2023, February 28). 'DDCNet: Deep dilated convolutional neural network for dense prediction' [Preprint]. arXiv:523:116-29.

26. Saroja, M., and Mary, A. B. (2023). 'Image Captioning Using Improved YOLO V5 Model and Xception V3 Model', pp. 1-19.

27. Shahir Zaoad, Md., Mannan, M. M. Rushadul, Mandol, Angshu Bikash, Rahman, Mostafizur, Islam, Md. Adnanul, and Rahman, Md. Mahbubur. (2023). 'An attention-based hybrid deep learning approach for Bengali video captioning'. Journal of King Saud University - Computer and Information Sciences, 35(1), pp.257-269.

28. Shekhar, C. C. (2020). 'Domain-specific semantics guided approach to video captioning.' In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1587-1596.

29. Simonyan, K., and Zisserman, A. (2014). 'Very deep convolutional networks for large-scale image recognition', arXiv preprint arXiv:1409.1556, pp. 1-14.

30. Vedantam, R., Zitnick, C. L., and Parikh, D. (2015). 'Cider: Consensus-based image description evaluation'. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4566-4575.

31. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). 'Sequence to sequence-video to text'. In Proceedings of the IEEE international conference on computer vision, pp. 4534-4542.

32. Vo-Ho, V. K., Luong, Q. A., Nguyen, D. T., Tran, M. K., and Tran, M. T. (2019). 'A Smart System for Text-Lifelog Generation from Wearable Cameras in Smart Environment Using Concept-Augmented Image Captioning with Modified Beam Search Strategy'. Applied Sciences, 9(9), pp. 1886.

33. Xu, J., Yao, T., Zhang, Y., and Mei, T. (2017). 'Learning multimodal attention LSTM networks for video captioning'. In Proceedings of the 25th ACM international conference on Multimedia, pp. 537-545.

34. Zheng, Q., Wang, C., and Tao, D. (2020). 'Syntax-aware action targeting for video captioning'. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13096-13105.

35. 'Cisco Report'. Accessed on: April 2024. Available: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html.

36. 'GitHub repository: Video-Captioning'. Accessed on: April 2024. Available: https://github.com/Shreyz-max/Video-Captioning

37. 'Keras Attention Layer'. Accessed on: April 2024. Available: https://keras.io/api/layers/attention_layers/attention/

38. 'Keras Tokenizer'. Accessed on: April 2024. Available: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

39. 'LSTM Cell'. Accessed on: April 2024. Available: https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2

40. 'The Power of Visual Communication'. Accessed on: April 2024. Available: https://www.publitas.com/blog/the-power-of-visual-communication/

41. 'The Power of Visual Communication: Why Video Brings Text to Life'. Accessed on: April 2024. Available: https://pictory.ai/blog/the-power-of-visual-communication-why-video-brings-text-to-life

42. 'VGG16 CNN Architecture'. Accessed on: April 2024. Available: https://datagen.tech/guides/computer-vision/vgg16/