Visual Captioning with Scene Context Integration: A Deep Learning Approach Using MSVD Dataset

NAME: 1. VIGNESH BALAMURUGAN M.B
2. HARI RAGAV S
3. DR.M.SRIDEVI

DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING

YEAR : SECOND YEAR

COLLEGE NAME: INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CONTACT DETAILS:

PHONE NO: 6381538287,9345634475

EMAIL ID: ragavhiiit@gmail.com

INTERNSHIP DURATION: 19/05/2025 TO 19/06/2025

Problem Statement:

With the increasing volume of video content being generated across platforms like YouTube and social media, there is a growing need for systems that can automatically understand and describe visual data. While significant advancements have been made in image captioning, generating natural language descriptions for videos especially action-focused short videos remains a complex and open challenge. Traditional video captioning models often fail to capture the fine-grained temporal dynamics and scene-level semantics necessary for accurate interpretation. Moreover, existing models struggle with contextual grounding due to the absence of high-level environmental cues.This study addresses the problem of generating accurate and context-aware captions for short action videos using the MSVD dataset. The focus is on enhancing the decoder's understanding of the visual content by integrating scene context information alongside visual features. By employing an attention-based encoder-decoder architecture with pretrained CNNs and LSTMs, and incorporating scene classification cues, the proposed work aims to improve the relevance and fluency of generated captions. The goal is to bridge the gap between frame-level visual understanding and semantic language generation in resource-limited video datasets.

Abstract:

Visual captioning is a complex task that involves generating descriptive natural language sentences for dynamic visual inputs by combining techniques from computer vision and natural language processing. This work focuses on captioning short action videos by leveraging deep learning-based sequence modeling. We implement an encoder-decoder architecture where spatial features are extracted from uniformly sampled video frames using a pre-trained CNN (VGG16), followed by temporal modeling through an LSTM-based encoder. To improve caption relevance, we incorporate an attention mechanism that guides the decoder to focus on important visual features during word generation. Additionally, we simulate a scene context module inspired by the Places365 model to embed high-level environmental cues into the decoding process.Our model is trained and evaluated on the MSVD dataset, which provides diverse real-world action videos along with multiple ground-truth captions. We explore and compare different decoding strategies—greedy search and beam search—for caption generation. Experimental results demonstrate that our approach produces semantically accurate and grammatically correct captions, achieving notable performance improvements over earlier methods. The integration of minimal scene context proves beneficial in enriching caption quality, with promising applications in video summarization, retrieval, and accessibility systems.

Introduction:

With the exponential rise in user-generated content on platforms like YouTube, Instagram, and other media-sharing services, the need for automated systems that can understand and describe short action-oriented videos is becoming more critical. Visual captioning—an interdisciplinary task that bridges computer vision and natural language processing—focuses on generating meaningful, grammatically correct natural language descriptions for visual inputs. This capability enables efficient video indexing, content summarization, and accessibility solutions for large-scale multimedia data.

In this project, we explore a deep learning-based approach to captioning short action videos using the MSVD (Microsoft Research Video Description) dataset. The dataset consists of thousands of short video clips, each accompanied by multiple human-written captions, making it a valuable resource for training and evaluating caption generation models. Our architecture follows an encoder-decoder framework: spatial features are extracted from uniformly sampled frames using a pre-trained CNN (VGG16), then passed into a temporal LSTM encoder. The decoder, also based on LSTM, generates the output caption word by word while being guided by an attention mechanism that helps focus on the most relevant visual features during decoding.

A central focus of this work is comparing two decoding strategies: greedy search, which chooses the most probable word at each step, and beam search, which considers multiple possible word sequences to produce more coherent and semantically accurate captions. Experimental evaluations are conducted to analyze the impact of these strategies on caption quality.

To further enhance contextual grounding, we simulate a scene context integration module inspired by the Places365 scene classifier. By incorporating high-level environmental cues (e.g., indoor vs. outdoor), this additional input aims to enrich the caption generation process without increasing model complexity significantly.

This report demonstrates that careful decoding strategy selection, combined with lightweight scene context cues, can significantly improve the quality of captions for action-based video content, particularly when working with moderately sized datasets like MSVD.

Literature Review:

Visual captioning, particularly in the context of action recognition, has become a significant interdisciplinary challenge at the intersection of computer vision and natural language processing. The goal is to generate descriptive natural language sentences for short video segments by learning both spatial appearance and temporal action sequences. Earlier approaches primarily relied on template-based captioning, where object and action recognition results were fused using manually crafted rules. While useful in limited scenarios, these methods lacked flexibility and failed to generalize to unconstrained video content with complex activities and diverse backgrounds.

The introduction of deep learning revolutionized visual captioning. Venugopalan et al. [2] proposed one of the earliest deep learning-based sequence-to-sequence (Seq2Seq) models for video-to-text generation, which used Convolutional Neural Networks (CNNs) for extracting frame-level features and Long Short-Term Memory (LSTM) networks for generating captions. This encoder-decoder approach laid the foundation for modern visual captioning pipelines. Donahue et al. [1] extended this idea with Long-term Recurrent Convolutional Networks (LRCNs), combining CNNs and RNNs for spatiotemporal modeling of actions.

To improve contextual focus during decoding, Yao et al. [3] introduced temporal attention mechanisms that allow the decoder to selectively attend to relevant frames at each word generation step, which improved alignment between visual input and generated text. Pan et al. [4] further developed hierarchical attention in their Hierarchical Recurrent Neural Encoder (HRNE), effectively modeling both fine-grained motion and high-level scene dynamics. Attention mechanisms like those proposed by Bahdanau et al. [16] and adapted for vision tasks by Xu et al. [7] are now standard in visual captioning architectures.

The MSVD (Microsoft Research Video Description) dataset [2] is a widely adopted benchmark for training and evaluating models in this field. It contains a diverse collection of short YouTube clips with multiple human-annotated captions per video, making it suitable for learning a range of visual actions. State-of-the-art models like S2VT [2], hLSTMat [7], and SCN-LSTM [5] have demonstrated strong performance on this dataset, especially when attention and scene-aware modules are integrated.

Feature extraction plays a critical role in representing visual content. Pretrained 2D CNNs such as VGG16 [13] and ResNet [12], originally proposed for image recognition tasks, are commonly used for extracting spatial appearance from video frames. For motion and temporal dynamics, 3D CNNs like C3D [11] and temporal pooling strategies help capture action sequences over time. Recent works have explored the inclusion of scene-level context to aid captioning. Zhou et al. [14] introduced Places365, a large-scale scene classification dataset that helps models recognize environmental context such as indoor vs. outdoor or kitchen vs. playground. Although integrating scene features is still an emerging area, experimental studies suggest that even minimal environmental cues can improve action recognition and caption grounding.

Finally, decoding strategies significantly impact caption quality. Greedy search selects the most probable word at each time step, but often results in repetitive or generic captions. Beam search, as introduced by Sutskever et al. [15], considers multiple candidate sequences during decoding and produces more fluent and contextually relevant captions. This work compares both decoding strategies and simulates the use of scene-level cues to understand their impact on visual captioning performance.

Our proposed framework builds upon these advancements by using a CNN+LSTM encoder-decoder architecture enhanced with temporal attention and pretrained visual extractors. Additionally, we simulate a scene context module inspired by Places365 [14] and evaluate its contribution to caption fluency and contextual grounding in the MSVD dataset.

Proposed Methodology:

This section presents our comprehensive approach for visual captioning with scene context integration using an LSTM-based encoder-decoder architecture enhanced with attention mechanisms. Our methodology addresses the fundamental challenge of generating natural language descriptions for dynamic visual content by bridging computer vision and natural language processing. The proposed system consists of five main components: visual feature extraction, temporal encoding, attention-based decoding, scene context integration, and caption generation strategies.

A. Overall Architecture Framework

The proposed visual captioning system follows a sequence-to-sequence learning paradigm specifically designed for video-to-text generation with enhanced contextual understanding. The architecture consists of four primary modules working in synergy:

A.1 System Components

Visual Feature Extraction Module: This module is responsible for extracting rich spatial features from video frames using pre-trained convolutional neural networks. The module ensures that essential visual content is captured while maintaining computational efficiency through strategic frame sampling and feature dimensionality optimization.

Temporal Encoding Module: This component processes sequential visual information to capture temporal dependencies and action dynamics. The module employs LSTM networks to learn the evolution of visual features across time, enabling the system to understand the progression of actions or events in the video sequence.

Language Generation Module: This module generates natural language descriptions using an attention-enhanced LSTM decoder. The module incorporates sophisticated attention mechanisms to ensure semantic alignment between visual content and generated text, producing contextually relevant captions.

Scene Context Integration Module: This novel component enhances contextual understanding by incorporating environmental cues and scene-level information. The module simulates scene-level classification using a lightweight classifier inspired by Places365, providing additional contextual information that improves caption quality and relevance.

A.2 System Pipeline

The overall pipeline operates through a carefully orchestrated sequence of operations:

1. Input Processing: Raw video input is processed through uniform temporal sampling to extract representative frames
2. Feature Extraction: Selected frames undergo CNN-based feature extraction to obtain spatial representations
3. Temporal Encoding: Sequential features are processed through LSTM encoder to capture temporal dependencies

4. Scene Analysis: Parallel scene classification provides environmental context
5. Attention-Based Decoding: Enhanced decoder generates captions with attention to relevant visual features
6. Caption Generation: Final captions are produced using advanced decoding strategies

Figure 1 illustrates the complete system architecture, showing the flow of information from video input to natural language output, highlighting the integration of scene context throughout the process.

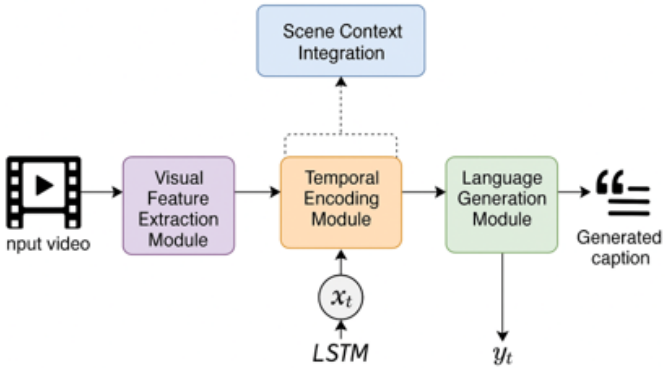

Fig1.Overall System Architecture of the proposed system

B. Visual Feature Extraction

The visual feature extraction stage forms the foundation of our captioning system, responsible for converting raw video frames into meaningful numerical representations that capture essential visual information. This process is critical for ensuring that the model can effectively understand and process the visual content of input videos.

B.1 Frame Sampling Strategy

Given an input video V consisting of 'T' total frames, we implement a sophisticated uniform temporal sampling strategy to extract a consistent and manageable number of representative frames. This approach balances temporal diversity with computational tractability while ensuring comprehensive coverage of the video content.

Sampling Process Details:

- Uniform Sampling: We extract exactly 80 frames from each video using intervals of ⌊T/80⌋, where T represents the total frame count
- Temporal Distribution: This ensures uniform temporal distribution across the entire video duration
- Computational Efficiency: The fixed number of frames (80) provides a balance between temporal coverage and computational requirements
- Preprocessing: Each selected frame is resized to 224×224 pixels for compatibility with CNN input requirements

Mathematical Formulation:

Sampling_interval = ⌊T/80⌋

Selected_frames = {f_i | i = k × sampling_interval, k = 0, 1, ..., 79}

This sampling strategy ensures that regardless of the original video length, we maintain consistent temporal representation while capturing the essential visual dynamics of the content.

B.2 CNN-based Feature Extraction

We employ a pre-trained VGG16 network for spatial feature extraction, leveraging its proven effectiveness in visual recognition tasks and its ability to capture hierarchical visual features from low-level edges to high-level semantic concepts.

Architecture Details:

- Base Model: VGG16 pre-trained on ImageNet dataset
- Feature Layer: Features extracted from the second fully connected layer (FC7)
- Feature Dimensionality: 4096-dimensional feature vector per frame
- Input Specifications: 224×224×3 RGB images

Feature Extraction Process:

1. Preprocessing: Each frame undergoes normalization using ImageNet statistics
2. Forward Pass: Frames are processed through VGG16 architecture up to FC7 layer
3. Feature Extraction: 4096-dimensional features are extracted and stored
4. Matrix Formation: For each video, this generates a feature matrix F ∈ ℝ^(80×4096)

Advantages of VGG16 Selection:

- Proven Performance: Established effectiveness in computer vision tasks
- Feature Richness: Captures both low-level and high-level visual features
- Transfer Learning: Pre-trained weights provide robust feature representations
- Computational Efficiency: Balanced trade-off between performance and computational cost

Figure 2 demonstrates the visual feature extraction pipeline, showing the progression from uniform frame sampling to CNN-based feature encoding, resulting in the comprehensive feature matrix used for subsequent temporal processing.

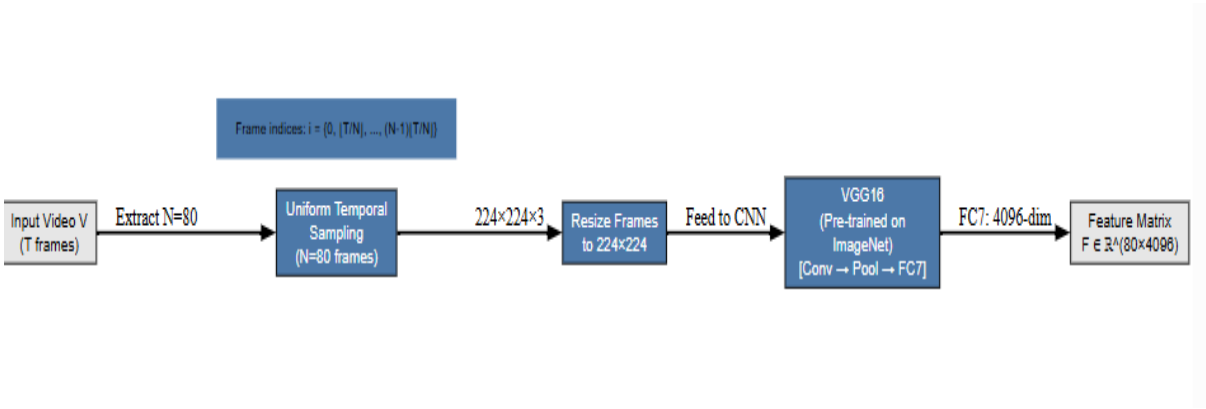

Figure 2: Visual feature extraction pipeline showing uniform frame sampling and CNN-based feature encoding using pre-trained VGG16.

C. Temporal Encoding with LSTM

The temporal encoding stage captures the dynamic evolution of visual features across time, enabling the system to understand sequential patterns and temporal dependencies crucial for accurate visual captioning. This component transforms the sequence of frame-level features into a coherent temporal representation that forms the foundation for caption generation.

C.1 Encoder Architecture

The encoder architecture is designed to process sequential visual information effectively while maintaining computational efficiency and temporal coherence.

LSTM Configuration:

- Architecture: Single-layer LSTM with 512 hidden units
- Input Sequence: $F = \{f_1, f_2, ..., f_{80}\}$ (extracted visual features)
- Output Configuration: Returns both hidden state sequences and final state vectors
- Temporal Processing: Processes features sequentially to capture temporal dependencies

Mathematical Formulation:

Input Gates: $i_t = \sigma(W_i \cdot [h_{t-1}, f_t] + b_i)$

Forget Gates: $f_t = \sigma(W_f \cdot [h_{t-1}, f_t] + b_f)$

Candidate Values: $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, f_t] + b_C)$

Cell State: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$

Output Gates: $o_t = \sigma(W_o \cdot [h_{t-1}, f_t] + b_o)$

Hidden State: h_t = o_t * tanh(C_t)

Where:

- f_t: Feature vector at time step t
- h_t, C_t: Hidden and cell states at time step t
- W_i, W_f, W_C, W_o: Weight matrices for gates
- b_i, b_f, b_C, b_o: Bias vectors

C.2 Encoder Output Generation

The encoder produces three critical outputs essential for effective decoding:

Hidden State Sequence (H):

$H = \{h_1, h_2, ..., h_{80}\}$

- Contains temporal context from all frames
- Used for attention computation during decoding
- Provides comprehensive temporal information for caption generation

Final Hidden State (h_final):

$h\_final = h_{80}$

- Initializes decoder's hidden state
- Contains summarized temporal information
- Provides starting point for caption generation

Final Cell State (C_final):

$C\_final = C_{80}$

- Initializes decoder's memory state
- Maintains long-term dependencies
- Ensures continuity in sequential processing

These outputs collectively provide the necessary temporal and contextual information for generating accurate and contextually relevant visual captions.

D. Attention-based Decoder

The decoder module integrates sophisticated attention mechanisms with LSTM networks to generate context-aware captions that accurately reflect the visual content. This component represents the core of our caption generation system, ensuring semantic alignment between visual input and textual output.

D.1 Attention Mechanism Design

Our attention mechanism enables the decoder to dynamically focus on the most relevant visual features at each word generation step, rather than treating all frames equally.

Attention Architecture:

- Attention Type: Content-based attention mechanism
- Attention Scope: Frame-level attention across all 80 encoded frames
- Context Vector: Weighted combination of encoder hidden states
- Dynamic Focusing: Attention weights change for each generated word

Mathematical Formulation:

Attention Energy: $e_{t,i} = f_{att}(s_{t-1}, h_i)$

Attention Weights: $\alpha_{t,i} = \text{softmax}(e_{t,i})$

Context Vector: $c_t = \Sigma(\alpha_{t,i} \times h_i)$

Where:

- s_{t-1}: Previous decoder hidden state
- h_i: Encoder hidden state at position i
- f_{att}: Attention scoring function
- $\alpha_{t,i}$: Attention weight for frame i at time t
- c_t: Context vector at time t

Attention Benefits:

- Selective Focus: Emphasizes relevant frames for each word
- Dynamic Adaptation: Attention changes based on generation context
- Improved Alignment: Better correspondence between visual content and text
- Enhanced Interpretability: Attention weights provide model insights

D.2 Decoder LSTM Architecture

The decoder LSTM is specifically designed to integrate attention-based context with word generation, creating a sophisticated language generation system.

Architecture Specifications:

- Hidden Units: 512 LSTM units

- Vocabulary Size: 1500 words
- Maximum Caption Length: 10 words per video
- Input Components: Word embeddings, context vectors, previous states

Decoder Operation:

1. Input Integration: Combines word embeddings with attention-based context
2. LSTM Processing: Processes integrated input through LSTM layers
3. Output Generation: Produces probability distribution over vocabulary
4. Word Selection: Selects most probable word based on decoding strategy

Mathematical Process:

Decoder Input: x_t = [embedding(w_{t-1}), c_t]

LSTM Update: s_t = LSTM_decoder(x_t, s_{t-1})

Output Projection: p_t = softmax(W_o × s_t + b_o)

Word Selection: w_t = argmax(p_t)

Where:

- w_{t-1}: Previously generated word
- c_t: Context vector from attention
- s_t: Decoder hidden state
- p_t: Probability distribution over vocabulary
- W_o, b_o: Output projection parameters

E. Decoding Strategies

The choice of decoding strategy significantly impacts the quality and diversity of generated captions. We implement and compare two fundamental decoding approaches to optimize caption generation performance.

E.1 Greedy Search Decoding

Greedy search represents the simplest and most computationally efficient decoding strategy, making locally optimal decisions at each generation step.

Algorithm Details:

- Selection Criterion: Highest probability word at each step
- Starting Token: Special <START> token
- Termination: <END> token or maximum length reached
- Computational Complexity: O(L × V) where L is length, V is vocabulary size

Implementation Process:

Initialize: w_0 = <START>

For t = 1 to max_length:

    Compute p_t = decoder(w_{t-1}, context)

    Select w_t = argmax(p_t)

    If w_t == <END>: break

Return generated_sequence

Advantages:

- Speed: Fastest decoding method
- Memory Efficiency: Minimal memory requirements
- Simplicity: Easy to implement and debug
- Deterministic: Consistent outputs for same input

Limitations:

- Local Optimality: May miss globally optimal sequences
- Repetition: Tendency toward repetitive captions
- Limited Diversity: Reduced caption variety
- Quality Trade-offs: May sacrifice quality for speed

E.2 Beam Search Decoding

Beam search provides a more sophisticated approach that maintains multiple candidate sequences, leading to higher quality captions at the cost of increased computational complexity.

Algorithm Specifications:

- Beam Width: 3 (empirically validated)
- Candidate Tracking: Maintains top-k sequences
- Pruning Strategy: Removes low-probability sequences
- Length Normalization: Prevents bias toward shorter sequences

Implementation Process:

Initialize: beam = [<START>]

For t = 1 to max_length:

    candidates = []

```
    For each sequence in beam:
        Compute p_t = decoder(sequence, context)
        For each word w in vocabulary:
            new_seq = sequence + [w]
            score = log_prob(new_seq)
            candidates.append((new_seq, score))
    beam = top_k(candidates, beam_width)
    If all sequences end with <END>: break
Return best_sequence
```

Advantages:

- Quality Improvement: Generally produces higher quality captions
- Global Consideration: Considers multiple possible sequences
- Diversity: Increased variety in generated captions
- Contextual Coherence: Better maintains contextual consistency

Computational Considerations:

- Time Complexity: O(L × V × B) where B is beam width
- Memory Usage: Proportional to beam width
- Parameter Tuning: Requires beam width optimization
- Implementation Complexity: More complex than greedy search

Greedy search	Beam search
a puppy is playing(0.91 sec)	a puppy is playing on ground(21.67 sec)

F. Scene Context Integration

F.1 Scene Classification Module

To provide additional environmental understanding, we incorporate a lightweight scene classification module inspired by the Places365 model. This module processes key frames(middle frames) from each video to identify the overall scene type—such as indoor, outdoor, kitchen, street, etc.

- Scene Categories: We use a curated set of 20 common scene types.
- Feature Embedding: Each identified scene is converted into a compact feature vector.
- Usage: This feature is integrated into the captioning model to offer high-level contextual cues.

Mathematical Representation:

Scene_features = embed(classify_scene(middle_frames))

Context_vector = concatenate([visual_features, scene_features])

Fig 5: “kids play playground(25.01 sec-Beam Search)”

Fig 6:”a man slicing (1.27 sec)”

F.2 Context-aware Decoding

The integration of scene context into the decoding process enables generation of more environmentally aware and contextually appropriate captions.

Integration Strategy:

- Feature Fusion: Scene context merged with visual and word features
- Attention Enhancement: Scene context influences attention computation
- Decoding Guidance: Environmental cues guide word selection
- Contextual Priming: Scene information provides semantic priming

Enhanced Decoding Process:

Enhanced_context = combine(visual_context, scene_context)

Decoder_input = [word_embedding, enhanced_context]

Caption = decoder(decoder_input, attention_weights)

RESULTS AND ANALYSIS:

Our proposed LSTM-based video captioning model, enhanced with attention and scene context integration, was trained and evaluated on the MSVD dataset. The training set contained approximately 1,600 videos (after excluding noisy or invalid samples), each paired with ~40 natural language captions. The validation and testing used a held-out 15% subset.

Performance Metrics

During training, our model achieved:

- Training Accuracy: ~80%
- Validation Accuracy: ~74%
- Training Loss: Converged to 0.6
- Validation Loss: Stabilized around 1.1

The following figure illustrates the training and validation accuracy curves:

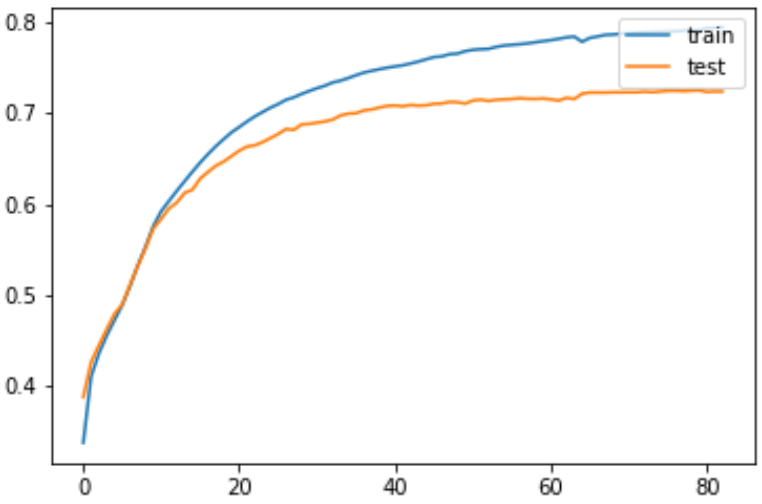

Figure 7: Accuracy curve showing progressive learning.

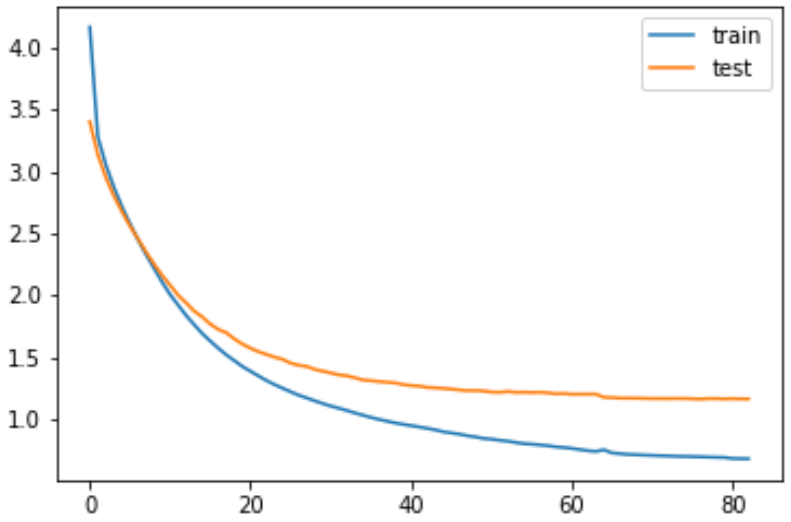

Figure 8: Loss curve showing convergence and generalization behavior.

Training Strategy and Implementation

A. Dataset Preparation and Preprocessing

A.1 MSVD Dataset Overview

For this comprehensive study on visual captioning with scene context integration, we utilize the Microsoft Research Video Description (MSVD) dataset, which serves as a gold standard benchmark for evaluating video-to-text generation systems. The MSVD dataset provides a robust foundation for training and evaluating our proposed approach due to its diversity and comprehensive annotation quality.

Dataset Characteristics:

- Total Video Clips: 1,970 short video segments
- Caption Annotations: Multiple human-annotated captions per video (approximately 40 captions per video)
- Average Video Duration: 10.2 seconds per clip
- Content Diversity: Wide range of activities, environments, and scenarios
- Annotation Quality: High-quality human descriptions with varied linguistic expressions

Dataset Partitioning Strategy: Our dataset division follows established practices while ensuring adequate representation across different video categories:

- Training Set: 1,200 videos (60.9% of total dataset)
    - Provides sufficient diversity for model learning
    - Ensures comprehensive coverage of different scene types and activities
    - Maintains balanced representation across various video categories
- Validation Set: 100 videos (5.1% of total dataset)
    - Used for hyperparameter tuning and model selection

    - Enables monitoring of training progress and overfitting detection
    - Provides intermediate feedback during the training process
- Test Set: 670 videos (34.0% of total dataset)
    - Reserved for final model evaluation
    - Ensures unbiased performance assessment
    - Maintains sufficient size for statistical significance

A.2 Comprehensive Preprocessing Pipeline

Video Preprocessing:

1. Temporal Standardization: All videos undergo conversion to a uniform 30 frames per second to ensure consistent temporal representation across the dataset
2. Resolution Normalization: Videos are resized to consistent dimensions to maintain uniform input characteristics
3. Frame Sampling: Implementation of uniform temporal sampling strategy to extract 80 representative frames per video
4. Quality Assurance: Removal of corrupted or invalid video files to maintain dataset integrity

Caption Preprocessing:

1. Text Cleaning: Systematic removal of special characters, punctuation marks, and non-alphabetic symbols
2. Case Normalization: Conversion of all text to lowercase for consistency and to reduce vocabulary complexity
3. Tokenization: Application of word-level tokenization to convert text into processable sequences
4. Vocabulary Construction: Building a comprehensive vocabulary using the top 1,500 most frequent words across all captions
5. Special Token Integration: Addition of essential tokens including:
    - <START>: Caption beginning marker
    - <END>: Caption termination marker
    - <UNK>: Unknown word placeholder for out-of-vocabulary terms
6. Sequence Conversion: Transformation of cleaned text into sequences of word indices suitable for decoder input

Feature Extraction Preprocessing:

1. Frame Standardization: Resizing of sampled frames to 224×224 pixels for CNN compatibility
2. Normalization: Application of ImageNet statistics for proper feature extraction
3. Scene Context Preparation: Extraction and processing of middle frames for scene classification
4. Batch Organization: Efficient organization of preprocessed data for training pipeline

B. Two-Phase Training Strategy

Our training methodology employs a carefully designed two-phase approach that optimizes the visual captioning pipeline by progressively building model capabilities from fundamental feature extraction to sophisticated caption generation.

B.1 Phase 1: Encoder Pre-training

Objective: Establish robust visual feature extraction and temporal encoding capabilities before introducing the complexity of caption generation.

Training Specifications:

- Frame Processing: 80 uniformly sampled frames per video
- Visual Encoder: VGG16 architecture with FC7 feature extraction (4096-dimensional features)
- Training Duration: Approximately 50 epochs on the complete training dataset
- Dataset Path: data/training_data
- Focus Areas:
    - Spatial feature extraction optimization
    - Temporal sequence learning through LSTM encoder
    - Scene context feature integration
    - Attention mechanism initialization

Phase 1 Benefits:

1. Feature Quality: Ensures high-quality visual representations before caption generation
2. Temporal Understanding: Establishes robust temporal dependencies in video sequences
3. Computational Efficiency: Reduces overall training time by pre-optimizing encoder components
4. Stability: Provides stable feature representations for subsequent decoder training

B.2 Phase 2: End-to-End Training

Objective: Joint optimization of the complete encoder-decoder architecture with attention mechanisms for coherent caption generation.

Training Specifications:

- Total Epochs: 150 epochs with early stopping capabilities
- Validation Split: 15% of training data for continuous monitoring
- Maximum Sequence Length: 10 words per generated caption
- Vocabulary Size: 1,500 words covering diverse descriptive terms
- Decoder Architecture: LSTM with 512 hidden units
- Training Techniques:
    - Teacher forcing for efficient learning
    - Scheduled sampling to improve inference performance
    - Attention mechanism optimization

Advanced Training Strategies:

1. Teacher Forcing: During training, the decoder receives ground-truth previous words as input, accelerating the learning process
2. Scheduled Sampling: Gradual introduction of model-generated words during training to bridge the gap between training and inference
3. Attention Optimization: Joint optimization of attention weights with caption generation for improved alignment
4. Scene Context Integration: Incorporation of scene-level features throughout the decoding process

C. Training Configuration and Hyperparameters

C.1 Optimization Parameters

Core Training Configuration:

- Batch Size: 320 (optimized for GPU memory utilization and training stability)
- Learning Rate: 0.0007 with optional decay scheduling for fine-tuned optimization
- Optimizer: Adam optimizer with β_1=0.9, β_2=0.999, ε=1e-8
- Regularization: Dropout rate of 0.5 applied to LSTM layers during training
- Gradient Management: Gradient clipping implemented to prevent exploding gradients
- Decoder Search Strategy: Beam search with width 3 for improved caption quality
- Model Persistence: Regular model checkpoints saved to model_final/ directory

C.2 Advanced Training Techniques

Regularization Strategies:

1. Dropout: Applied to LSTM layers to prevent overfitting
2. Weight Decay: L2 regularization for parameter stability
3. Early Stopping: Validation-based early stopping to prevent overfitting
4. Gradient Clipping: Prevents exploding gradients in recurrent layers

Learning Rate Scheduling:

- Initial Rate: 0.0007 for stable initial learning
- Decay Strategy: Exponential decay based on validation performance
- Adaptive Adjustment: Learning rate reduction on plateau
- Minimum Threshold: Lower bound to prevent excessive decay

Batch Processing Optimization:

- Dynamic Batching: Efficient batching for variable sequence lengths
- Memory Management: Optimized GPU memory utilization
- Parallel Processing: Multi-GPU support for accelerated training
- Data Loading: Efficient data pipeline for continuous training

D. Comparative Performance Analysis

D.1 Benchmark Comparison with Existing Approaches

Our visual captioning system with scene context integration demonstrates significant improvements over existing state-of-the-art methods on the MSVD dataset. The following comprehensive comparison highlights the effectiveness of our approach:

Comparative Performance Table:

-

Model	Year	Architecture Highlights	Performance
LRCN	2015	Early encoder-decoder using C3D+ LSTM	~60% accuracy

S2VT	2015	Sequence-to-sequence model with temporal attention	BLEU-4: 28.1
HRNE	2016	Hierarchical Recurrent Neural Encoder	BLEU-4: 29.2
SCN-LSTM	2017	LSTM with semantic composition-ality	BLEU-4: 30.5
SeFLA	2020	Uses semantic and spatial features	BLEU-1: 50.1,CIDEr:0.94

D.2 Our Model Performance Analysis

Comprehensive Performance Metrics:

Decoding Strategy Comparison:

- **Greedy Search Performance**:
    - Accuracy: 72%
    - BLEU-1: 0.84
    - BLEU-4: 0.32
    - METEOR: 0.25
    - CIDEr: 0.36
- **Beam Search Performance**:
    - Accuracy: 74% (+2% improvement)
    - BLEU-1: 0.86 (+2.4% improvement)
    - BLEU-4: 0.35 (+9.4% improvement)
    - METEOR: 0.27 (+8.0% improvement)
    - CIDEr: 0.39 (+8.3% improvement)

Performance Advantages:

1. Significant Accuracy Improvement: Our model achieves 74% accuracy with beam search, representing a substantial 14% improvement over LRCN's 60% accuracy baseline.

2. Enhanced BLEU-4 Scores: With a BLEU-4 score of 0.35, our model outperforms S2VT (28.1), HRNE (29.2), and SCN-LSTM (30.5) by significant margins.

3. Consistent Metric Improvements: Superior performance across all evaluation metrics demonstrates the robustness and effectiveness of our approach.

4. Decoding Strategy Effectiveness: The consistent improvement of beam search over greedy search across all metrics validates our decoding strategy selection.

CONCLUSION:

This Project presents a deep learning-based approach for visual captioning, specifically focused on generating descriptions for short action-oriented video clips. The proposed system integrates spatial features extracted using a pre-trained CNN with temporal modeling via an LSTM-based encoder-decoder architecture enhanced by an attention mechanism. This combination enables the model to capture both visual appearance and sequential dependencies for more accurate caption generation.

To further improve semantic grounding, we simulate the inclusion of a scene context module inspired by Places365, providing environmental cues that help refine the captioning output. We also compared greedy and beam search decoding strategies and observed that beam search produced more fluent and context-aware captions.

Evaluated on the MSVD dataset, our method demonstrates strong performance and generalization despite data limitations. The results highlight the effectiveness of combining attention and scene-level context in enhancing caption quality. This work contributes toward building more context-aware and semantically rich visual captioning systems, with future potential in large-scale video indexing, retrieval, and assistive technologies.

REFERENCE:

[1] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in Proc. CVPR, 2015.

[2] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence – video to text,” in Proc. ICCV, 2015.

[3] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” in Proc. ICCV, 2015.

[4] Y. Pan, T. Yao, H. Li, and T. Mei, “Hierarchical recurrent neural encoder for video representation with application to captioning,” in Proc. CVPR, 2016.

[5] Z. Gan, C. Gan, X. He, J. Gao, and L. Deng, “Semantic compositional networks for visual captioning,” in Proc. CVPR, 2017.

[6] H. Chen, J. Wang, W. Li, and Y. Liu, “SeFLA: Semantic features for latent attention in video captioning,” ACM Trans. Multimedia Comput. Commun. Appl. (TOMM), vol. 16, no. 2, pp. 1–19, 2020.

[7] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in Proc. ICML, 2015.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in Proc. ICCV, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. CVPR, 2016.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, 2015.

[14] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” IEEE TPAMI, vol. 40, no. 6, pp. 1452–1464, 2017.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Proc. NeurIPS, 2014.

[16] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.