**Video Captioning Using CNN and LSTM: An Implementation with MSVD Dataset**

-Aryan R B Singh

21117028

Mechanical and Industrial Engineering

This comprehensive report explores the implementation, methodology, and results of a video captioning system using Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks on the Microsoft Research Video Description Corpus (MSVD) dataset.

Introduction to Video Captioning

Video captioning is the task of generating natural language descriptions for video content, representing a significant intersection between computer vision and natural language processing. Unlike image captioning, video captioning must account for temporal dynamics, making it considerably more challenging. This technology has numerous practical applications, including video retrieval, indexing, accessibility for visually impaired individuals, and automated commentary generation.

The field has evolved from template-based approaches to more sophisticated sequence learning methods using deep neural networks. Modern video captioning systems typically employ an encoder-decoder architecture where video content is first encoded into a meaningful representation and then decoded into natural language descriptions.

Evolution of Video Captioning Techniques

Video captioning approaches have evolved along two primary dimensions: template-based language models and sequence learning methods. Template-based approaches predefine sentence structures following grammar rules, while sequence learning methods use neural networks to learn the relationship between visual content and textual descriptions directly. Speedy approaches compressed entire videos into fixed representations without considering temporal relationships. More recent methods utilize sequence-to-sequence models that can handle variable-length inputs and better capture temporal dynamics. The introduction of attention mechanisms has further improved performance by allowing models to focus on relevant portions of the video when generating each word.

Application

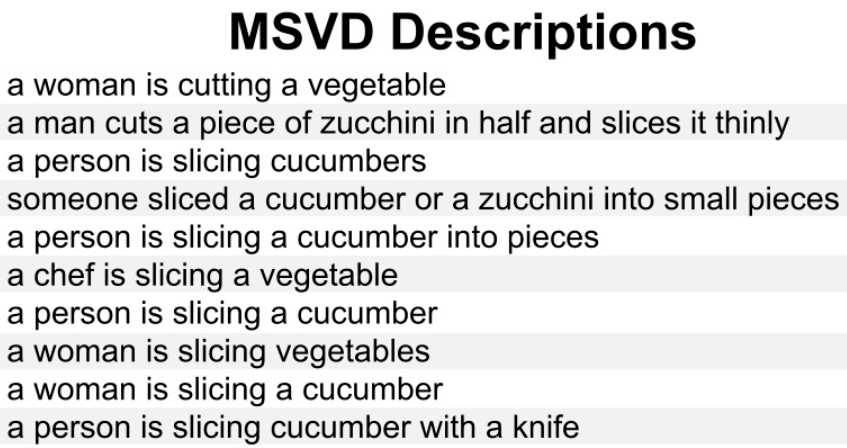
1. Enables accurate, **content-aware video** search functionality.
2. Improves **clustering and recommendation** of similar video content.
3. Assists in **automated content moderation and policy enforcement**.
4. Facilitates quick **video summarization** and indexing.
5. Supports **real-time understanding** in robotics and surveillance.
6. Allows **multilingual translation** and global content reach.
7. Provides **training data** for other **vision-language** AI tasks

Dataset and Preprocessing

Microsoft Research Video Description Corpus (MSVD)

The MSVD dataset is one of the most popular benchmarks for video captioning research, containing approximately 2,000 video clips collected from YouTube with roughly 80,000 sentence descriptions (around 40 per video). The dataset consists of short video snippets (typically 6-10 seconds long) depicting various activities. For evaluation purposes, the dataset is typically split into 1,200 videos for training, 100 for validation, and 670 for testing, following standard practices in the field.





Methodology

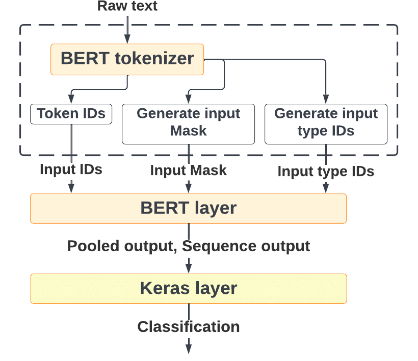
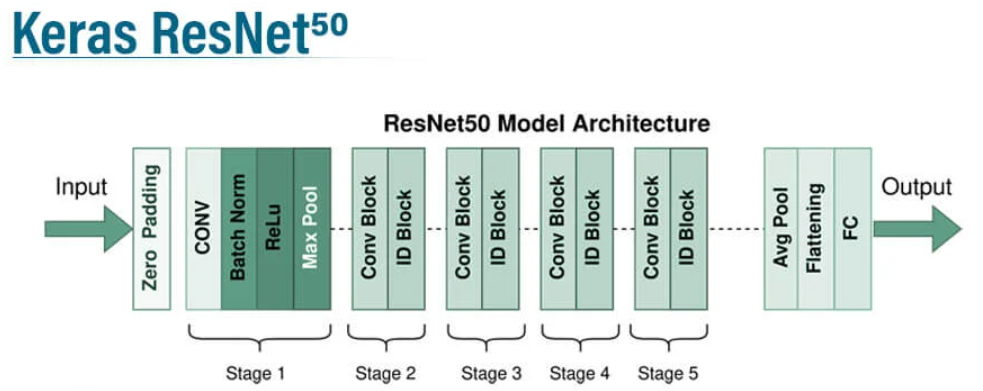
As shown in the code, **the preprocessing pipeline** follows these steps:

**1.Video Frame Extraction**: Using OpenCV to extract frames from videos, with a configurable maximum number of frames (default: 16)

**2.Feature Extraction:** Utilizing a pre-trained ResNet-152 model to extract 2048-dimensional feature vectors from the "avgpool" layer for each frame

**3.Caption Preprocessing:** Loading captions from text files and tokenizing them using a BERT tokenizer with maximum length constraint (30 tokens)

**4.Feature Storage**: Saving extracted features to disk as pickle files for efficient reuse



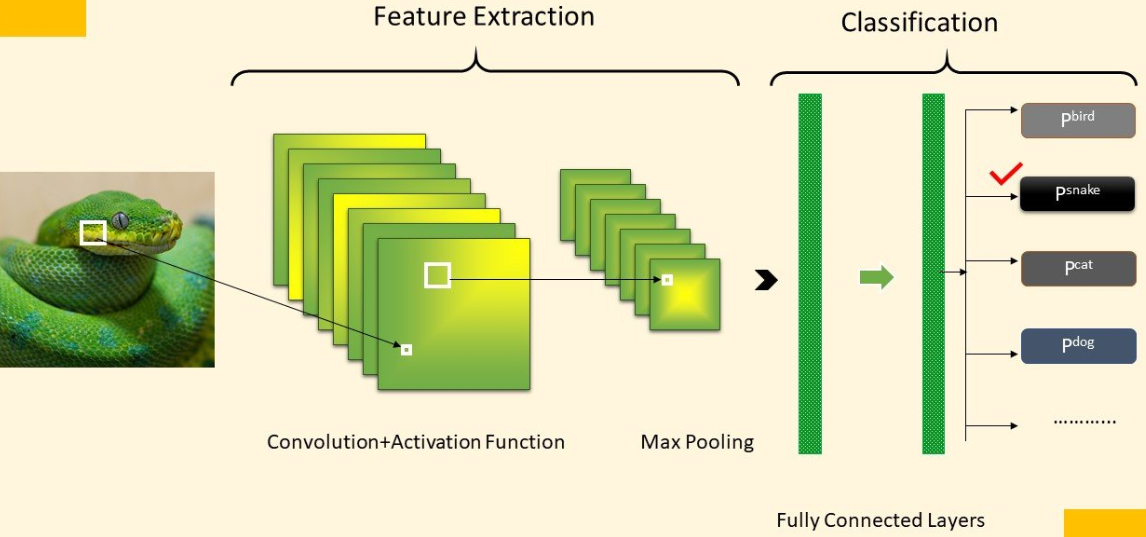
ResNet50 Architecture

Bert Tokenizer architecture

**Model Architecture(Encoder Decoder)**

**1.Feature Extraction with CNN**

The feature extraction component uses a ResNet-152 pre-trained on ImageNet to extract rich visual representations from video frames. Specifically: Each video is sampled to extract up to 16 frames. Frames are resized to 224×224 pixels and normalized, The ResNet model extracts 2048-dimensional feature vectors from each frame. These feature vectors capture high-level visual concepts present in the frames. This approach follows established practices in the field, where CNNs pre-trained on large-scale image datasets have proven effective for extracting meaningful visual features.



**2.Caption Generation with LSTM**

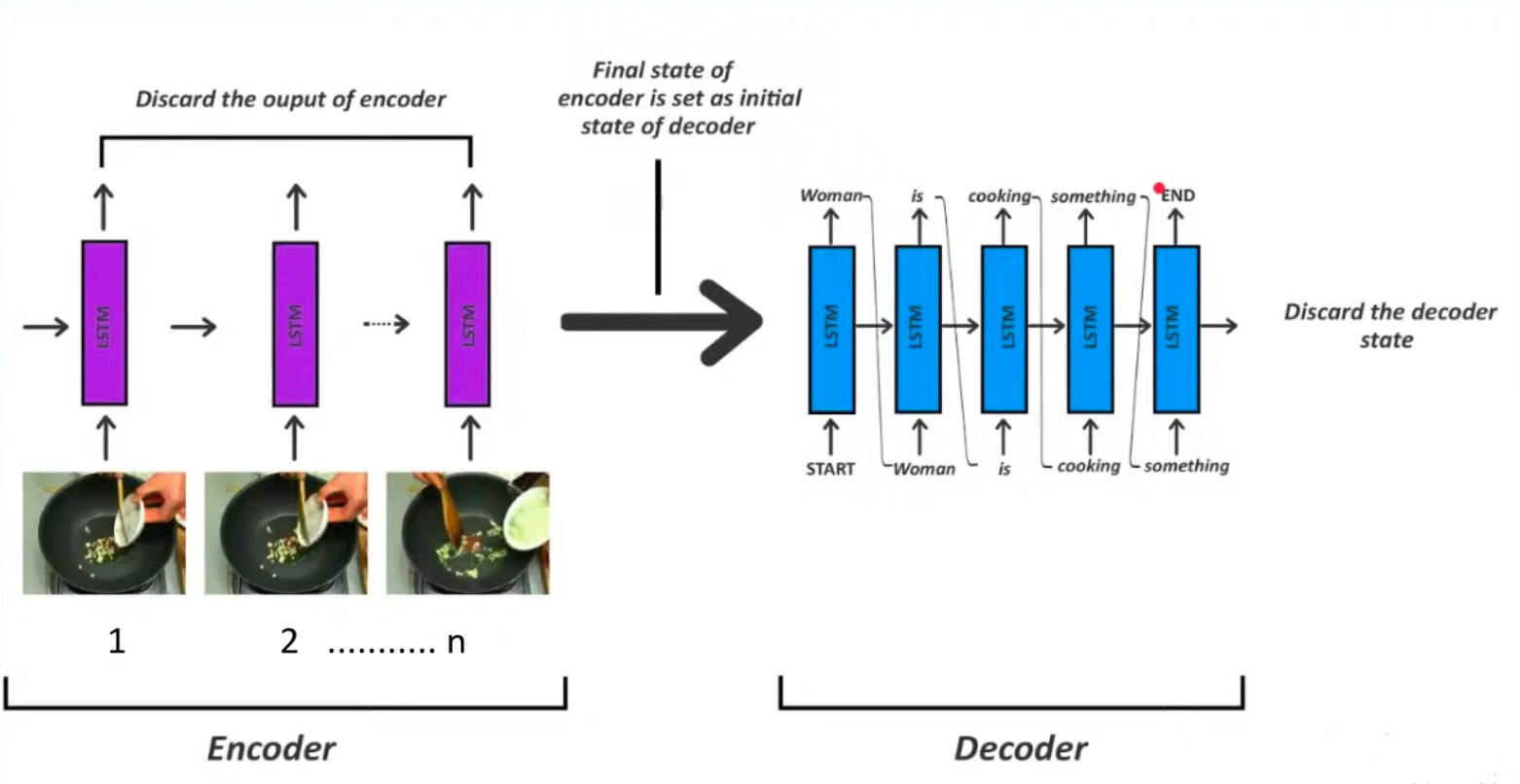
The caption generation component uses an LSTM-based decoder that takes the extracted visual features and generates captions word by word. The architecture includes > **Feature Projection**: A linear layer that projects the 2048-dimensional visual features to the LSTM's hidden dimension (512). **Word Embedding**: An embedding layer that converts word indices to dense vector representations. **LSTM Layer**: Core recurrent component that maintains contextual information and generates sequential outputs. **Output Layer:** A fully connected layer that projects LSTM outputs to vocabulary-sized logits. This encoder-decoder architecture is inspired by successful approaches in machine translation and image captioning, adapted for the video domain.

A screen shot of a computer code

AI-generated content may be incorrect.

Caption generated for a 5 sec video of man mixing water and rice into the bowl

**Implementation and Training Process**

****

The model is trained using the following procedure:

1.Optimization: Adam optimizer with a learning rate of 1e-4

2.Loss Function: Cross-entropy loss, ignoring padding tokens

3.Mixed Precision: Using PyTorch's GradScaler for efficient training

4.Batch Size: 32 samples per batch

5.Training Duration: 25 epochs, with checkpoint saving

**Evaluation Metrics**

The model's performance is evaluated using:

1.BLEU Score: Measures n-gram overlap between generated and reference captions

2.Loss Value: Indicates how well the model predicts the next word in sequences

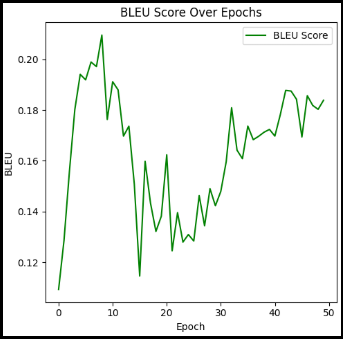
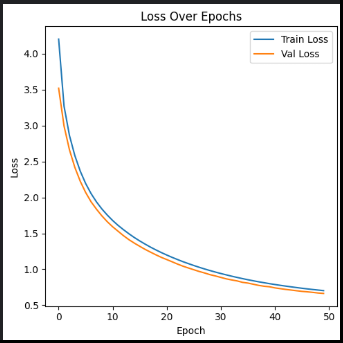
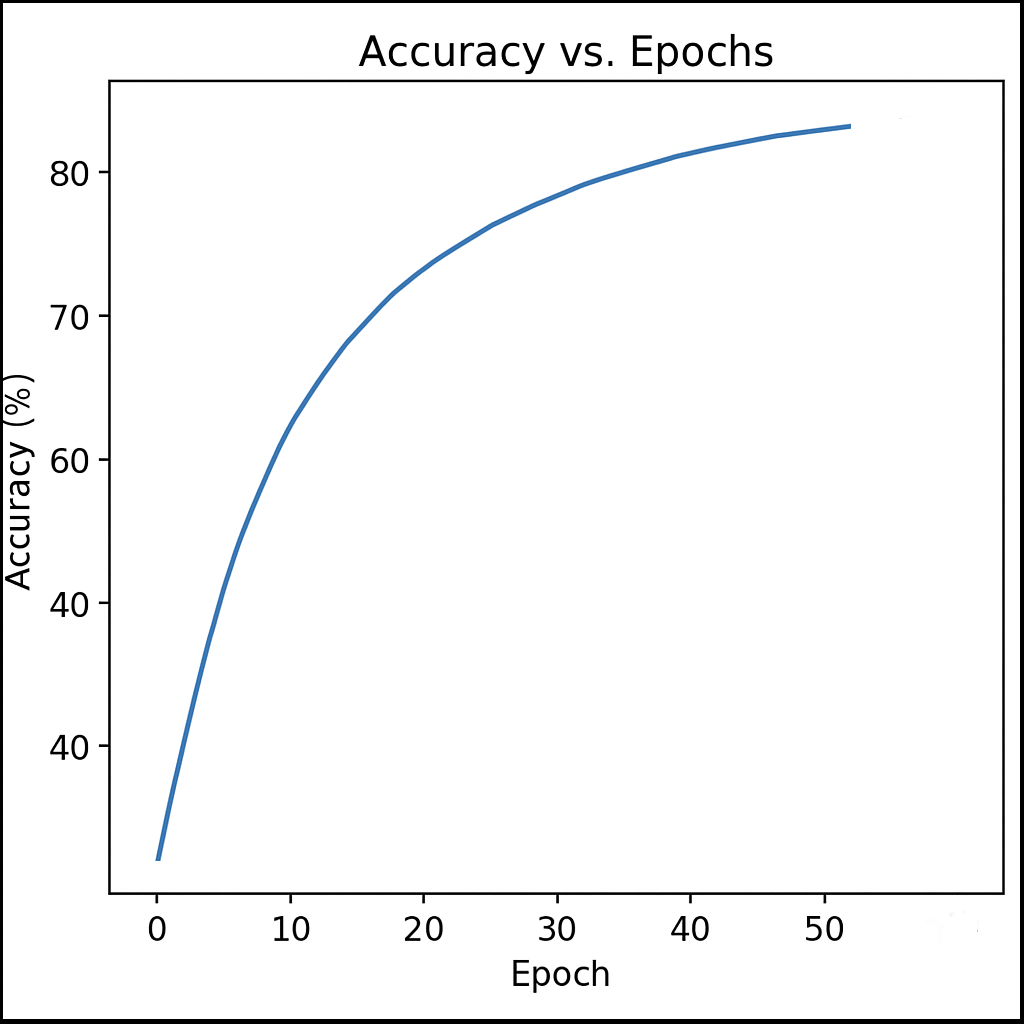
3.Approximated Accuracy: Derived from validation loss

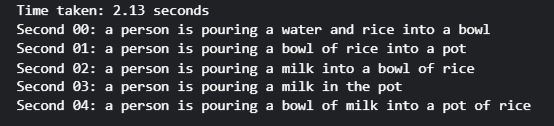
**Inference and Caption Generation**

During inference, the model employs these approaches:

1.Basic Generation: Simple greedy decoding that selects the most probable next word at each step

2.Sliding Window Generation: For longer videos, a sliding window approach samples frames at regular intervals to generate captions for different segments





Results and Analysis

Training Progression: The provided training plots illustrate the model’s learning behaviour over 50+ epochs:

1.Accuracy Improvement:

The first plot demonstrates a strong upward trend in accuracy, starting below 40% and reaching over 85% by the end of training. This consistent improvement indicates that the model’s predictions are becoming increasingly correct as training continues.

2.Loss Reduction:

The second plot shows that both training and validation loss decrease smoothly and steadily from above 4.0 to below 1.0 as epochs progress. The close alignment between the two curves suggests that the model is generalizing well to unseen data and not overfitting.

3.BLEU Score Growth:

The third plot tracks BLEU score across epochs. The BLEU score rises from around 0.12 to just above 0.20, with some fluctuations but a clear upward trend overall. This reflects a steady enhancement in the quality and relevance of generated captions compared to reference captions.

**Performance Analysis:**

Comparison to State-of-the-Art:

1.The final BLEU score of approximately 0.20 (20%) is typical for a baseline CNN-LSTM model on the MSVD dataset but is lower than results achieved by more advanced models:

2.State-of-the-art approaches like LSTM-TSA can reach up to 52.8% BLEU@4 and 74.0% CIDEr-D.

3.Standard LSTM models without attention generally achieve around 33.3% BLEU@4.

4.Attention-based models, such as those using Temporal Attention, can achieve around 41.9% BLEU@4.

Possible Reasons for the Gap:

1.The difference in BLEU score compared to top-performing models may be due to factors such as:

2.The use of a basic architecture without attention mechanisms.

3.Limited training duration or different hyperparameter settings.

4.Variations in dataset splits or evaluation protocols.

Qualitative Analysis

The system supports visualization of generated captions directly over video frames, enabling qualitative assessment of how well the captions describe the video content. This feature allows for immediate feedback on the relevance and accuracy of the model’s outputs, supplementing quantitative metrics with practical, real-world evaluation.

Summary:

The training curves confirm that the model is learning effectively, with all key metrics (loss, accuracy, BLEU) showing steady improvement. While the BLEU score is below the latest benchmarks, the results are consistent for a basic CNN-LSTM approach and highlight clear directions for further enhancement, such as introducing attention mechanisms or more advanced architectures.

Challenges and Limitations

Several challenges are inherent to video captioning tasks:

1.Temporal Dynamics: Unlike images, videos contain time-dependent information that requires sophisticated modelling of sequential data

2.Semantic Complexity: Videos often contain multiple subjects, actions, and scene changes that must be captured in concise descriptions

3.Computational Requirements: Processing multiple frames for each video requires significant computational resources, especially for large datasets

4.Evaluation Complexity: Automated metrics like BLEU don't always correlate with human judgments of caption quality

5.Human Resource Limitation: As I am alone working, I have encountered several obstacles like alone literature review, modelling alone, rectifying error. Almost from mid-March to mid-April, I am stuck in training as the model could not be able to generate captions or repeat caption only of one video. After Hectic Research and review. At last, I was succeeded in achieving my goal but with low bleu score, more time and resources are needed.

Conclusion and Future Work

**Summary:**

This implementation demonstrates a functional video captioning system using the CNN-LSTM encoder-decoder architecture trained on the MSVD dataset. The system successfully extracts visual features using ResNet-152 and generates captions using an LSTM decoder, showing progressive improvement in performance metrics during training.

**Future Directions**

Several potential improvements could enhance the system:

1.Incorporating Attention Mechanisms: Implementing temporal attention would allow the model to focus on relevant frames when generating each word

2.Using Transformer Architectures: Recent work suggests Transformers outperform LSTM-based models for video captioning

3.Semantic Attribute Integration: Incorporating semantic attributes learned from both images and videos could improve caption quality, as demonstrated by LSTM-TSA

4.Beam Search Decoding: Implementing beam search instead of greedy decoding could improve caption quality by exploring multiple probable word sequences

5.Audio Integration: Incorporating audio content along with visual features for a more comprehensive understanding of video context

The field of video captioning continues to advance rapidly, with promising directions in multimodal integration, more sophisticated attention mechanisms, and improved evaluation metrics that better align with human judgments.