

Sentence Vectors Enable GRU And Multi-Headed Attention-Based Video Captioning Framework For Specifying Events In MSVD Video Dataset

K. Annapoorneshwari Shetty^{1*}, Subrahmanya Bhat²

^{1*}Research Scholar Srinivas University Email ID: annapoorna.nikesh@gmail.com Orcid ID(if you have): 0000-0003-1900-256X

²Professor Srinivas University Email ID: sbhat.ccis@srinivasuniversity.edu.in Orcid ID(if you have): 0000-0003-2925-1834

Abstract—The fast growing digital culture has attracted people's attention and motivated them to interact with fascinating multimedia materials, including images, videos, voice notes and messages. Video is the most often used digital material format globally; it is used for marketing for companies, education, proof of concern, memorial ceremonies, and amusement among other purposes. The primary goal of video/image captioning is to represent the dynamics of a video clip using simple and understandable conversational language. The motivation for captioning comes from its ability to enhance video accessibility for those who are deaf or hard-of-hearing, improve concentration and memory of information and allow viewing in sound-sensitive environments. In this proposed framework, we demonstrate how to create captions for videos using Gated Recurrent Unit (GRU) along with a Sentence Vector Branch (SVB) and skip connections employing two word embedding methods: Bidirectional Encoder Representations from Transformers (BERT) and Embeddings from Language Model (ELMO). The encoder section of the model uses skip connections to address the issue of vanishing gradients and the sentence vector branch is used to improve the efficiency of the encoder-decoder architecture. The decoder generates a suitable description using the sentence vector branch together with past generated words. All of our experiments make use of the Standard MSVD databases. BLEU@4, METEOR, ROUGE and CIDEr measures have been applied for the performance assessment. Comparative analysis of BERT and ELMO word embeddings reveals their respective strengths. Features are extracted from pre-trained CNNs NASNet-Large, VGG16, Inceptionv3, DenseNet-169 and ResNet50. In the MSVD dataset, BERT beats ELMO in the case of the GRU-based model, whereas ELMO beats BERT in the case of the LSTM-based model; among LSTM and GRU, LSTM outperforms GRU. DenseNet-169 beats every feature extraction method used in the MSVD dataset.

Keyword – Sentence Vector Branch, Skip Connection, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN).

I. INTRODUCTION

The internet has fundamentally transformed the way humanity leads their lives. A wide range of individuals, including both school students and high-ranking personnel of global corporations rely on the internet. The proliferation of digital culture has engrossed individuals and heightened their propensity to engage with social media, navigate the internet and retrieve images, videos, text and other multimedia content. In contemporary times, the usage of mobile phones has become ubiquitous. With the growth of internet infrastructure and the increasing number of smartphone users, the volume of digital content on social networks such as Instagram and Facebook has also surged. Today, video viewing is the favored mode of material consumption. Approximately 96% of individuals in 2020 reported an increase in their online video consumption. Furthermore, out of every 10 individuals 9 expressed a desire to increase their video consumption. Indeed, by 2022 it is projected that the typical individual will allocate 100 minutes each day to viewing online videos [1]. Thus, both presently and in the next years the volume, variety

of videos available on YouTube, Instagram, and other platforms will experience a billion-fold growth.

The overall video captioning framework is explained in Fig. 1 [2].

The efficacy of text-based video retrieval can be enhanced by the automated captioning of the video, while comprehending the action and events depicted in it. Contemporary times have witnessed an unprecedented abundance of visual data and there is a universal affinity for visual content. Thus, a robust method is necessary to handle this data. In contrast to humans, computers do have challenges in understanding visual information and automatically generating its precise interpretation. To machines people, possess superior capabilities, efficiency and accuracy in carrying out these tasks. The rate of video uploads to YouTube exceeds five hundred hours per minute [3]. Observing them all is almost literally unfathomable. By means of online video-streaming and sharing platforms that operate 24/7, vast amounts of material are effectively stored and accessed. Given the proliferation of video content, the analysis and filtration of offensive material has become an essential requirement. Engaging in manual analysis of such large amounts of data is becoming unfeasible and necessitates computational assistance, which can be offered by computer vision by producing precise descriptions of the videos. Efficient analysis of descriptions produced from video captioning can facilitate the automation and regulation of video content.

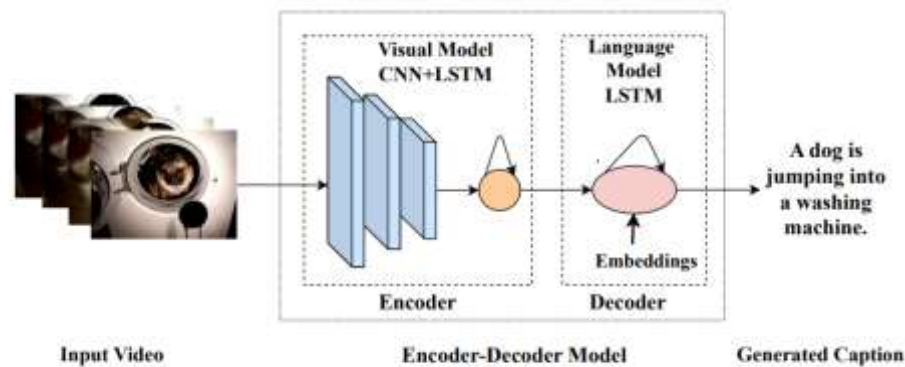


Fig. 1. The overview video captioning framework.

Efforts to improve video captioning have increased in recent years, with the most promising approach being encoder-decoder neural networks [4]. These networks process video data by combining recurrent neural networks (RNNs) for phrase encoding and decoding with 2D or 3D convolutional neural networks (CNNs) [5], [6]. The addition of subtitles transforms a video from a sequence of still images into a narrative. The process of video captioning follows a sequence-to-sequence architecture, which is employed to find word embedding using BERT [7], ELMO [8], and to extract image features using VGG16 [9], ResNet50 [10], DenseNet169 [11], NASNet-Large [12], InceptionV3 [13], and other model architectures. Video processing is more difficult than image processing due to the presence of two distinct kinds of information: spatial and temporal. A phrase vector branch and skip connections [14] form the basis of the model's encoder-decoder design. To construct the encoder, we used GRU, which takes picture features taken from video frames and uses them to extract temporal information. A multi-head attention [15] layer decoder is employed to generate a contextually rich description of the movie by selectively retrieving information from relevant representational dimensions. The vanishing gradient problem is one potential outcome of using a deep network; to circumvent this, skip connections have been implemented. To help the decoder generate a suitable description, sentence vectors have been utilized.



Sentences:

- A man lights a match book on fire.
- A man playing with fire sticks.
- A man lights matches and yells.

Fig. 2. An example of video captioning from MSVD dataset. There are different captions possible for different image frames. Understanding the activity in the video requires collecting different frames and extracting spatial and temporal information.

LSTM is a variant of RNN that addresses the issue of short-term memory [16], which is a problem with RNN in general. GRU is an optimized and less heavy variant of LSTM. Update and Reset are the two gates that make up GRU. You can learn how much of your past to remember by using the Update Gate, and how much to forget by using the Reset Gate.

A. Motivation

In today's technological world, video processing is essential for a number of reasons, not the least of which is the prevalence of social media platforms, which rely on videos as their primary medium for disseminating information and entertainment. Surveillance, proof-of-concern, social, and instructional videos, among others, have been increasingly important in permeating our surroundings and everyday lives. With the ever-increasing volume of video material, video captioning has become an indispensable tool for tasks such as object recognition, video summarizing and description, and data searching. It can also enhance military operations and surveillance by identifying dangers and providing soldiers and weaponry with the information they need to eliminate them, and it can help the visually impaired by explaining their surroundings.

B. Contribution

- The new model is proposed with modified version of LSTM called GRU has been used.
- Five different Convolutional Neural Network(CNN) are used for video feature extraction namely InceptionV3, VGG16, ResNet50, NASNetLarge, and DenseNet-169 model.
- The two well-known Word embedding, BERT, and Elmo are used.
- The whole framework is proposed with unique technique called skip connections to avoid the vanishing gradient problem.
- The model is also utilized with new technique called Sentence Vector Branch (SVB) to upgrade the performance of the Encoder-Decoder architecture.
- The proposed model experimented with Standard MSVD Dataset.
- The proposed model is compared with LSTM model on MSVD dataset.

II. BACKGROUND AND RELATED WORKS

The study of existing literature addressing the problem of video captioning is explained. The video consists of a series of frames, and captioning creates a series of texts. Therefore, a sequence-to-sequence architecture is utilized to caption videos and the process of captioning videos is likewise a sequence to sequence one.

In the study [17], [18], the author employs a deep Long Short-Term Memory (LSTM) to decipher the target sequence from the vector. This is achieved by utilizing a multilayered

LSTM to transform the input sequence into a vector of given dimensions. The main discovery was that the LSTM model achieved a BLEU score of 34.8 on the whole test set when translating from English to French using the WMT-14 dataset. The LSTM model's BLEU score was penalized when processing items that were not part of the standard vocabulary. They compared their multilayered Long short-Term memory model with SMT system model which don't perform well compared to multilayered Long Short-term Memory model as it gets Bleu score of 33.3 on the same dataset.

In the study [19], the seq2seq architecture was used for video captioning events. Several iterations of the seq2seq architecture have been widely adopted in recent years. Other approaches, also known as hierarchical boundaries, offer a recurrent video encoding method that could identify and take advantage of the video's hierarchical structure. The authors suggest a novel LSTM cell that can recognize abnormalities in frames or segments and adjust the encoding layer's temporal connections appropriately. This strategy differs from the traditional encoder-decoder method, which uses a recurrent layer to encode a video iteratively.

According to seq2seq research from earlier papers, the decoder cells use the encoder's predefined output and the contextual information of the prior word to generate the next word. As a result, the output that is generated frequently depends more and more on the previous hidden cell state, and the broader meaning of encoded information is frequently lost. The attention mechanism notion has now emerged and addresses the drawbacks of the seq2seq design. The model may maintain context from start to finish thanks to the attention approach.

The authors of [20] introduced a paradigm for video captioning that incorporates stacked attention and semantic hard pull. This model utilizes Semantically Sensitive Video Captioning (SSVC), which is the distinctive architecture shown in this work. It alters the process of context formation. This process employs two novel techniques: "attention" and "spatial hard pull." The authors prioritize both quantitative and qualitative analysis of their model due to the absence of precise metrics for assessing video captioning methods. Common computerized scoring metrics have drawbacks that SS Score eliminates. The performance of cutting-edge architectures is enhanced by using the novelties listed above, according to this article.

A very deep convolutional network for large-scale image recognition is shown in the article. In a scenario where numerous photos must be identified, this study investigates the impact of the convolutional network's depth on accuracy. A detailed examination of networks with increasing depth using a model with tiny (3×3) convolution filters is the main contribution. This illustrates how improving the depth to 17–20 layers can greatly improve upon current configurations.

The authors in [9] proposed a model of deep residual learning for image recognition. It is more challenging to train deep neural networks. The approach provides a residual learning approach to make training networks much more profound than those previously utilized. It intentionally reformulates the layers so that they learn residual functions concerning the layer inputs rather than learning unreferenced functions.

The model [19] employs a sequence-to-sequence-video-to-text; they proposed a novel sequence-to-sequence paradigm to generate video subtitles. They made use of LSTM and RCNN, which have been shown to function at the forefront of creating image captions. To generate a description of the activity in the video clip, their LSTM framework learns to link a sequence of video frames to a string of words. Video-sentence pairs are used to teach it. The model can naturally learn the sequence model of the generated sentences or a language model [21], as well as the temporal structure of the succession of frames. They tested multiple iterations to leverage diverse visual features on two distinct datasets and a typical batch of videos.

A. OUTCOME OF LITERATURE REVIEW

- The most known model used for captioning of video tasks is the Sequence-to-sequence model. However, to increase the efficiency in prediction better model is proposed.
- To improve the content of captioning of the video LSTM model is used. However, GRU can also be used to address some of the issues in LSTM.
- The context of the previous word and the encoder's fixed output is used by the decoder cells to construct the subsequent word in the encoder-decoder paradigm.

III. METHODOLOGY

Video processing [22] is a complex task involving image processing to gather spatial information as well as temporal information from video frames. Thus it is important to capture both dimensions using deep learning models. CNN models have been state-of-the-art for capturing information in the images and many complex computer vision problems have been solved using deep CNN models. We have approached the problem by employing CNN models for capturing spatial information and also using GRU for extracting temporal information. Artificial neural networks, a machine learning technology, include convolutional neural networks. Although they have been around for a long time, their popularity has lately increased due to their success in image recognition. A valuable tool that we may utilize to handle any type of data where the convolution operation can be applied is a convolutional neural network.

The effectiveness of CNN [23] is underpinned by its capacity to effectively evaluate vast amounts of data, encompassing images and videos. Their primary utility is classifying images, object localization, and extracting image features like edges or corners. Usually, they consist of one or more hidden layers, and each one of those layers has a collection of trainable filters. These filters can detect features and attributes within the image required to solve the computer vision task. Finally, it's important to note that the ReLU activation function is often used following the convolutional layer. The pooling layer, which applies filters like the convolutional layer but calculates the maximum or average item rather than convolution, frequently comes next.

VGG16 is relatively less deep model but very effective in detecting image features. We have also used very deep CNN models such as ResNet50 and NASNet-Large with 100+ convolution layers. These models have been very effective in capturing the spatial information.

To capture temporal information of the video GRU is used [24]. We had used the state-of-the-art BERT word embedding technique for capturing contextual information in the captions. In this project, we have done the following to generate our required output as for first we had extracted the frames from the video and then from that extracted frames we had extracted the frame-level features using RNN models or CNN models After that we had used encoder and other techniques for getting the contextual representation of our input. After that we have to generate an accurate description which can be done with the help of decoder and then training the model and evaluating the model on different test set.

STEPS FOR THE DESIGN:

- We have extracted the frames from the videos in the dataset as shown in Fig.3.
- Extracted frames have been given for the feature extraction using pre-trained CNN models.
- After feature extraction step, for better contextual representation of video encoder-decoder architecture is used.
- Video captions have been embedded with state-of-the-art word embedding techniques BERT and ELMO.

- GRU and LSTM-based models have been trained on the extracted features and embedded captions.
- Testing results have been evaluated qualitatively and quantitatively using MS-COCO caption evaluation model by analyzing BLEU [25], METEOR [26], ROUGE [27] and CIDEr [28] scores.

A. MSVD Dataset

In any machine learning or deep learning problem, the dataset plays a very critical role. Deep learning models are data-hungry and require a lot of data for producing an artificial intelligent system. The models learn from the data samples present in the dataset and to generalize the machine learning model, the dataset should include all kinds of real-world data samples. Only a good-quality dataset can produce a sound and robust machine learning model.

One of the widely used datasets for video captioning is MSVD (Microsoft Research Video Description Corpus) dataset [29]. We have employed the MSVD dataset that has 1970 small YouTube video clips and captions associated with them. These video clips have only single activity and do not contain audio. The captions in the dataset are available in various languages and we have chosen to work with English annotations. Video clips have been recorded with on an average 30 fps and there are on an average 41 captions for each video clip.

B. Feature Extraction

To begin with, the dataset is preprocessed to remove any absent entries and prepare the captions to be in an appropriate format. The dataset has been divided into training-validation-test splits in the proportion of 75 : 5 : 20. These splits comprise videos and captions for training the machine learning model. Video is a sequence of image frames; depending on the recording frame rate (i.e., usually 30 fps), it may contain hundreds of frames. Every video is represented by a specified number of frames ($N = 28$) extracted from it. OpenCV library is used for extracting frames from the videos and the interval between the frames is computed by the following formula:

$$\text{Frame Interval} = \frac{\text{Number of frames in the video}}{N} \quad (1)$$

where, N = Number of frames to extract.

TABLE I
INPUT IMAGE SHAPE AND OUTPUT FEATURE VECTOR SIZE OF DEEP CNN MODELS.

Model	Input Shape	Output Feature Vector Size
ResNet50	$224 \times 224 \times 3$	2048
VGG16	$224 \times 224 \times 3$	4096
InceptionV3	$299 \times 299 \times 3$	2048
DenseNet-169	$224 \times 224 \times 3$	1664
NASNet Large	$331 \times 331 \times 3$	4032

Feature extraction is a dimensionality reduction technique in which high dimensional input is downsized to a more significant representation for processing. All the extracted frames of every video are passed through pre-trained deep CNN models, producing the feature vectors. Feature vectors of all N (i.e., $N = 28$) frames of every video are collected, which represents the whole video and used for training the video captioning model. We have employed five different pre-trained deep CNN models for experiments, namely ResNet50, VGG16, InceptionV3, DenseNet-169 and NASNet-Large. These models have been trained on the ImageNet dataset and are capable of producing great results for

various image-processing tasks. Table I shows the extensive list of models and their output vector dimension for a single video frame:

1. VGG16: Developed by visual graphics group, VGG16 is a convolutional neural network that is 16 layers deep. It contains thirteen convolutional layers with ReLu activation function in the hidden layers. ReLu activation is used because it is more computationally efficient and it results in the faster learning of the model and also decreases the vanishing gradient problem probabilities. In addition, the model includes five max pooling layers and three dense layers, totalling 21 levels. However, it retains just sixteen weight layers, namely the learnable parameters layer. The VGG16 model can handle input tensors of scale 224 and 244, each with 3 RGB channels. Nevertheless, VGG16 has many drawbacks, including sluggish training speed and a substantial number of 138 million trainable parameters, resulting in an exploding gradient approach.

2. ResNet50: ResNet50 is a robust Convolutional Neural Network (CNN) with over 50 layers that effectively addresses the issue of vanishing gradients. As CNN models become more complex, the phenomenon of vanishing gradients often arises, leading to a detrimental effect on network performance. The vanishing gradient problem arises when we back propagate the gradient to earlier layers, resulting in a greatly reduced gradient and significantly subpar performance. Each subsequent successful architecture decreases the error rate by augmenting a deep neural network with more layers. While this approach is effective with a smaller number of layers, the vanishing/exploding gradient problem becomes prevalent in deep learning as the number of layers increases. This results in the angle either reducing to zero or having an extremely big value. Hence, expanding the number of layers results in a corresponding rise in the error rate throughout both the training and testing phases. ResNet50 incorporates a "skip connection" functionality, allowing for the training of several deep layers without encountering problems with vanishing gradient values. The skip connection interfaces the activations of one layer with those of other layers by bypassing some intermediate layers. Thus, a block is formed that persists. The ResNet50 is constructed by aggregating these residual blocks.

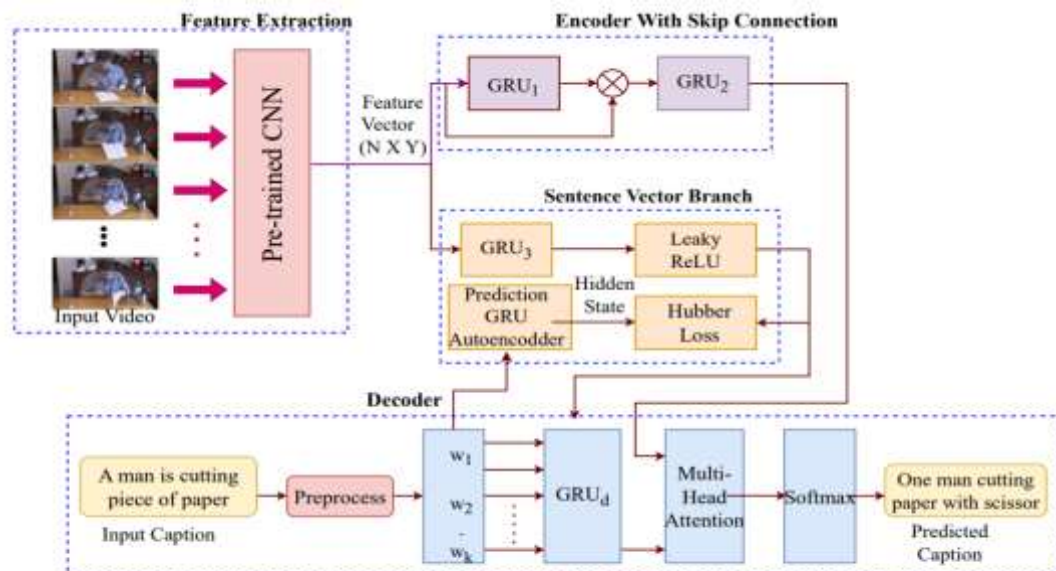


Fig. 3. The architecture of the proposed GRU-based model.

3. InceptionV3: It has a deeper network compared to the Inception V1 and V2 models, it has 48 layers consists of convolution layers, max pooling, ReLu and softmax activation

functions used, batch normalization and dense layers but its speed is not compromised. The model has been giving higher efficiency as well. It is computationally less expensive because of factorization into convolution layers and factorization into asymmetric convolution and factorization of 7*7 convolutions into asymmetric convolutions and this all is done to reduce the no. of parameters which will decrease the computational cost of the model and speed will not be compromised.

4. DenseNet-169: There is some fundamental difference between ResNet50 and DenseNet-169. ResNet50 is using an additive method which means it will give one previous output as input to the future layer and DenseNet-169 instead of using additive method takes all previous output as an input for a future layer. If we have many layers there will be feature map explosion so, to avoid this created a dense block to solve this issue, and each dense block has a predetermined number of layers inside. In order to reduce the size of the feature maps, the output from that particular dense block is sent to a layer known as a transition layer, which performs Max pooling followed by one-by-one convolution. So the progress layer considers Max pooling, which normally prompts a decrease in the size of your element maps.

5. NASNet-Large: Network architecture engineering is streamlined by Neural Architecture Search (NAS). An algorithm that looks for the best way to perform a given task is known as this one. Baker et al.'s ground breaking work [30] has seen a lot of interest in the field of neural architecture search (NAS), which has led to a lot of interesting ideas for NAS methods that are better, faster, and cheaper. Business administrations, for example, Google's AutoML and open-source libraries, for example, Auto-Keras make NAS available to the more extensive AI climate. It is a convolutional neural network that is trained on more than a million images from the ImageNet database. It is neural search architecture network.

C. Encoder with Skip Connections

The first phase was extracting the videos' spatial information from the video frames as feature vectors. Information about the action on a single frame of the video is contained in the feature vector of that frame. The GRU-based encoder network is used to extract the temporal information from the set of feature vectors. Two GRU layers and a skip connection were used in the encoder's construction. The decoder receives the encoder output and, given the current and encoder output, produces the subsequent word.

D. Sentence Vector Branch

The sentence vector branch helps the decoder create more accurate captions by giving it contextual information. The GRU-Layer receives extracted feature vectors and creates a M-dimensional vector, where M varies depending on the embedding method. For BERT and ELMO embeddings, it produces output vectors with dimensions of 768 and 1024, respectively. The sentence vector is created by passing this output vector through the Leaky ReLU activation and normalizing layer. The output of a hidden GRU-based auto-encoder that was trained on the caption throughout the training process is compared to this sentence vector.

E. Multi-Head Attention

Several heads are used in multi-head attention to gather data from several representational areas [31]. The sentence vector includes contextual information, while the encoder's output includes spatio-temporal information about the video. However, in order to construct the next word of the caption, the decoder uses scalar dot product attention [32] to choose only a subset of features.

IV. RESULTS AND ANALYSIS

Solving video captioning problem involves extensive use of deep learning algorithms. For understanding the problem closely, we have been studying the literature extensively to think about an appropriate approach to start with. There are two important aspects of the problem: (1) spatial information from the image frames and (2) temporal information. Our approach is to extract spatial information in the form of feature vectors using deep CNN models and train the LSTM model with captions for understanding temporal information. As the initial part of the solution, we have worked on preparing the dataset and BERT word embeddings for the captions.

A. Preparing Dataset for Training

We have divided the dataset into Train : Val : Test split in the ratio of 75 : 5 : 20 (all in percentage) which provides us with 1477 training, 98 validation and 395 test video clips. For training the models we need to first preprocess the dataset and in the initial step, we need to extract the image frames from the video clips and prepare frame-caption pair for training. We have analyzed the dataset to find out how many image frames we need to extract from the video clips on the basis of the number of captions available for each video. we have on average of 41 captions associated with a single video clip and a minimum of 18 captions are there for one clip. To keep it unsophisticated, we have extracted 28 image frames from each video clip for preparing the dataset. These N(28) frames are given as input to a pre-trained deep CNN model which will produce a Y - dimensional feature vector. In this manner, for every video clip, we get the N × Y dimensional feature vector.

For extracting the video frames we have used the OpenCV library. Since the duration of the video clips varies we need to find out the appropriate time interval for extracting the frames so that the entire video can be represented using them. The N × Y feature vector collects all the features from these N frames representing the information of the activity in the video.

B. Elmo Word Embedding for Captions

There are numerous techniques to convert the data into a vector format, such as assigning a distinct number to each word. However, this method has a problem in that the numbers produced are random. They miss the connections between the words. One hot encoding, which assigns a single word a value of 1 in the matrix and a value of 0 otherwise, is the second method for creating a vector for our data and making the computer comprehend what is going on. The drawbacks of this kind of word embedding are that they are computationally inefficient and fail to capture the relationship with other words. In addition to representing (1) complex facets of word use (including syntax and semantics) and (2) how these uses differ across linguistic contexts (i.e., to model polysemy), ELMO is a highly contextualized word representation. The size of the resulting word embedding vector in decimal places is 1024. A deep bidirectional language modeling (BLM) that had previously been trained on a large text corpus was used to create these word vectors as learnt functions. They significantly improve the state of the art for a number of intricate NLP problems, including textual entailment, sentiment analysis, and question answering, and are easy to include into existing models.

C. BERT Word Embedding for Captions the Video captioning problem is an exciting combination of computer vision and natural language processing. The machine is trained for understanding the context of the video and generating suitable caption which describes the activity in the video clip. A machine learning model is not capable of processing text data due to which a representation of text data is used which is called embedding. Embeddings are vectors that can be used as input feature in machine learning models that process text data. BERT (Bidirectional Encoder Representations from

Transformers) embedding is a very popular word embedding in NLP domain. It is very popular for producing state-of-the-art results in various NLP problems. The output vector generated by BERT embedding has 768 decimal values.

Over the years there are many word embedding techniques developed. These techniques can be broadly classified into two ways: (1) Static word embedding in which we assign some static values to the words and (2) Contextual embedding which aims to learn a vector representation for each word while preserving the contextual information. Generating the captions highly relies on the context of the previously generated words. BERT is one such contextual embedding technique that preserves context information of the words in the vocabulary.

We have tried to implement both static embedding and contextual embedding to analyze and compare the network performance. Captions for each video vary in length. We have first padded the captions by adding the start of string and end of string indicators and then tokenized them to prepare the vocabulary. The vocabulary is refined to exclude rarely used words by a word count threshold. The caption length is limited to 20 tokens and the captions exceeding maximum length is clipped, whereas the captions with smaller length are padded.

D. Evaluation Metrics

The intricacies of the fields of computer vision and natural language processing are both involved in creating captions from the videos. The generated captions are analyzed using metrics from the Natural Language Processing domain, such as BLEU, METEOR, ROUGE, and CIDEr. To compare our results with previous studies, we performed a quantitative study using the Microsoft COCO caption assessment tool. A predicted caption has been compared to each ground truth caption using these parameters. Since the metrics also account for the meaning and fluency of the caption, higher values indicate a high connection between the predicted caption and the ground truth. Because they more closely resemble human judgment, recent studies favor the aforementioned measurements above others.

E. BLEU (Bilingual Evaluation Understudy)

$$\text{BLEU} = \min \left(1, \exp \left(1 - \frac{\text{reference-length}}{\text{output-length}} \right) \right) \times \left(\prod_{i=1}^4 \text{Precision}_i \right)^{\frac{1}{4}} \quad (2)$$

The BLEU score is determined by contrasting a proposed translation with one or more reference translations. Although BLEU was designed for translation, it may assess text produced for various NLP tasks. For a perfect match BLEU score is 1.0, but a perfect mismatch is valued at 0.0. BLEU score computation is based on a comparison of n-grams in the proposed translation with the reference translation; n = 1 corresponds to a token whereas an n = 2 corresponds to a word pair. While comparing the sentences, the order of the words is not taken into consideration instead, the syntactic accuracy is questioned by the BLEU score. It just cares about the total number of words that exactly match the ground truth. Equation 2 defined the BLEU score as follows:

The precision is defined by the following equation:

$$\text{Precision} = \frac{C_w}{T_w} \quad (3)$$

Where, number of correct words in the generated sentence is C_w and T_w represents total number of words in the generated sentence.

F. METEOR (Metric for Evaluation for Translation with Explicit Ordering)

$$\text{Precision} = P = \frac{UG_{PR}}{UG_P} \quad (4)$$

METEOR was proposed to address the problems with BLEU. It substituted semantic matching for the exact lexical match that BLEU requires. WordNet, an English lexical database, is used by METEOR to account for several match levels, such as exact word matches, stemmed word matches, synonymy matching, and paraphrase matching. Unigram is used to determine METEOR score for predicted and reference sentences. With recall weighted higher than precision, METEOR is based on the harmonic mean of unigram precision and recall. Equation 4 defines precision P as the ratio of the count of unigrams that occur simultaneously in both predicted and reference sentences (UG_{PR}) to the count of unigrams in the predicted sentence (UG_P).

$$\text{Recall} = R = \frac{UG_{PR}}{UG_R} \quad (5)$$

Whereas, Recall R as described in equation 5 is the ratio of the count of unigrams that occur in both the predicted and reference sentences (UG_{PR}) to total count of unigrams in the reference sentence (UG_R) i.e.

$$\text{Mean harmonic score } F_{mean} = \frac{10PR}{R + 9P} \quad (6)$$

The harmonic mean combines precision and recall, with recall weighted nine times higher than precision. Mean harmonic score (F_{mean}), is calculated using Precision and Recall as defined in the following equation 6:

A penalty is imposed for longer matches and non-adjacent mappings between the predicted and reference sentences. Unigrams are grouped into as few chunks as possible, with a chunk being defined as a group of unigrams that are together in both the reference and the predicted sentence, in order to calculate the penalty. Penalty (P_n) can be calculated as in equation 7:

$$\text{Penalty } P_n = 0.5 \times \frac{c}{U_M} \quad (7)$$

Where, (c) represents the count of chunks, and the count of unigrams matched is represented by (UM). Finally, the following equation 8 can be used to calculate the METEOR score for a given alignment:

$$\text{METEOR} = F_{mean} \times (1 - P_n) \quad (8)$$

G. ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

$$\text{Recall } R_{LCS} = \frac{LCS(R_s, P_s)}{L_r} \quad (9)$$

To assess text summaries, the ROUGE measure was established in 2004. It calculates the recall score of the produced sentences that correlate to the reference phrases using n-grams. Both ROUGE and BLEU are determined by altering the number of n-grams. Rouge is based on recall, whereas Bleu is focused on precision. A variant that assesses captions for pictures and videos is called Rouge-L. It relied on the Recall and Precision value of the Longest common subsequence between the produced and each reference phrase. Recall and Precision are calculated in equations 9 and 10 respectively:

$$\text{Precision } P_{LCS} = \frac{LCS(R_s, P_s)}{L_p} \quad (10)$$

F-measure score based on LCS for predicted summary P_s of length L_p and reference summary R_s of length L_r can be computed as in equation 11:

$$\text{ROUGE}_{LCS(R_s, P_s)} = F_{LCS} = \frac{(1 + \beta^2) \times R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \quad (11)$$

Where, $LCS(R_s, P_s)$ illustrates the longest common subsequence of reference summary R_s and predicted summary P_s . β is the ratio of LCS precision and LCS recall.

H. CIDEr (Consensus-based Image Description Evaluation)

$$\text{CIDEr}_n(c_i, S_i) = \frac{1}{m} \sum_j \left(\frac{g^n(c_i) \times g^n(s_{ij})}{\|g^n(c_i)\| \times \|g^n(s_{ij})\|} \right) \quad (12)$$

CIDEr is a human consensus-based evaluation method for captions of images. The major objective of this evaluation metric is to compare a predicted text with one or more reference captions that have been commented by humans for the image. The system performs stemming and converts every word in the candidate and reference sentences to its underlying root form. The CIDEr algorithm interprets each sentence as a set of n-grams, with each gram consisting of one to four words. To evaluate the agreement between the predicted and reference sentences, we tally the n-grams present in both phrases. The Term Frequency Inverse Document Frequency (TF-IDF) method was employed to assign a lower weight to N-grams, which are especially common in the reference texts. The computation of CIDEr_n is performed using equation 12.

Where, $g^n(c_i)$ is a vector denoting all n-grams with length n and term $\|g^n(c_i)\|$ denotes magnitude. The same is true for $g^n(s_{ij})$.

A CIDEr variant recognised for evaluating picture and video description quality is called CIDEr-D. Stemming was removed from the original CIDEr, ensuring that the right word forms were employed. The application of a Gaussian penalty to the difference between predicted and reference captions stopped the repetition of phrases with higher confidence. Lastly, clipping was used to reduce the number of n-grams in the CIDEr numerator as a penalty for sentence length. It has been demonstrated that the CIDEr-D variant and the original CIDEr score have a high spearman's rank correlation.

I. Results of MSVD dataset

Numerous experiments have been performed with our model by employing various state-of-the-art feature extraction models and word embedding approaches. We have used five feature extraction models namely ResNet50, VGG16, InceptionV3, DenseNet-169 and NASNet-Large along with two-word embedding techniques ELMO and BERT. With diverse experimentations and parameter tuning, we have achieved great results with both our LSTM and GRU-based models. The Adam Optimizer is employed for training, with a batch size of 64 and a learning rate of $6e-4$. When the beam width is set to 3, beam search is used for testing. The results have been analyzed both quantitatively using the Microsoft COCO caption evaluation tool as well as qualitatively by analyzing the generated captions for the videos. For providing the quantitatively analysis and comparison of the results, the BLEU-4, METEOR, ROUGE-L, and CIDEr scores have been used.

J. Training Results of GRU-Based Model

Video features have been given as input to the Encoder and sentence vector branch. These branches have been implemented with GRU to reduce the training complexity and speed-up the caption generation. Testing of the model has been performed on unidentified videos that show how the model performs in the real world environment. Among all the models, NASNet Large outperforms which can be seen in Table and it also generates contextually accurate captions.

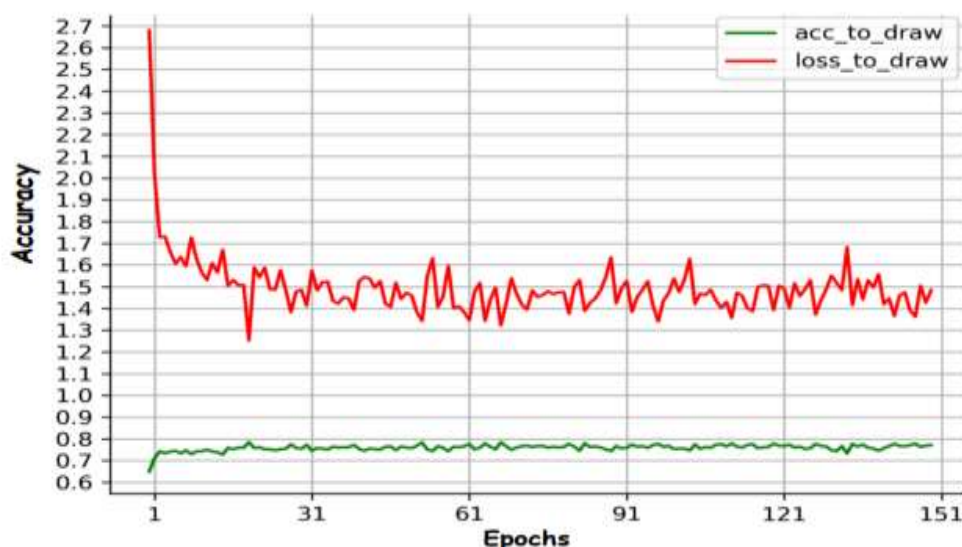


Fig. 4. Training Plot of GRU-based Model trained with the features extracted using VGG16 and BERT embedding.

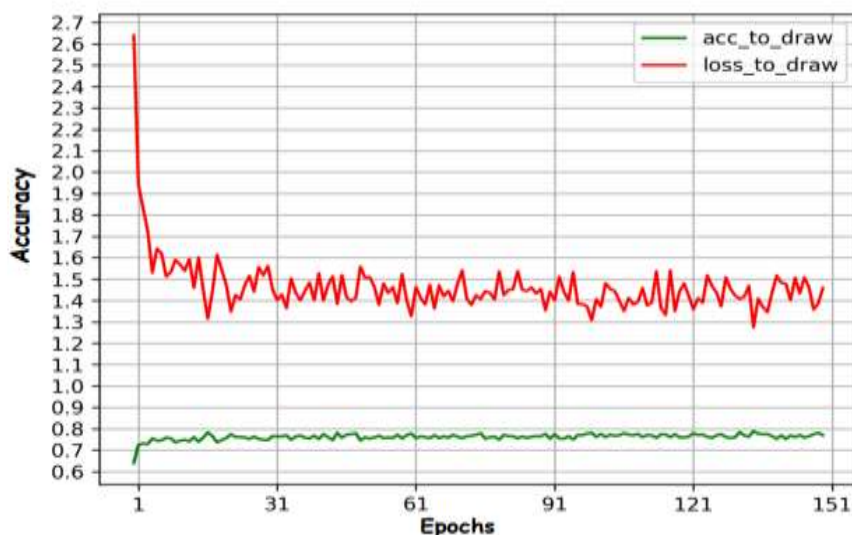


Fig. 5. Training Plot of GRU-based Model trained with the features extracted using VGG16 and ELMO embedding.



Fig. 6. For video: *AJJ-iQkbrNE 97 109.avi*, the generated caption is: *a man is slicing some vegitable and ground Truth captions are: a man slices a pineapple, a pineapple is being sliced, someone is slicing a pineapple with a knife, a man is cutting a pineapple etc.*

Fig. 7. For video: *SJZFNwfcSOw 3 18 5.avi*, the generated caption is: *a man plays a guitar and ground Truth captions are: a man is playing a guitar, a young man is singing and playing the guitar, a guy is playing guitar, a man is singing a song and playing a guitar etc.*



Fig. 8. For video: *x8UI09tk6zk_0_8_5.avi*, the generated caption is: **a man is playing basketball** and ground Truth captions are: a basketball player shoots a basket, a basketball player shoots a free throw, a young man is throwing a basketball into the hoop, boy playing basket ball etc.



Fig. 9. For video: *AJJ-iQkbrNE_97_109.avi*, the generated caption is: **a man is slicing some vegitable** and ground Truth captions are: a man slices a pineapple, a pineapple is being sliced, someone is slicing a pineapple with a knife, a man is cutting a pineapple etc.



Fig. 11. For video: *x8UI09tk6zk_0_8_5.avi*, the generated caption is: **a man is playing basketball** and ground Truth captions are: a basketball player shoots a basket, a basketball player shoots a free throw, a young man is throwing a basketball into the hoop, boy playing basket ball etc.

TABLE III COMPARISON OF TRAINING RESULTS OF GRU-BASED MODEL USING PRE-TRAINED CNN MODELS AND ELMO EMBEDDING ON THE MSVD DATASET.

Model	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
ResNet50	75.3	62.9	51.4	40.9	26.2	61.7	57.0
VGG16	74.3	58.7	52.9	39.7	29.1	63.9	62.8
InceptionV3	75.6	63.0	57.9	42.4	28.7	64.9	62.4
DenseNet-169	76.0	62.6	52.8	41.5	29.0	63.7	56.5
NASNet Large	78.0	67.7	55.3	45.7	32.4	67.5	70.3

TABLE IV COMPARISON OF TRAINING RESULTS OF GRU-BASED MODEL USING PRE-TRAINED CNN MODELS AND BERT EMBEDDING ON THE MSVD DATASET.

Model	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
ResNet50	76.6	63.2	52.7	41.9	28.7	63.6	58.9
VGG16	76.0	60.7	50.9	40.9	30.0	65.4	65.8
InceptionV3	77.6	64.5	54.2	43.8	29.9	66.3	64.6
DenseNet169	76.2	63.0	51.8	40.5	30.0	65.1	57.0
NASNet Large	79.1	66.4	55.8	46.2	33.1	69.7	70.7

Table V illustrates the experiment results for ResNet50 features involving different word embeddings. The ResNet50 comparison table data make it apparent that BERT consistently performs better than ELMO.

TABLE V COMPARISON OF TRAINING RESULTS USING RESNET50 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	76.6	63.2	52.7	41.9	28.7	63.6	58.9
ELMO	75.3	62.9	51.4	40.9	26.2	61.7	57.0

Table VI displays the experiment results for VGG16 features with various word embeddings. The VGG16 comparison table data make it quite evident that BERT consistently performs better than ELMO.

TABLE VI COMPARISON OF TRAINING RESULTS USING VGG16 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	76.0	60.7	50.9	40.9	30.0	65.4	65.8
ELMO	74.3	58.7	52.9	39.7	29.1	63.9	62.8

Table VII displays the experiment results for InceptionV3 features using various word embeddings. The comparison table results of InceptionV3 show that BERT outperforms ELMO in all scenarios.

TABLE VII COMPARISON OF TRAINING RESULTS USING INCEPTIONV3 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	77.6	64.5	54.2	43.8	29.9	66.3	64.6
ELMO	75.6	63.0	57.9	42.4	28.7	64.9	62.4

Table VIII displays the experiment results for DenseNet-169 features with various word embeddings. The DenseNet-169 comparison table findings show that BERT performs better than ELMO in all three scenarios, with the exception of the BLUE-4 score, when ELMO performs better than BERT.

TABLE VIII COMPARISON OF TRAINING RESULTS USING DENSENET-169 CNN MODELS AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	76.2	63.0	51.8	40.5	30.0	65.1	57.0
ELMO	76.0	62.6	52.8	41.5	29.0	63.7	56.5

Table IX displays the experiment results for NASNet-Large features with various word embeddings. The NASNet-Large comparison table data make it very evident that BERT consistently performs better than ELMO.

TABLE IX COMPARISON OF TRAINING RESULTS USING NASNET-LARGE CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDE _r
BERT	79.1	66.4	55.8	46.2	33.1	69.7	70.7
ELMO	78.0	67.7	55.3	45.7	32.4	67.5	70.3

K. Training Results of LSTM-Based Model

Implementation of the encoder and the sentence vector branch using LSTM is a complex architecture with a large amount of parameters to train. Our model has been trained for 150 epochs with each video feature extraction approach using the Adam optimizer. Examination of the results reveals that DenseNet-169 and NASNet Large surpass all the other models and produces realistic captions.

TABLE X COMPARISON OF TRAINING RESULTS OF LSTM-BASED MODEL USING PRE-TRAINED CNN MODELS AND BERT EMBEDDING ON THE MSVD DATASET.

Model	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDE _r
ResNet50	78.8	64.4	54.2	43.1	32.1	68.1	69.8
VGG16	79.4	66.9	56.4	46.5	32.5	70.2	71.1
InceptionV3	79.9	68.1	58.1	48.1	32.8	69.8	62.1
DenseNet-169	82.2	71.6	63.3	54.3	34.2	71.8	71.8
NASNet Large	81.8	69.6	59.7	50.7	35.1	71.6	77.0

TABLE XI COMPARISON OF TRAINING RESULTS OF LSTM-BASED MODEL USING PRE-TRAINED CNN MODELS AND ELMO EMBEDDING ON THE MSVD DATASET.

Model	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDE _r
ResNet50	79.5	66.2	55.8	45.3	33.4	70.5	72.6
VGG16	79.3	65.2	53.6	45.0	31.7	68.5	72.9
InceptionV3	79.4	67.5	57.5	47.2	34.1	70.9	74.4
DenseNet-169	82.2	71.2	61.7	51.8	35.5	71.4	78.9
NASNet Large	81.4	69.4	59.7	50.3	35.9	73.3	80.7

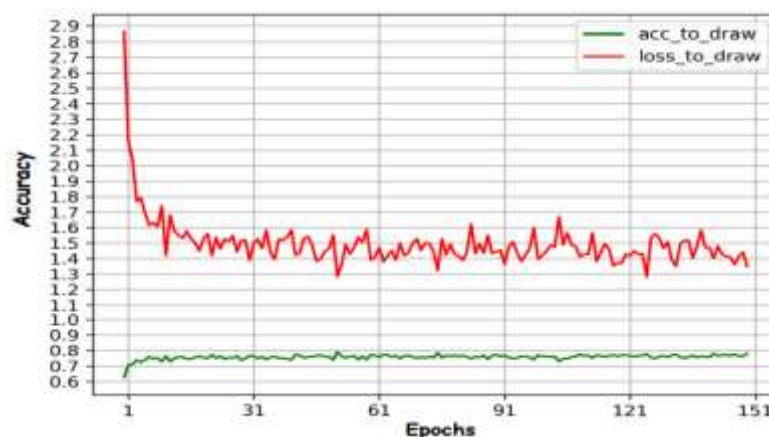


Fig. 12. Training Plot of LSTM-based Model trained with the features extracted using VGG16 and BERT embedding.

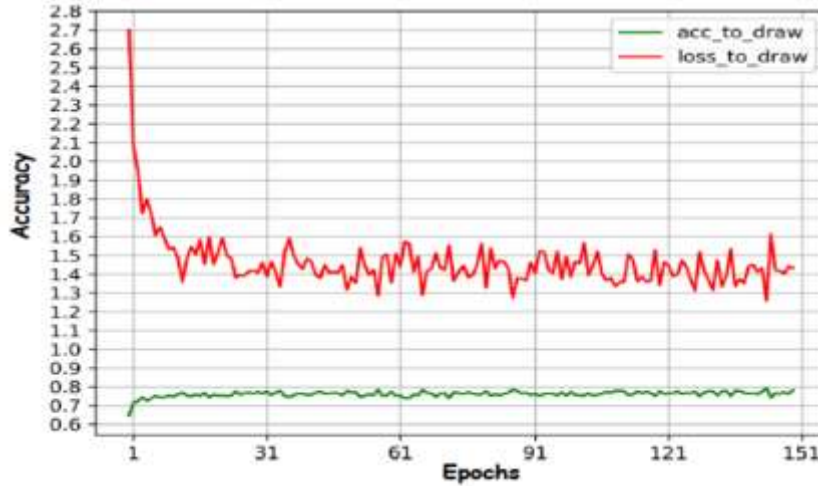


Fig. 13. Training Plot of LSTM-based Model trained with the features extracted using VGG16 and ELMO embedding.



Fig. 14. For video: AJJ-iQkbRNE 97 109.avi, the generated caption is: a person is slicing a fruit and ground Truth captions are: a man slices a pineapple, a pineapple is being sliced, someone is slicing a pineapple with a knife, a man is cutting a pineapple etc.



Fig. 15. For video: SJZFNwfcSOw 3 18 5.avi, the generated caption is: a boy is playing a guitar and ground Truth captions are: a man is playing a guitar, a young man is singing and playing the guitar, a guy is playing guitar, a man is singing a song and playing a guitar etc.



Fig. 16. For video: x8Ul09tk6zk 0 8 5.avi, the generated caption is: a man is dribbling a basketball and ground Truth captions are: a basketball player shoots a basket, a basketball player shoots a free throw, a young man is throwing a basketball into the hoop, boy playing basket ball etc.

Table XII displays the experiment results for ResNet50 features with various word embeddings.

TABLE XII COMPARISON OF TRAINING RESULTS USING RESNET50 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	78.8	64.4	54.2	43.1	32.1	68.1	69.8
ELMO	79.5	66.2	55.8	45.3	33.4	70.5	72.6

As it is clear from the comparison Table results of ResNet50, ELMO outperforms BERT in all the cases.

Table XIII presents the study's results for VGG16 properties with various word embeddings.

TABLE XIII COMPARISON OF TRAINING RESULTS USING VGG16 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	79.4	66.9	56.4	46.5	32.5	70.2	71.1
ELMO	79.3	65.2	53.6	45.0	31.7	68.5	72.9

As it is clear from the comparison table results of VGG16, BERT outperforms ELMO in all the three cases except CIDEr where ELMO outperforms BERT.

Table XIV demonstrates the experiment outputs for InceptionV3 properties with various word embeddings. As it is clear from the comparison table results of InceptionV3, BERT outperforms ELMO in one of the cases that is Bleu except that ELMO outperforms BERT in all the cases.

TABLE XIV COMPARISON OF TRAINING RESULTS USING INCEPTIONV3 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	79.9	68.1	58.1	48.1	32.8	69.8	62.1
ELMO	79.4	67.5	57.5	47.2	34.1	70.9	74.4

Table XV exhibits the experiment results for DenseNet-169 features that include various word embeddings. It is evident that BERT outperforms ELMO in every instance, with the exception of ELMO outperforming BERT in one metric (BLUE).

TABLE XV COMPARISON OF TRAINING RESULTS USING DENSENET-169 CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	82.2	71.6	63.3	54.3	34.2	71.8	71.8
ELMO	82.2	71.2	61.7	51.8	35.5	71.4	78.9

Table XVI displays the experiment results for NASNet-Large features with various word embeddings. It is evident that in one instance, Bleu, BERT is outperforming ELMO, with the exception that ELMO is outperforming BERT.

TABLE XVI COMPARISON OF TRAINING RESULTS USING NASNET-LARGE CNN MODEL AND WORD EMBEDDINGS ON THE MSVD DATASET.

Embedding	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
BERT	81.8	69.6	59.7	50.7	35.1	71.6	77.0
ELMO	81.4	69.4	59.7	50.3	35.9	73.3	80.7

TABLE XVII PERFORMANCE COMPARISON OF GRU-BASED MODEL WITH EXISTING APPROACHES ON THE MSVD TEST DATASET.

Model	Bleu4	METEOR	ROUGE	CIDEr
S2VT [19]	–	29.8	–	–
h-RNN [33]	49.9	32.6	–	65.8
S2VT+RL [34]	45.6	32.9	69.0	80.6
BAE [35]	42.5	32.4	–	63.5
TA [33]	41.9	29.6	–	51.7
S-VC [36]	35.1	29.3	–	–
Proposed (NASNet Large + BERT)	46.2	33.1	69.7	70.7

TABLE XVIII PERFORMANCE COMPARISON OF LSTM-BASED MODEL WITH EXISTING APPROACHES ON THE MSVD TEST DATASET.

Model	Bleu4	METEOR	ROUGE	CIDEr
S2VT [19]	–	29.8	–	–
h-RNN [33]	49.9	32.6	–	65.8
S2VT+RL [34]	45.6	32.9	69.0	80.6
BAE [35]	42.5	32.4	–	63.5
TA [33]	41.9	29.6	–	51.7
S-VC [36]	35.1	29.3	–	–
Proposed (NASNetLarge + BERT)	50.7	35.1	71.6	77.0

L. Various Ablation studies

Our model has been employed in numerous investigations, which have integrated a variety of advanced neural network models and word-encoding techniques. Additional neural network models and two-word embedding techniques were implemented as part of the ablation study. Our GRU-based models have produced substantial results as a consequence of a wide range of experiments. The Adam Optimiser is utilised for training purposes, with a batch size of 64 and a learning rate of $6e-4$. The beam search method is employed to conduct the testing, with a beam width of 3. The results were analysed quantitatively using the Microsoft COCO caption evaluation tool and qualitatively by examining the generated captions for the videos.

For GoogleNet features, Table XIX shows the result comparison. It is observed that BERT outperformed concerning all the mentioned metrics.

In Table XX, for Faster-RCNN features, it is observed that BERT outperformed concerning all the metrics mentioned above.

In Table XXI, for I3D features, it is observed that BERT outperformed with respect to all the metrics mentioned above.

In Table XXII, for concatenated GoogleNet and C3D features, it is observed that BERT outperformed with respect to all the metrics mentioned above.

TABLE XIX COMPARISON OF WORD EMBEDDINGS ON GOOGLNET FEATURES ON MSVD DATASET

Embedding	BLEU1	BLEU2	BLEU3	BLUE4	METEOR	ROUGE	CIDER
BERT	69.3	51.9	40.7	29.2	25.0	61.0	28.7
Elmo	67.2	50.1	38.5	27.4	22.8	58.26	26.9
Glove	65.2	49.5	37.6	26.8	21.3	57.3	25.8

TABLE XX COMPARISON OF WORD EMBEDDINGS ON FASTER-RCNN FEATURES ON MSVD DATASET

Embedding	BLEU1	BLEU2	BLEU3	BLUE4	METEOR	ROUGE	CIDER
BERT	64.4	48.5	38.7	29.2	23.2	57.4	28.4
Elmo	62.2	47.1	37.2	28.6	22.5	56.7	27.7
Glove	61.9	46.5	36.9	28.1	22.1	56.2	27.2

TABLE XXI COMPARISON OF WORD EMBEDDINGS ON I3D FEATURES ON MSVD DATASET

Embedding	BLEU1	BLEU2	BLEU3	BLUE4	METEOR	ROUGE	CIDER
BERT	70.8	55.4	44.7	34.3	26.9	62.6	34.4
Elmo	68.5	53.2	44.2	34.1	25.8	61.7	32.5
Glove	66.8	52.1	42.9	32.6	23.7	60.9	30.8

TABLE XXII COMPARISON OF WORD EMBEDDINGS ON GOOGLNET + C3D FEATURES ON MSVD DATASET.

Embedding	BLEU1	BLEU2	BLEU3	BLUE4	METEOR	ROUGE	CIDER
BERT	74.9	60.5	50.3	40.1	29.7	65.5	51.6
Elmo	72.8	59.2	48.7	38.3	28.8	64.9	50.8
Glove	70.1	58.5	47.3	36.8	26.4	62.7	48.9

M. Ablation studies on MSR-VTT Dataset

MSRVTT dataset [37] is an extensive dataset utilized for video captioning. The MSRVTT (Microsoft Research Video to Text) dataset consists of twenty significant categories and a diverse range of video materials. It serves as an open domain, extensive benchmark that establishes a connection between vision and language understanding. The compilation consists of 10,000 chosen clips extracted from 7180 videos. Content such as music, individuals, games, sports, news, education, automobiles, beauty, and advertising fall under the classification of open-domain video. The mean duration of each clip ranges from 10 to 30 seconds, resulting in a cumulative total of 41.2 hours. To ensure that each clip included sufficient semantics, 1327 AMT staff were employed to annotate each video with twenty real-world phrases. More than 200,000 clip-sentence combinations, amounting to 1.8 million words and 29316 unique words, are accessible.

TABLE XXIII COMPARISON OF TRAINING RESULTS OF LSTM-BASED MODEL USING PRE-TRAINED CNN MODELS AND BERT EMBEDDING ON THE MSRVTT DATASET.

Model	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDE _r
ResNet50	56.9	42.8	28.3	17.4	15.1	40.5	15.1
VGG16	58.5	43.0	28.7	18.1	16.1	40.9	21.7
InceptionV3	64.5	47.0	31.4	19.9	16.8	43.5	14.0
DenseNet169	53.5	37.1	22.9	13.1	13.7	36.5	11.1
NASNet Large	42.6	30.5	20.0	12.1	11.7	35.7	16.2

Table XXIII presents the ablation results on a larger dataset, wherein the InceptionV3 model surpasses all other models owing to the use of inception modules, which provide the model with a more refined degree of detail.

TABLE XXIV COMPARISON OF TRAINING RESULTS OF LSTM-BASED MODEL USING PRE-TRAINED CNN MODELS AND ELMO EMBEDDING ON THE MSRVT DATASET.

Model	Bleu ₁	Bleu ₂	Bleu ₃	Bleu ₄	METEOR	ROUGE	CIDEr
ResNet50	60.1	44.3	30.0	19.4	16.4	41.9	15.8
VGG16	61.0	45.6	31.7	20.8	17.5	45.2	23.9
InceptionV3	65.5	47.1	32.9	22.4	18.8	47.4	17.1
DenseNet169	56.7	39.5	25.4	14.9	15.1	38.6	14.1
NASNet Large	55.4	41.7	29.3	19.2	16.2	42.2	21.7

Table XXIV presents the ablation results on a ELMO emdeddings, wherein the InceptionV3 model again surpasses all other models.

V. CONCLUSION AND FUTURE WORK

Multiple tests have been conducted using our model, utilizing different cutting-edge feature extraction methods and word embedding techniques. Our study employed five feature extraction models in conjunction with two-word embedding strategies. Through a range of experiments, we have obtained excellent outcomes with both our models based on Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) models. In this study, we conducted feature extraction on the MSVD dataset using five distinct models. We incorporated sentence vector branch and skip connection into the architecture to enhance accuracy. Among the five models under consideration, NASNet-Large in GRU outperformed the other four models in the project. Additionally, when comparing word embedding in the MSVD dataset, BERT outperformed ELMO in most cases. After conducting identical experiments on the LSTM model using the MSVD dataset, NASNet-Large demonstrated superior performance compared to all four feature extraction models. When comparing word embedding, ELMO and BERT yielded different results. In most cases, ELMO outperformed BERT in three metrics: METEOR, CIDEr, and ROUGE. However, in Bleu, BERT outperformed ELMO word embedding.

Various word embedding models, such as GLOVE, can be employed to compare the outcomes of different word embeddings. Additionally, other feature extraction methods can be utilized to observe the disparity in accuracies generated by the various assessment metrics. This study aims to evaluate and contrast the results obtained from several 3D feature extraction techniques, such as Temporal Segment Networks (TSN), Convolutional 3D (C3D), and other similar approaches. In addition to YouCook II and MPII-MD, the model will be evaluated on other video datasets. In addition to 2D CNNs, 3-D CNNs enhance the video-to-text conversion methodology.

ACKNOWLEDGMENT

The authors express their gratitude to Dr. Dinesh Naik and Ms. Mehvish Marwaha for their essential assistance in the writing and editing of this paper. We also value their support and patience throughout the procedure.

REFERENCES

- [1] J. Barnard, "Online video viewing to reach 100 minutes a day in 2021 - Zenith — zenithmedia.com," <https://www.zenithmedia.com/online-video-viewing-to-reach-100-minutes-a-day-in-2021/>, 2021, [Accessed 06-Jun-2023].
- [2] D. Naik and C. Jaidhar, "A novel multi-layer attention framework for visual description prediction using bidirectional lstm," *journal of big data*, vol. 9, 11 2022.
- [3] M. Mohsin, "10 YouTube Statistics That You Need to Know in 2022 — oberlo.com," <https://www.oberlo.com/blog/youtube-statistics>, 2022.

- [4] N. Patwari and D. Naik, "En-de-cap: An encoder decoder model for image captioning," in 2021 5th international conference on computing methodologies and communication (ICCMC). IEEE, 2021, pp. 1192–1196.
- [5] D. Naik and C. Jaidhar, "A novel multi-layer attention framework for visual description prediction using bidirectional lstm," *journal of big data*, vol. 9, no. 1, p. 104, 2022.
- [6] R. M. R. Guddeti, "Human action recognition using multi-stream attention-based deep networks with heterogeneous data from overlapping sub-actions," *neural computing and applications*, vol. 36, no. 18, pp. 10 681–10 697, 2024.
- [7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.
- [8] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: <http://arxiv.org/abs/1802.05365>.
- [9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in the 3rd International Conference on Learning Representations (ICLR2015), 2015. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.
- [12] H. Liu, F. Qu, Y. Yang, W. Li, and Z. Hao, "Soybean variety identification based on improved resnet18 hyperspectral image," in *journal of physics: conference series*, vol. 2284, no. 1. IOP Publishing, 2022, p. 012017.
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [14] H. Mule and D. Naik, "Video to text generation using sentence vector and skip connections," in *international conference on machine learning and big data analytics*. Springer, 2022, pp. 515–527.
- [15] M. Rashmi and R. M. R. Guddeti, "Multi-stream multi-attention deep neural network for context-aware human action recognition," in 2022 IEEE Region 10 Symposium (TENSYP). IEEE, 2022, pp. 1–6.
- [16] N. Yadav and D. Naik, "Generating short video description using deep-lstm and attention mechanism," in 2021 6th international conference for convergence in technology (I2CT). IEEE, 2021, pp. 1–6.
- [17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *advances in neural information processing systems*, vol. 27, 2014.
- [18] R. M. and R. M. R. Guddeti, "Human action recognition using multi-stream attention-based deep networks with heterogeneous data from overlapping sub-actions," *neural computing and applications*, vol. 36, no. 18, pp. 10 681–10 697, June 2024. [Online]. Available: <https://doi.org/10.1007/s00521-024-09630-0>.
- [19] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence - video to text," in *proceedings of the IEEE international conference on computer vision (ICCV)*, December 2015.
- [20] M. M. Rahman, T. Abedin, K. S. S. Prottoy, A. Moshruha, and F. H. Siddiqui, "Video captioning with stacked attention and semantic hard pull," *peerj computer science*, vol. 7, p. e664, 2021.

- [21] D. Naik and C. Jaidhar, "Semantic context driven language descriptions of videos using deep neural network," *journal of big data*, vol. 9, no. 1, p. 17, 2022.
- [22] D. Naik and J. CD, "Video captioning using sentence vector-enabled convolutional framework with short-connected lstm," *multimedia tools and applications*, vol. 83, no. 4, pp. 11 187–11 213, 2024.
- [23] P. Shailendra, M. Rashmi, S. Ramu, and R. M. R. Guddeti, "Fake news detection in hindi using embedding techniques," in *2022 IEEE Region 10 Symposium (TENSYP)*. IEEE, 2022, pp. 1–5.
- [24] H. Maru, T. Chandana, and D. Naik, "Comparitive study of gru and lstm cells based video captioning models," in *2021 12th international conference on computing communication and networking technologies (ICCCNT)*. IEEE, 2021, pp. 1–5.
- [25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Jul. 2002, pp. 311–318.
- [26] S. Banerjee and A. Lavie, "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*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72.
- [27] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *text summarization branches out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: <https://www.aclweb.org/anthology/W04-1013>
- [28] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," *CoRR*, vol. abs/1411.5726, 2014.
- [29] D. Chen and W. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 190–200.
- [30] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [31] B. Sun, H. Dai, J. Sun, and X. Wei, "Research on intrusion detection models with multi-head attention mechanism based on graph neural network." *engineering letters*, vol. 33, no. 5, 2025.
- [32] Z. Mahmoud, N. Kryvinska, M. Abdalsalm, A. Solyman, A. Alfatemi, and A. Musyafa, "Toward utilizing bidirectional multi-head attention technique for automatic correction of grammatical errors." *IAENG international journal of computer science*, vol. 51, no. 11, pp. 1886–1897, 2024.
- [33] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *proceedings of the IEEE international conference on computer vision*, 2015, pp. 4507–4515.
- [34] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7008–7024.
- [35] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, jul 2017. [Online]. Available: <https://doi.org/10.1109%2Fcvpr.2017.339>
- [36] G. Li, S. Ma, and Y. Han, "Summarization-based video caption via deep neural networks," in *proceedings of the 23rd ACM international conference on multimedia*,

ser. MM '15. New York, NY, USA: association for computing machinery, 2015, p. 1191–1194. [Online]. Available: <https://doi.org/10.1145/2733373.2806314> [37] J. Xu, T. Mei, T. Yao, and Y. Rui, “Msr-vtt: A large video description dataset for bridging video and language,” in proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5288–5296.