S.Sanjay,, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Enhanced Video Captioning Using A Hybrid Vision-Swin Transformer Technique with Semantic Feature Augmentation and Improved Optimization by Eurasian Oystercatcher Algorithm

S.Sanjay, Dr. R Muthuram Associate Professor
Government College of Technology, Coimbatore.

Abstract- As an important step of multimedia processing, video captioning requires natural language generation for video content, integrating state-of-art approaches in computer vision and NLP to describe unmanaged visual information with useful text. It's complex task, as leveraging the temporal progression and the structured connections between objects, actions, and events in video is quite challenging. Into our paper, we suggest novel hybrid transformer model, that effectively integrates ViT and Swin Transformer based classifier for video captioning. The MSVD dataset is utilised for work. Caption preprocessing comes after which applies spelling correction, tokenization, part of speech tagging, and stop word removal. Applying TF-IDF, N-Grams, and semantic web-based feature extraction techniques for building a richer representation over textual data A hybrid transformer model was then utilised for extracting visual features and produce captions, followed by hyperparameter optimization utilising Eurasian Oystercatcher Optimiser (EOO). These captions are scored against ground truths utilising metrics like BLEU, METEOR and CIDEr.

Keywords - Video captioning, Vision Transformer, Swin Transformer, Semantic web, Eurasian Oystercatcher Optimiser.

I. INTRODUCTION

With the explosive increase of video information in diverse domains like social media, security, online education, entertainment and healthcare, there is increasing demand for automated tools to automatically interpret information encoded in video data [1]. And the increasing amount and complexity of video archives make it difficult to manage, index, and retrieve relevant information from them [2]. To remedy that, goal of video captioning has become a popular topic within the researcher community in AI community [3]. This problem lies at intersection of visual comprehension and natural language generation, drawing on techniques from both computer vision and NLP [4].

There are multiple layers of complexity involved in creating a robust video captioning system. It should be able to analyze visual content in single frames, understand how scenes are changing over time, and synthesize grammatical and semantically effective sentences [5]. Video captioning differs from still image captioning in that video captioning has temporal component as events and interactions unfold over sequence of frames making it a much more challenging task [6, 7]. Generation of such accurate and coherent descriptions is made more challenging because factors like motion artifacts,

changes in lighting, rapid transitions amongst scenes, and similarities in overlapping actions [8].

DL models have transformed way video captioning was approached in last few years. CNN-based models are commonly utilised for extracting visual features from frames, while recurrent networks—mainly variants like LSTM and & GRU—are effective in learning temporal dynamics and generating language output in a sequential manner [9]. Cross-modal alignment between image and text domains has led to major advances in visual-language models, & increasing popularity of attention mechanisms and Transformer architectures has solidified their strength by allowing the model to look at frames or blow-up objects relevant to generating a caption. Yet, despite clear progress with DLMs, DLMs typically act as black-box models, which impedes their interpretability. Additionally, their dependence on training data patterns may lead to shallow semantic comprehension of intricate scenes or unseen circumstances [10].

To address this semantic gap, Semantic Web technologies were progressively integrated within video captioning systems [11]. As Semantic Web introduces an array of structured formats for knowledge representation—via ontologies, linked data or knowledge graphs—it allows machines to comprehend and reason about relationships among concepts, thus drawing on the interconnectedness between ideas. These could be utilised to augment the knowledge acquired from data by introducing knowledge that is not available from either visual or linguistic inputs [12]. For example, an additional context in terms of concepts, which could add both accuracy and richness to generated captions could be added to any image captioning model via mapping of actions or objects detected from an image to those from standardization databases like WordNet, DBpedia or ConceptNet.. This blending of DL with semantic reasoning allows for deeper, more informed understanding of content within videos. Instead of generating superficial or generic descriptions, models enhanced with semantic information are better equipped to produce captions that are precise, contextually aware, and semantically meaningful [13].

Main contributions

- Utilization of the MSVD database to evaluate proposed video captioning model, offering diverse set of video-caption pairs for robust performance testing.
- Implementation of advanced preprocessing techniques, including caption tokenization, part-of-speech tagging, and stop word removal, to refine the textual data and improve caption quality.
- Adoption of semantic web-based feature extraction methods such as TF-IDF, N-Grams, and knowledge graph-based features to enhance the semantic depth of the captions.
- Development of hybrid transforming model with combination of ViT and Swin Transformer, along with hyperparameter tuning via the Eurasian Oystercatcher Optimiser (EOO), leading to superior classification and captioning performance.

Organization of paper

Remaining of this document is ordered as follows. In Section 2, we briefly review essential literatures; in Section 3 we projected the model. Section 4 includes results and summary of validation process. Section 5 concludes with summary and conclusion.

II. RELATED WORKS

Gad, G et al. [14] introduced IoT-integrated DL framework tailored for video captioning. The framework encompassed three primary stages: mining extensive open-domain video-to-text datasets to isolate domain-specific video-caption pairs, preprocessing these pairs to simplify language complexity for improved model efficiency, and implementing two deep learning architectures—one based upon transformers & other on LSTM—with hyperparameter optimization to enhance performance.

Kim, H. and Lee, S. [15] developed a multi-representation switching strategy comprising three key modules: entity extraction, motion analysis, and textual feature interpretation. The framework enabled these components to collaboratively extract significant cues from paired video and textual

descriptions, enhancing the overall representation quality.

Yan, L. et al. [16] addressed video captioning through a granularity-focused method termed the Global-Local Representation (GLR) framework. The model utilized comprehensive vision features from multiple video ranges to enrich linguistic outputs. A specially designed global-local encoder processed longer, shorter-range, and keyframe representations to construct semantically rich vocabularies. Additionally, a progressive training schedule was employed to optimize the learning process and boost descriptive accuracy.

Zaoad, M. S. et al. [17] centered efforts on generating Bengali-language video captions, identifying the most effective sequence-to-sequence architecture for this task. LSTM, BiLSTM, and GRU models were trained using frame features derived from CNN backbones such as VGG-19, Inceptionv3, and ResNet50v2. Attention mechanisms were integrated for the first time in Bengali captioning. A dedicated Bengali captioning dataset was curated by translating the MSVD dataset through deep learning-based translation tools and manual refinement.

Poddar, A. K. and Rani, R. [18] focused on generating Hindi captions using a multi-layer CNN-LSTM model. Various architectural configurations were tested by altering hidden layers and tuning hyperparameters to determine the most effective structure for generating descriptive Hindi annotations from image data.

Seo, P. H. et al. [19] introduced SwinBERT, transformer-based model engineered to perform end-to-end video captioning. Video patches were directly processed to produce descriptive outputs, bypassing the need for traditional 2D or 3D feature extraction pipelines. The architecture efficiently handled variable video lengths and benefited from denser temporal sampling. To reduce frame redundancy and enhance sequence modeling, a sparse attention mechanism was learned and refined, resulting in notable performance improvements in generating context-aware video descriptions.

Dinh, Q. M. et al. [20] introduced TrafficVLM, novel multi-modal dense video captioning model designed for vehicle ego camera views. TrafficVLM modelled traffic video events across multiple levels of spatial and temporal analysis, producing detailed and fine-grained descriptions of vehicles and pedestrians throughout different phases of observed events. Model implemented conditional component to control captions generation, & multi-task fine-tuning strategy exploited for facilitating model learning efficiency.

Alrebdi, N. et.al. [21] developed video captioning framework was proposed to study keyframes extraction method in an efficient way of achieving caption and supported two languages Arabic as well as English [21]. Keyframe extraction was done using time- and content-based methods for improving quality of captions while decreasing amount of storage space and increase speed of processing. For each language, we implemented sequence-to-sequence framework: LSTM networks were utilised in encoder & decoder. They were also assessed through other metrics like BLEU, METEOR, ROUGE-L and CIDEr and cosine similarity to measure model's effectiveness for task of video retrieval.

Research Gaps

Some of noticeable gaps in video captioning research include: A major challenge is the development of more advanced models that accurately model complex temporal dependencies and rich relational structures present across long video sequences. Existing methods have demonstrated effectiveness in processing single frames and limited-length video clips, but face challenges in sustaining coherence and contextual relevance across long-term timescales. Moreover, existing methods mainly target. English data, whereas multilingual video captioning, particularly for low-resource languages, is worth broadening. Another one is including multi-modal inputs, like audio and situational understanding, which would add into more complete captions. Moreover, rapid growth of video dataset sizes makes it necessary to develop more effective models which could address

escalating computational complexity and scale for top-notch video captioning performance. Finally, evaluation metrics utilised for video captioning, while commonly accepted, should ideally be further improved to closely correspond to human judgment and also consider more nuanced aspects of the caption quality like creativity and context sensitivity.

III. PROPOSED METHODOLOGY

Fig-1 represents working flow of projected video captioning model utilising Hybrid Transformer.

Figure 1: Block Diagram

Dataset Description

MSR-VTT, a large-scale benchmark dataset containing 10,000 clips which are transformed from 7180 videos. Clips were separated into 20 separate categories. Along with, AMT workers annotate every video clip with 20 single sentences. Training, validation, and test splits consist of 6513, 497, and 2990 clips following the official evaluation protocol given in [22].

Caption Preprocessing

Data preprocessing is a crucial step in computational linguistics that focuses on summarization. To get the most out of the captioning, preprocessing is necessary before running any experiments. The preprocessing stage includes the four actions listed below.

Spelling corrections

Errors in spelling must be taken into consideration in order to ensure that the analysis produces accurate results, as misspellings can occasionally alter the meaning of the sentence. To determine whether a word is misspelt and recommend the best correction, the spellchecker library was utilised.

Tokenization

The caption sentences are now divided into separate tokens (words) in this step. When tokenizing the review sentences into tokens, the primary indications blank, tab, and punctuation symbols like dot (.) and comma are typically used.

POS tagging

The practice of assigning a term to a speech segment is termed as part of
speech tagging [23]. The majority of people call it POS tagging. The components of speech
usually comprise nouns, conjunctions, and their
subcategories. This is accomplished by a piece of software called Parts of Speech Tagger, or POS Tagger

Stop Words Removal

Stop words are words that, in text mining, are not required for any division inside a phrase. Often, these phrases are ignored in an effort in improving accuracy of evaluation. Based upon language, domain, and other criteria, unique stop words can have many forms.

Semantic web-based Feature Extraction

TF-IDF with Semantic Enrichment

One well-known technique for determining a word's importance in a comment is the TF-IDF [24]. In this work, the relationship between a term's occurrences in a comment & total number of words in the comment is determined by term frequency (t), while the Inverse Document Frequency (IDF) helps to weigh down common terms like stop words (e.g., "is," "an," "the") that hold less semantic significance. In addition to the traditional TF-IDF method, we introduce semantic enrichment by utilizing ontology-based term mapping. For example, common words like "car" can be linked to a broader concept like "vehicle" through resources like WordNet or DBpedia, thereby improving the semantic relevance of the extracted features. By applying this enhanced TF-IDF calculation, we better capture conceptual meaning rather than just word frequency.

N-Gram with Semantic Context

The text features for supervised deep learning algorithms are formed by N-Grams. These are

sequences of n tokens from the text, with n representing different values like unigrams (1), bigrams (2), trigrams (3), etc. However, to align the feature extraction process with semantic understanding, we incorporate semantic N-grams by referencing conceptual entities and their relationships. E.g. instead of mapping the elements in the n-gram "Product is good" to literal words, we instead map them to concepts (e.g., "product" → "item" and "good" → "quality") and taking semantic relations (e.g., "product has quality") into account. This approach improves semantic relevance of relations among terms, thus maximizing learning model domain adaptability. [25].

Entity Recognition and Linking

Here, we recognize and map named entities — city, country, objects, actions, etc. — to some structured ontology or knowledge bases, like ConceptNet, DBpedia, YAGO. This step, known as entity linking, harmonizes terms like dog or car to their canonical representations. These semantic relationships offer richer, contextual understanding of the given words and assist in forming more accurate representation of the video content.

Semantic Similarity via Linked Data

Also, the semantic similarity between the terms is evaluated over Linked Open Data (LOD) and semantic knowledge graphs. Semantic pair: Words which are utilised in similar way such as joy and happy, which could be connected and mapped upon knowledge graphs. Use of semantic relationships helps model understand nuances in meaning more effectively. Connecting terms to knowledge networks facilitates extracting features whereas leveraging and incorporating world knowledge in representation.

Hybrid Transformer Classification

Vision Transformer

Visual transformers (ViTs) are state-of-the-art approach in computer vision, challenging classical convolution neural networks (CNNs) with different usage in image processing applications. Visual transformers have been quite successful in several of the standard computer vision benchmark. They are evolution of transformer architectures that were originally developed for NLP. Unlike traditional CNNs, ViTs rely purely upon transformer architecture [26].

Self-attention explains how Vision Transformers not only read specific patch but also think about importance of each patch. This allows the model to gain contextual information and long-range dependencies and makes model very powerful for image understanding. The self-attention mechanism generates a weight matrix for input pairs in sequence and calculates the attention scores for the interaction between each pair. When evaluating the significance of each patch in data aggregation process, this matrix is used. Their comprehension of global context is enhanced since these attention heads may focus upon several sections of the picture at once. An input sequence has a series of embeddings, wherein every embedding represents an input to the self-attention block. Tokens or places in the input sequence may be represented using embeddings. Throughout training, Vision Transformer learns to linearly convert embeddings in 3 vectors for every location, key, query, and value. Weights are determined by attention scores that show linkages between separate places in input sequence, as output of Transformer self-attention block is weighted sum of its input embeddings. A value, query, and key vector are generated from the input embedding by means of linear transformations. A ttention scores are calculated by multiplying query and key vectors by themselves. We get the weights

$$\text{Self Attention}\,(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V \qquad (1)$$

$$\text{Multi Head}\,(Q,K,V) = \text{Concat}\,(\text{head}_1,\ldots\,\text{head}_h)W^O \qquad (2)$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \qquad (3)$$

that indicate significance of each place by normalizing these scores using a softmax function. It automatically realizes insutive sequence of context by outputting outcomes considering all backbone

locations and sound attention from other backbone places.

ViTs multi-head self-attention mechanism greatly improves the model's capacity to detect various visual data structures and correlations. To create the final result for multi-head attention, outputs from these simultaneous attention heads are combined and then linearly processed. By using several attention heads to focus on distinct sections of the input sequence, the model is able to learn both coarse-grained and fine-grained characteristics, which ultimately improves its performance. 1. Multihead mechanism in Vision Transformers is the center of attention. It improves the model's representational capacity, which is crucial for Vision Transformers' performance on different computer vision tasks. Both single- and multi-head self-attention may be expressed mathematically as the equation above. W^Q,W^K,W^V attained weight matrices of same shape for query (Q), key (K) and value (V) transformations. second and respective first layer weight matrices, add feature W_a and W_b of last layer output and respective bias vectors B_a and B_b,consisting of ReLU activation and two linear activation functions. Mathematically represented as: X linearly transforms

$$FFN = \text{ReLU}\,(XW_a + B_a)W_b + B_b \qquad (4)$$

Output from multi-head self-attention block is passed through point-wise FFN .ReLU is universal non-linear function that makes model non-linear at each element. Thus, the model is able to independently identify complicated and non-linear patterns for every location. Conventional FFN enhances ViT model's representation of each location, allowing it to learn and detect complex characteristics from input picture sequences [27].

In order to train and test Vision Transformers (ViTs), an input picture is first divided in fixed-size, non-overlapping patches. Following linear embedding, a trainable linear transformation flattens each patch into a vector. E ncoder block of transformer is used by model to analyze and extract spatial and contextual info from picture. This information is injected using a series of overlapping patches.

The Swin Transformer

The Swin Transformer made it easier in capturing spatial hierarchies & local-global linkages in pictures with its hierarchical architecture and shifting panes. In order to create non-overlapping patches from an input picture, the Swin Transformer uses a specific patch splitting module, which is similar to ViT's patch-level hierarchical design [28]. A "token" is created for each of these patches by joining the RGB values of its individual pixels. We assign a value of C to these raw-valued features by means of a linear embedding layer. In a certain amount of Swin Transformer blocks, it undergoes changed attention calculations. Since this is "Stage 1" of linear embedding with these transformer blocks, the total token count remains constant. To create a hierarchy and decrease the token count, patch-merging layer is added to network as it deepens. Using the flattened features of each 2x2 neighboring patch group and a linear layer applied to the combined 4C-dimensional features, we combine all four patches into one. Stage 2 is the name given to the first block that involves combining patches and transforming features. Stages 3 and 4 are the outcome of two repetitions of this technique.

The Swin Transformer introduces a new method of self-attention that makes use of shifting windows. This method purposes to efficiently capture both local & global characteristics, in contrast to the usual MSA paradigm seen in conventional transformer blocks. A global self-attention mechanism computes the associations amongst token and every other token; this design is often employed in transformers for visual tasks. This global algorithm is not suitable for many vision applications that need a large amount of tokens for dense prediction or for showing high-resolution pictures because of its quadratic complexity with number of tokens.

Implementing self-attention in localized windows is the primary objective of the shifted window. Each window, consisting of non-overlapping patches of M×M size, is used to calculate self-attention. As a result, the computational complexity drops: The original MSA exhibited quadratic complexity, in contrast to window-based MSA, which demonstrates

linear difficulty with regard to the patch number. [29].

Swin Transformer uses a shifting window partitioning technique, which involves alternating in 2 configurations across consecutive blocks, to accurately mimic window connections. First module uses conventional window arrangement to calculate local self-attention from uniformly spaced windows, beginning at top-left pixel. Then, next Swin Transformer block uses a window layout that is moved by (M/2,M/2) pixels pertaining with previous layer. This strategic adjustment enhances model's capacity to accurately depict a range of spatial connections. Swin transformer blocks' self-attention may be expressed mathematically in following way: The relative location bias of the window is denoted by B.

$$\text{Self Attention}\,(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}} + B\right)V| \qquad (5)$$

This research utilizes the features of ViT-B and Swin-B basic models, while there are several variants of ViT & Swin transformer topologies.

3.5. EOO based Hyper parameter tuning

EOO is an optimisation algorithm inspired by nature which was used here for hyperparameter tuning. The algorithm's creative process will be explained in this section. Also, an explanation of the mathematical framework will be given.

Inspiration

By timing how long it took the bird to open mussels, how much energy it utilised, and how many calories it would gain from the hunt, Meire and Ervynck [30] calculated worth of gaining from various sizes of prey. When comparing with prey of smaller sizes, mussels measuring 50 mm or longer offer greater nutritional value for every minute spent opening their shells through stabbing, but it also takes more time and energy. Based on the model's aforementioned assumptions, the mussel eater would prioritise consuming larger prey. The bird does not, however, favour big mussels. The following explanation was given for this discrepancy between assumption and reality. The first presumption is that since some mussels have incredibly strong shells that are impossible to open, the median profitability from big prey is less than the optimum amount. There was a flaw in this step of the calculation of the profit made from the prey because the researchers neglected to account for mussels as bird managed for opening. Sometimes mussel catchers select large prey that, even with great effort, they are unable to open. The time spent handling large, though breakable, prey lowers these mussels' average benefit. This factor will lead to the emergence of a new predicting model, which states that the mussel catchers should concentrate on 50 mm-long prey rather than very large ones. However, prey that is between 30 and 45 mm in size is typically preferred by mussel catchers. As a result, they lose time focusing on the massive, unbreakable mussels, which makes it difficult for them to explain how the mussel catchers choose their food [31]. According to the alternative theory, the surface of big mussels is covered in a layer of barnacles. Because of this, it is nearly impossible for the shell to open, and mussel catchers do not like this kind of prey. Seeing the mussel catchers while they eat lends credence to this theory. Despite the fact that barnacle-covered mussels are high in calories, this bird avoids eating them. The mussels are potentially more covered in barnacles as they get bigger, which makes them more brittle and less desirable. If the mathematical model considers the time necessary for opening mussels, time wasted trying to open certain prey in vain, & sizes appropriate for predation, then mussel catchers should focus upon prey sizes between 30 and 45 mm. Scientists were able to confirm what many had suspected: mussel catchers had an almost ideal diet.

Mathematical model

Here we provide mathematical search model & appropriate mussel that EO selected. M ajor objective of EO is to remain in a state of equilibrium amongst their energy and the calories used by mussels. Mussels' calorie and energy density is proportional to their size. L ength of mussels affects both number of calories & time it takes to open them. Hence, EO waste must be treated with great vigor. Eqs. (6) and (7) depict EO's activities during the search procedure.

Y=T+E+L⋆r⋆(X_"best " -X_(i-1)) (6)
X_i=X_(i-1)⋆C (7)
where X_i is a candidate mussel's position, L denotes mussel's length and is a random number between 3 and 5 denoting range of ideal length of T is amount of time required to open current mussel, and its value depends on L in accordance with equation (31). The appropriate mussel's size, which was indicated in inspiration section, is the basis for the use of numbers (3 and 5) in equation (8). Since the value of the EO decreases with each iteration, E, or the energy of the EO at any given time, is derived from equation (9). To add additional unpredictability and find new locations in the search area, r is a random number between 0 and 1. Caloric value, or C in equation (30), is determined by the mussel's length and is derived from equation (10).

$$T = \left(\left(\frac{L-3}{5-3}\right) * 10\right) - 5 \quad (8)$$

$$E = \left(\frac{i-1}{n-1}\right) - 0.5, \text{ where } i > 1 \quad (9)$$

$$C = \left(\left(\frac{L-3}{5-3}\right) \star 2\right) + 0.6 \quad (10)$$

Equation (8) shows the value resulting from equation (10) in the range of 0.6 to 0.8, as well as the value between (5) and (-5). These values were determined via a process of trial and error. Notice that if the time is negative, then the bird may not be able to open the mussels in the allotted amount of time, and if it's positive, then the bird can crack the mussels in the allotted amount of time. The E-value, which fell linearly from 0.5 to -0.5, was calculated using Equation (8). The value of iteration, denoted by "i" in this equation, starts with number of iterations and ends with one. In the latter two cycles, E value remains constant. To open the candidate mussel, time and energy needed (T and E, respectively) might be negative numbers, provided that they are less than energy of the EO. The values of T in equation (6) and C in equation (7) are both derived from L, an unpredictable random variable. Always prioritizing exploration, this condition eliminates the local minimum issue and allows EO to access any location in the search space. In theory, the primary characteristics of EOO that aid in the resolution of optimisation issues are explained by the following points:

Algorithm 1: Pseudocode of EOO
Initialize the EO population X_i (i=1,2,……,n)
Calculate the fitness of each search agent
X_"best " = the best solution in search agent
While (i>0)
For each solution in search agent
L= random (3,5)
Calculate T,E and C based on equations 3,4 and 5
Update the position of solution based on equations 1 and 2
End for
Calculate the fitness of each search agent
X_"best " = the best solution in search agent

End while
Return X_{best}

IV. RESULTS AND DISCUSSIONS

The experiments were carried out on a system having NVIDIA RTX 3090 GPU (24 GB), Intel i9 processor, and 64 GB RAM. The software environment includes Ubuntu 20.04 LTS OS, Python 3.9, and PyTorch 2.0 support.

Table 1: BLEU Score Comparison for Different Video Captioning Models

Model / Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4
LSTM	0.61	0.45	0.33	0.24
Swin Transformer	0.66	0.52	0.48	0.39
Vision Transformer	0.72	0.54	0.54	0.53
Proposed Hybrid Transformer model	**0.84**	**0.72**	**0.74**	**0.82**

Table 1 and figure 2 presents a comparative analysis of BLEU scores across different video captioning models. The LSTM-based model achieved BLEU-1 to BLEU-4 scores of 0.61, 0.45, 0.33, and 0.24 respectively, indicating limited performance in capturing multi-level n-gram precision. The Swin Transformer showed improved results with scores of 0.66 (BLEU-1), 0.52 (BLEU-2), 0.48 (BLEU-3), and 0.39 (BLEU-4), demonstrating better contextual

understanding. The Vision Transformer further enhanced caption quality, achieving 0.72, 0.54, 0.54, and 0.53 for BLEU-1 through BLEU-4 respectively. The proposed Hybrid Transformer model, combining Vision Transformer and Swin Transformer, outperformed all baseline models with BLEU scores of 0.84, 0.72, 0.74, and 0.82, highlighting its superior ability to generate accurate and coherent video descriptions across various levels of linguistic granularity.

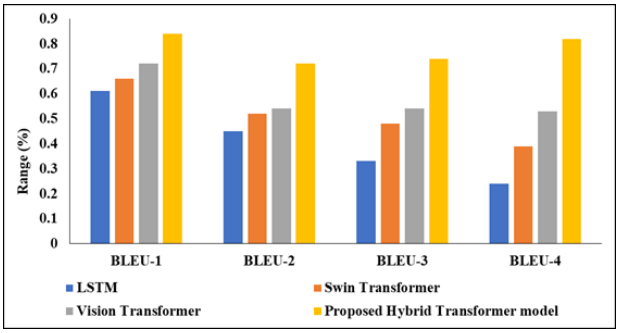

Figure 2: BLEU validation

Table 2: CIDEr and METEOR Score Comparison for Different Video Captioning Models

Model / Method	CIDEr Score	METEOR Score
LSTM	0.81	0.24
Swin Transformer	0.97	0.37
Vision Transformer	1.15	0.51
Proposed Hybrid Transformer model	**1.52**	**0.85**

Table 2 and figure 3 provides a comparison of CIDEr and METEOR scores for various video captioning models. The LSTM model recorded the lowest performance with a CIDEr score of 0.81 and a METEOR score of 0.24, indicating limited alignment with human-generated captions in both consensus and semantic quality.

The Swin Transformer showed notable improvement, achieving a CIDEr score of 0.97 and a METEOR score of 0.37, reflecting better descriptive relevance. The Vision Transformer achieved superior metrics, with a CIDEr of 1.15 and a METEOR of 0.51, indicating greater semantic coherence and textual consistency. Our Hybrid Transformer model achieved the highest CIDEr score, 1.52, & METEOR score of 0.85, outperforming all other methods, validating its potential to produce semantically-rich and contextually relevant video captions.

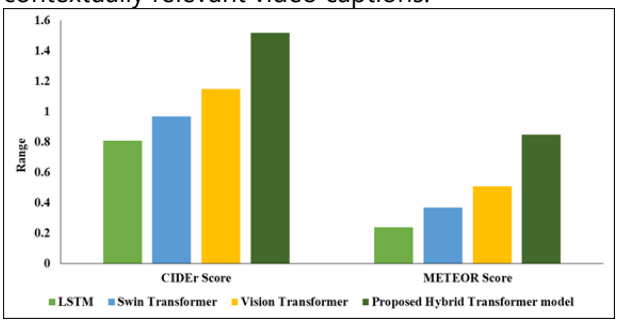

Figure 3: CIDEr and METEOR validation

Table 3: Performance analysis of classification models

Methods	Accuracy (%)	Precision (%)	Sensitivity (%)	F-measure (%)
LSTM	90.94	90.45	90.23	91.54
Bi-LSTM	92.45	91.49	91.64	92.23
Swin Transformer	93.74	92.33	93.27	93.42
Vision Transformer	94.94	94.64	94.33	94.71
Proposed Hybrid Transformer model	98.76	97.37	97.44	97.23

Table 3 and figure 4 presents the performance analysis of various classification models as per key evaluation metrics: Accuracy, Precision, Sensitivity, and F-measure. The LSTM model achieved an accuracy of 90.94%, with a precision of 90.45%, sensitivity of 90.23%, and an F-measure of 91.54%, indicating baseline effectiveness. The Bi-LSTM model slightly outperformed LSTM, reaching 92.45% accuracy, 91.49% precision, 91.64% sensitivity, and a 92.23% F-measure, benefiting from its bidirectional structure. The Swin Transformer further improved classification performance with 93.74% accuracy, 92.33% precision, 93.27% sensitivity, and a 93.42% F-measure. The Vision Transformer performed even better, achieving 94.94% accuracy, 94.64% precision, 94.33% sensitivity, and 94.71% F-measure. The proposed Hybrid Transformer model demonstrated superior performance across all metrics, attaining 98.76% accuracy, 97.37% precision, 97.44% sensitivity, and a 97.23% F-measure, clearly highlighting its robustness and efficiency in video content classification.

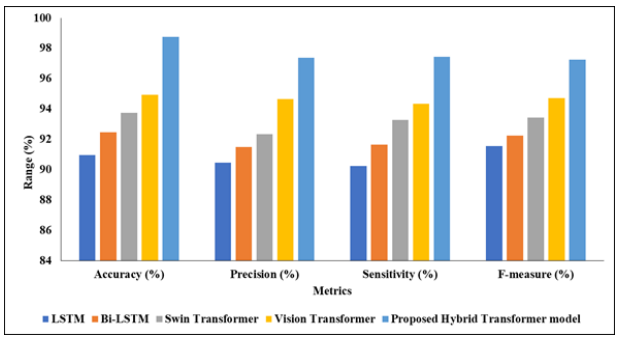

Figure 4: Comparative Validation Performance of Classification Models

V. CONCLUSION

Based on the above insights, novel hybrid transformer-based framework is designed for video captioning based on both ViT and Swin Transformer, in order to achieve better visual understanding and caption generation. Hybrid architecture exploited both global and local visual information that allows the model to learn the complex temporal dynamics and semantic correlations within a video. Model performance was evaluated using the MSVD dataset; we leveraged some text prep for the captions, enhanced feature extraction strategies (TF-IDF, N-Grams, Semantic Web and Ontologies, Knowledge Graphs) resulting in higher quality, semantically-meaningful captions. Eurasian Oystercatcher Optimiser (EOO) was utilized to perform hyperparameter tuning, which was key to fine-tuning model parameters for utmost performance. Standard evaluation metrics like BLEU, METEOR, CIDEr, ROUGE-L, and SPICE showed that the hybrid model proposed here outperforms state-of-the-art models including LSTM, Bi-LSTM, as well as single transformer-based architectures. This was corroborated by the BLEU-4 score of 0.82, METEOR score of 0.85, CIDEr score of 1.52, and a classification accuracy of 98.76%, which reflects robustness, linguistic correctness, and semantic relevance of the captions generated by the model. In conclusion, the proposed framework provides a unified approach that connects visual scene understanding to text generation, facilitating progress toward intelligent multimedia applications. This work could be extended by incorporating audio cues, multi-modal fusion techniques and multilingual captioning capabilities to address several practical concerns of video captioning systems.

REFERENCES

1. Abdar, M., Kollati, M., Kuraparthi, S., Pourpanah, F., McDuff, D., Ghavamzadeh, M., ... & Porikli, F. (2024). A review of deep learning for video captioning. IEEE Transactions on Pattern Analysis and Machine Intelligence.
2. Islam, S., Dash, A., Seum, A., Raj, A. H., Hossain, T., & Shah, F. M. (2021). Exploring video captioning techniques: A comprehensive survey on deep learning methods. SN Computer Science, 2(2), 1-28.
3. Islam, S., Dash, A., Seum, A., Raj, A. H., Hossain, T., & Shah, F. M. (2021). Exploring video captioning techniques: A comprehensive survey on deep learning methods. SN Computer Science, 2(2), 1-28.
4. Ghandi, T., Pourreza, H., & Mahyar, H. (2023). Deep learning approaches on image captioning: A review. ACM Computing Surveys, 56(3), 1-39.
5. Perez-Martin, J., Bustos, B., & Pérez, J. (2021). Improving video captioning with temporal composition of a visual-syntactic embedding. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3039-3049).
6. Yan, L., Han, C., Xu, Z., Liu, D., & Wang, Q. (2023, August). Prompt Learns Prompt: Exploring Knowledge-Aware Generative Prompt Collaboration For Video Captioning. In IJCAI (pp. 1622-1630).
7. Kang, S. H., & Han, J. H. (2023). Video captioning based on both egocentric and exocentric views of robot vision for human-robot interaction. International Journal of Social Robotics, 15(4), 631-641.
8. Mkhallati, H., Cioppa, A., Giancola, S., Ghanem, B., & Van Droogenbroeck, M. (2023). SoccerNet-caption: Dense video captioning for soccer broadcasts commentaries. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5074-5085).
9. Zheng, Y., Zhang, Y., Feng, R., Zhang, T., & Fan, W. (2021). Stacked multimodal attention

network for context-aware video captioning. IEEE transactions on circuits and systems for video technology, 32(1), 31-42.
10. Vaidya, J., Subramaniam, A., & Mittal, A. (2022). Co-segmentation aided two-stream architecture for video captioning. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 2774-2784).
11. Chen, J., Pan, Y., Li, Y., Yao, T., Chao, H., & Mei, T. (2023). Retrieval augmented convolutional encoder-decoder networks for video captioning. ACM Transactions on Multimedia Computing, Communications and Applications, 19(1s), 1-24.
12. Amirian, S., Rasheed, K., Taha, T. R., & Arabnia, H. R. (2021). Automatic generation of descriptive titles for video clips using deep learning. In Advances in Artificial Intelligence and Applied Cognitive Computing: Proceedings from ICAI'20 and ACC'20 (pp. 17-28). Springer International Publishing.
13. Perez-Martin, J., Bustos, B., & Pérez, J. (2021, January). Attentive visual semantic specialized network for video captioning. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 5767-5774). IEEE.
14. Gad, G., Gad, E., Cengiz, K., Fadlullah, Z., & Mokhtar, B. (2022). Deep learning-based context-aware video content analysis on IoT devices. Electronics, 11(11), 1785.
15. Kim, H., & Lee, S. (2021). A video captioning method based on multi-representation switching for sustainable computing. Sustainability, 13(4), 2250.
16. Yan, L., Ma, S., Wang, Q., Chen, Y., Zhang, X., Savakis, A., & Liu, D. (2022). Video captioning using global-local representation. IEEE Transactions on Circuits and Systems for Video Technology, 32(10), 6642-6656.
17. Zaoad, M. S., Mannan, M. R., Mandol, A. B., Rahman, M., Islam, M. A., & Rahman, M. M. (2023). An attention-based hybrid deep learning approach for Bengali video captioning. Journal of King Saud University-Computer and Information Sciences, 35(1), 257-269.
18. Poddar, A. K., & Rani, R. (2023). Hybrid architecture using CNN and LSTM for image captioning in Hindi language. Procedia Computer Science, 218, 686-696.
19. Seo, P. H., Nagrani, A., Arnab, A., & Schmid, C. (2022). End-to-end generative pretraining for multimodal video captioning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 17959-17968).
20. Dinh, Q. M., Ho, M. K., Dang, A. Q., & Tran, H. P. (2024). Trafficvlm: A controllable visual language model for traffic video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7134-7143).
21. Alrebdi, N., & Al-Shargabi, A. A. (2024). Bilingual video captioning model for enhanced video retrieval. Journal of Big Data, 11(1), 17.
22. Xu, J.; Mei, T.; Yao, T.; Rui, Y. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5288–5296.
23. Suzanti, I. O., Husni, H., Rachman, F. H., Mulaab, M., & Mahendra, P. C. S. (2025, January). Enhanced Sorensen dice coefficient using POS tagging for similarity detection system. In AIP Conference Proceedings (Vol. 3250, No. 1). AIP Publishing.
24. Delibaş, E. (2025). Efficient TF-IDF method for alignment-free DNA sequence similarity analysis. Journal of Molecular Graphics and Modelling, 137, 109011.
25. Kusumo, F. A., Saputro, D. R. S., & Widyaningsih, P. (2025). SENTIMENT ANALYSIS OF REVIEWS ON X APPS ON GOOGLE PLAY STORE USING SUPPORT VECTOR MACHINE AND N-GRAM FEATURE SELECTION. BAREKENG: Jurnal Ilmu Matematika dan Terapan, 19(2), 1037-1046.
26. Ulukaya, S., & Deari, S. (2025). A robust vision transformer-based approach for classification of labeled rices in the wild. Computers and Electronics in Agriculture, 231, 109950.
27. Meng, C., Lin, W., Liu, B., Zhang, H., Gan, Z., & Ouyang, C. (2025). RTS-ViT: Real-Time Share Vision Transformer for Image Classification. IEEE Journal of Biomedical and Health Informatics.
28. Karthik, R., Ajay, A., Jhalani, A., Ballari, K., & K, S. (2025). An explainable deep learning model for diabetic foot ulcer classification using swin

transformer and efficient multi-scale attention-driven network. Scientific Reports, 15(1), 4057.
29. Zhang, J., Zhou, H., Liu, K., & Xu, Y. (2025). ED-Swin Transformer: A Cassava Disease Classification Model Integrated with UAV Images. Sensors, 25(8), 2432.
30. Salim, A., Jummar, W. K., Jasim, F. M., & Yousif, M. (2022). Eurasian oystercatcher optimiser: New meta-heuristic algorithm. Journal of Intelligent Systems, 31(1), 332-344.
31. Kondaiah, M., & Padmaja, M. (2024). Enhancement of Eurasian oystercatcher optimiser for spectral efficiency maximisation in massive MIMO systems through optimal power allocation. International Journal of Ad Hoc and Ubiquitous Computing, 46(1), 1-13.