IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Improving Tamil Image Captioning: Comparative Study Of Techniques And Models

[1]Suriyaprakash G, [2]Jaijeevaalshree N, [3]Dr. Thamizharasan S
[1]Student, [2]Student, [3]Asst. Professor
[1]School of Computer Science and Technology,
[1]Vellore Institute of Technology, Vellore, India

Abstract: The problem of producing precise and contextually relevant textual descriptions from visual inputs has attracted a lot of attention in the emerging subject of image captioning. Using the Flickr8k dataset as a benchmark, this proposed work compares three different approaches to improve Tamil image captioning capabilities. It also clarifies the relative benefits of each approach in tackling the complex problems associated with non-English image captioning. Using Long Short-Term Memory (LSTM) networks and the Convolutional Neural Network (CNN) VGG16, the first methodology establishes a baseline by taking advantage of LSTM's sequential data processing strengths and VGG16's skill in extracting significant visual characteristics. The second approach improves caption accuracy by using an attention-based encoder-decoder architecture that uses LSTM layers enhanced with an attention mechanism to carefully choose relevant image segments at the time of caption synthesis. Metrics like BLEU scores and attention plot are used to examine this model's efficacy and provides the accuracy of its contextual captioning. The last method pushes the boundaries of innovation even further by combining Vision Transformers (ViTs) with Transformer Networks for image processing and sequence production, respectively. By utilising the complementary abilities of Transformer Networks and ViTs, this sophisticated model seeks to significantly enhance caption quality by deciphering the intricate relationship between visual content and verbal descriptions. The thorough comparative research clarifies the trade-offs related to striking a balance between caption authenticity, computational demands and model complexity. It also highlights how important linguistic issues are when discussing Tamil, a language distinguished by its complex morphological and syntactic structure. This work adds to the growing body of knowledge on multilingual image captioning by providing crucial insights and direction for further research aimed at creating more advanced, linguistically sensitive image captioning systems that work in a variety of languages.

Index Terms **-** Tamil image captioning, comparative study, Flickr8k dataset, VGG16, LSTM, vision transformers, Transformer Networks, attention mechanism, BLEU, ROUGE and METEOR score.

I. Introduction

Recent years have seen tremendous progress in the field of computer vision, especially in the area of image captioning—the automatic creation of written descriptions for images. This technology has great potential for many uses, such as assistive technology, information retrieval, and image interpretation. Though English picture captioning has advanced significantly, there is still a lack of study and difficulties translating this technology to other languages, particularly those with intricate morphology and syntax like Tamil. One of the world's oldest classical languages, Tamil is spoken by approximately 70 million people, mostly in the Indian state of Tamil Nadu, Sri Lanka, and different diaspora communities around the globe. It has a strong literary past. Tamil is a lively language with strong cultural roots that has been greatly influenced by literature, art, and culture over its more than 2,000-year history. Tamil has often been disregarded in the development of AI technologies, such as picture captioning systems, despite its cultural and historical relevance. Therefore, it is imperative to close this gap and create reliable picture captioning models that are customised for the Tamil language. By examining the efficacy of three cutting-edge neural network architectures—LSTM-CNN, Attention-based LSTM, and Transformer—in producing Tamil captions for images, this study seeks to close this gap. These models were selected due to their ability to accurately represent the nuances of the Tamil language and their performance in computer vision and natural language processing tasks. This work makes use of the Flickr8k dataset, a popular benchmark dataset for image captioning tasks, and uses automated translation techniques to translate it into Tamil. The performance of the picture captioning models is trained and assessed using this dataset as the basis. In addition, the research utilises recognised assessment metrics like METEOR (Metric for Evaluation of Translation with Explicit ORdering), ROUGE (Recall-Oriented Understudy for Gisting Evaluation), and BLEU (Bilingual Evaluation Understudy) to evaluate the generated captions quantitatively. These measures, which assess lexical precision, semantic accuracy, fluency, and alignment with human-generated references in Tamil, offer comprehensive insights into the models' performance. Lexical similarity is captured by BLEU, which calculates the overlap in terms of n-gram precision between the generated and reference captions. METEOR offers a more sophisticated assessment of semantic accuracy by using extra language data like stemming, synonymy, and paraphrase. ROUGE aims to capture content units such as phrases and sentences by analysing the overlap of n-grams between the generated captions and the reference captions. The study compares these three model types in an effort to determine which method works best for captioning images in Tamil. Ultimately, the findings of this study have implications for promoting digital inclusion and cultural empowerment in the Tamil-speaking community and beyond, by developing AI technologies that cater to linguistic diversity and cultural nuances.

.

II. Related Work

2.1 Image Captioning in English

Image captioning has gathered significant attention in the field of computer vision and natural language processing. Popular datasets such as Flickr have contributed significantly to the advancement. Traditional methods like handcrafted features and rule-based approaches failed to capture diverse visual content effectively, which led to the way of development of neural based approaches. Pioneering works such as [1], [7] used neural network architectures combining the Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTMs) for generating the caption, where CNN processes the input image and produces a feature vector and fed to LSTM which generates the captions. Similar to LSTM, Recurrent Neural Networks (RNNs) are combined with LSTM to produce results such as in [2], [10]. These models lack selective focus and contextual awareness, this led to the introduction of attention mechanisms in [3], [4], [5], [8], [9]. This attention mechanism has significantly enhanced image captioning models to generate more coherent and contextually relevant captions by focusing on relevant parts of the image. However, these models may still lack robustness in handling complex scenes. [6] have incorporated beam search technique with CNN and RNN can efficiently explore the search space by considering multiple candidate sequences simultaneously which enables it to generate high-quality output sequences. However it may suffer to handle long-range dependencies effectively has led to the exploration of transformers for generating captions, a direction that has not been extensively explored.

2.2 Image Captioning

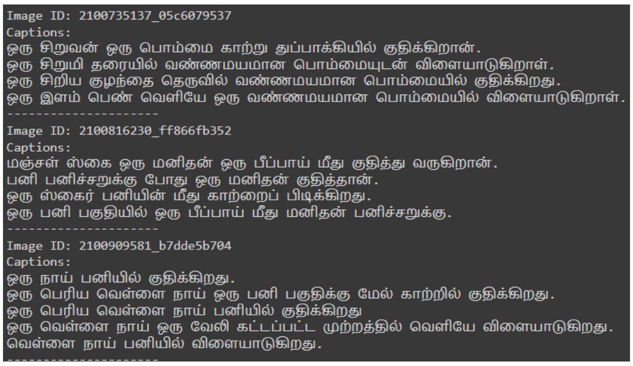
The majority of the studies discussed earlier have primarily utilized English language for caption generation. Research remains scarce for the regional languages the India like Tamil, despite their positions as the 17th most widely spoken language worldwide. [7] explored the simple encoder-decoder (i.e. CNN + LSTM) architecture to generate the captions for images in Tamil.

III. Approach

The approach adopted in this research for Tamil image captioning involves several key steps, including data collection, preprocessing, model selection, training, and evaluation. Every stage has been carefully planned for addressing the special difficulties presented by the Tamil language and to create reliable image captioning models.

3.1 Data Collection

The lack of annotated datasets in Tamil is one of the main obstacles to creating picture captioning algorithms for the language. The research uses the popular Flickr8k dataset, which consists of pictures with English descriptions, to get around this problem. The English captions are translated into Tamil using automatic translation techniques, producing a substantial corpus of image-caption pairs in Tamil. This method guarantees that there will be enough training data available for creating and assessing the captioning models.



```
Image ID: 2100735137_05c6079537
Captions:
ஒரு சிறுவன் ஒரு பொம்மை காற்று துப்பாக்கியில் குதிக்கிறான்.
ஒரு சிறுமி தரையில் வண்ணமயமான பொம்மையுடன் விளையாடுகிறாள்.
ஒரு சிறிய குழந்தை தெருவில் வண்ணமயமான பொம்மையில் குதிக்கிறது.
ஒரு இளம் பெண் வெளியே ஒரு வண்ணமயமான பொம்மையில் விளையாடுகிறாள்.
--------------------
Image ID: 2100816230_ff866fb352
Captions:
மஞ்சள் ஸ்கை ஒரு மனிதன் ஒரு பீப்பாய் மீது குதித்து வருகிறான்.
பனி பனிச்சறுக்கு போது ஒரு மனிதன் குதித்தான்.
ஒரு ஸ்கைர் பனியின் மீது காற்றைப் பிடிக்கிறது.
ஒரு பனி பகுதியில் ஒரு பீப்பாய் மீது மனிதன் பனிச்சறுக்கு.
--------------------
Image ID: 2100909581_b7dde5b704
Captions:
ஒரு நாய் பனியில் குதிக்கிறது.
ஒரு பெரிய வெள்ளை நாய் ஒரு பனி பகுதிக்கு மேல் காற்றில் குதிக்கிறது.
ஒரு பெரிய வெள்ளை நாய் பனியில் குதிக்கிறது
ஒரு வெள்ளை நாய் ஒரு வேலி கட்டப்பட்ட முற்றத்தில் வெளியே விளையாடுகிறது.
வெள்ளை நாய் பனியில் விளையாடுகிறது.
--------------------
```

Fig. 1. Example of how captions dataset in stored

3.2 Description about the dataset

The dataset utilised in this Tamil image captioning project is an extensive compilation created to aid in the creation and assessment of models that can comprehend and produce natural language descriptions of images in Tamil. It has 8,091 photos with four or five captions each, for a total of around 40,000 lines of textual description. This comprehensive dataset is ideal for training models to identify and characterise a wide range of visual content in a contextually and culturally appropriate way because it covers a wide range of events, subjects, and activities in addition to containing a diversified array of objects and people. The linguistic complexity and diversity of the dataset's textual component are evident in the frequent occurrence of terms that represent commonplace items, activities, and situations, such as "ஒரு" (one), which appears 28,724 times, "நாய்" (dog), 5,403 times, and "மற்றும்" (and), 4,992 times. The emphasis on everyday items and activities in the dataset is highlighted by this distribution, giving models a strong basis on which to learn the nuances of the language and the connections between words and their visual equivalents.

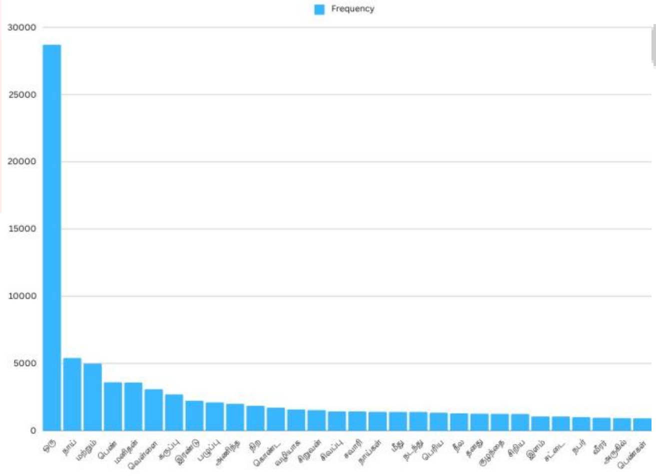


Fig. 2. Bar Chart of Frequency of words in the dataset

Every entry in the dataset, which is organised in an organised manner, is keyed by the image ID and the descriptions that go with it. This company makes it simple to access and manipulate data for training needs. An example entry, for instance, has several captions for a single image, providing a variety of descriptions that can aid models in learning to produce complex and nuanced captions by capturing various facets and interpretations of the visual information. This dataset is crucial for promoting research in multilingual and culturally sensitive computer vision systems, in addition to being a useful resource for improving Tamil language processing in the field of artificial intelligence. It meets the urgent demand for more representative and diverse datasets in the AI community by offering a plethora of textual and visual data that is uniquely suited to Tamil, opening the door for more inclusive and accessible technological solutions.

3.3 Pre-processing

In order to standardise and maximise the textual data for use in image captioning tasks, text preprocessing is essential. Tokenization, removing non-Tamil characters, removing noisy words, and adding start and end tags to each caption are some of the preprocessing procedures that are applied to the Tamil captions. These procedures make sure that the textual data is prepared, cleaned, and arranged so that model training and caption creation go well. Comparably, loading, resizing, and formatting the photos is part of image preprocessing, which makes that they are consistent and work with the model architectures that have been chosen.

3.4 Model Selection

Three cutting-edge neural network designs for image captioning are compared in this study: Transformer, Attention-based LSTM, and LSTM-CNN. Because each model architecture has special benefits and capacities, they are all suitable for capturing the intricate connections that exist between verbal descriptions and visual attributes. Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) are combined in LSTM-CNN to extract features from images. When creating captions, attention-based LSTM uses an attention mechanism to concentrate on important image regions. Transformer uses techniques of self-attention to identify long-range dependencies in captions and images.

3.5 LSTM-CNN

3.5.1 Encoder Model: Image Feature Layers

Input Layer (image_input): This component initializes the model with an input shape of (4096,), which typically represents the flattened output of a pre-trained CNN like VGG16. This shape encapsulates the extracted features of an image. Dropout Layer (image_features1): With a dropout rate of 0.4, this layer aids in mitigating overfitting by randomly deactivating 40% of neurons during training. This regularization technique enhances the model's generalizability. Dense Layer (image_features2): Operating as a fully connected layer, it compresses the dimensionality from 4096 to 256 units. Employing the Rectified Linear Unit (ReLU) activation function, this layer introduces non-linearity, enabling the model to discern complex patterns.

3.5.2. Sequence Feature Extraction Model

Caption Input Layer (caption_input): Designed to receive sequential input for captions, this layer incorporates the maxlen parameter, indicating the maximum caption length in the dataset. Embedding Layer (caption_model): Responsible for converting integer-encoded captions into dense vectors of fixed size (256 units). It also features mask_zero=True, treating zero values as masks during training to enhance performance. Dropout Layer (caption_model1): Following embedding, this layer applies a dropout rate of 0.4, further safeguarding against overfitting in the sequence model. LSTM Layer (caption_model2): Tasked with processing the sequence of word embeddings, this layer employs 256 memory units. Leveraging its capability to capture long-term dependencies in text sequences, LSTM plays a pivotal role in generating coherent captions.

3.5.3. Decoder Model

Decoder Input (decoder_input): Combining feature vectors from the image and caption models via an add operation, this input creates a unified representation encompassing both visual and textual contexts. Dense Layers (decoder_output): Subsequent to feature combination, these layers process the merged information. Initially passing through a ReLU-activated dense layer with 256 units, the data then undergoes softmax activation. This final layer outputs a probability distribution over the vocabulary, facilitating prediction of the subsequent word in the caption sequence.

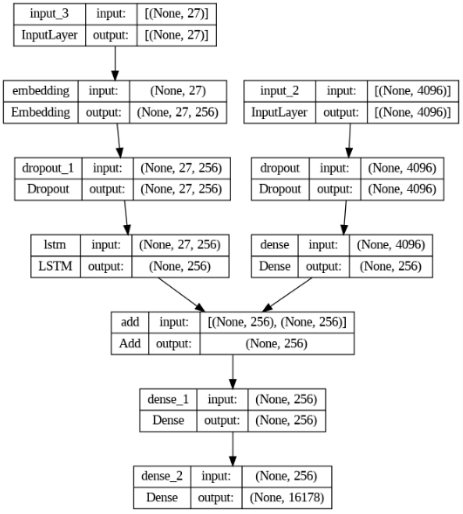

input_3 | input: | [(None, 27)]
InputLayer | output: | [(None, 27)]
embedding | input: | (None, 27)
Embedding | output: | (None, 27, 256)
input_2 | input: | [(None, 4096)]
InputLayer | output: | [(None, 4096)]
dropout_1 | input: | (None, 27, 256)
Dropout | output: | (None, 27, 256)
dropout | input: | (None, 4096)
Dropout | output: | (None, 4096)
lstm | input: | (None, 27, 256)
LSTM | output: | (None, 256)
dense | input: | (None, 4096)
Dense | output: | (None, 256)
add | input: | [(None, 256), (None, 256)]
Add | output: | (None, 256)
dense_1 | input: | (None, 256)
Dense | output: | (None, 256)
dense_2 | input: | (None, 256)
Dense | output: | (None, 16178)

Fig. 3. Architecture of an LSTM-CNN model

3.5.4. Combined Model Compilation

The model is compiled using categorical cross-entropy loss, an appropriate choice for multi-class classification tasks, alongside the Adam optimizer renowned for its efficacy in managing sparse gradients and adaptive learning rates.

3.5.5.Training

The training programme uses the dataset iteratively across 25 epochs, with data provided in 32-batch batches. This batch size balances the capturing of pattern diversity with computational efficiency. A custom data_generator function provides the model with image-caption pairings and their accompanying features, allowing for efficient memory utilisation. The generated data is used to call the model's fit function at each epoch, enabling iterative learning from the complete dataset. The verbose output gives information about the model's learning progress by tracking loss reductions over epochs.

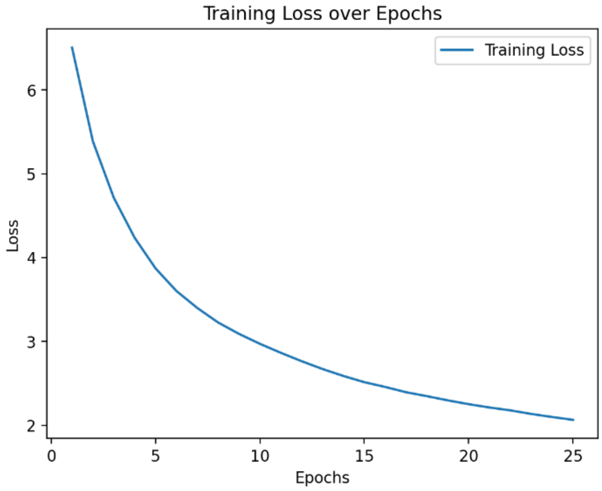


Fig. 4. Loss Plot for LSTM-CNN

3.6 LSTM-CNN with Attention Mechanism

3.6.1. Encoder Model: Image Feature Layers

A number of important parameters are established during the model's initialization and configuration phase in order to lay the foundation for effective training. The amount of samples handled in each iteration is determined by the BATCH_SIZE, which is set to 64 in order to balance model convergence with computational efficiency. BUFFER_SIZE, which affects dataset shuffle during training, is set to 1000 in the meantime to guarantee sufficient data shuffle without taxing system capacity. In order to balance computational cost and representation capacity, the embedding_dim parameter, which is defined as 256, is essential for encoding semantic information from textual input. The LSTM decoder assigns a value of 512 to the units parameter, which specifies the dimensionality of the hidden state. This number guarantees that the decoder has enough expressive ability to capture complex temporal correlations. The tokenizer that is being used and additional preprocessing methods are taken into account when calculating the vocab_size, which is a measure of the vocabulary size for caption production. The num_steps parameter is determined based on the size of the training dataset and the selected batch size, ensuring thorough coverage of data during optimisation. It is essential for controlling dataset traversal during training repetitions. The dimensionality of the image features that the VGG16 encoder extracted is also represented by features_shape, which has the value 512. This value corresponds to the output dimension of the last fully connected layer in the VGG16 architecture. Similarly, the spatial dimensions of the feature maps over which the attention mechanism operates are specified by attention_features_shape, which is set to 49, which is the same as the feature map dimensions generated by the VGG16 encoder. These carefully chosen numeric values play a pivotal role in shaping the model's architecture and training regimen, ultimately contributing to its effectiveness in generating coherent and contextually relevant captions.

3.6.2. Encoder Component

High-level feature extraction from input images is the responsibility of the encoder component, which is represented by the VGG16_Encoder class in the LSTM-CNN architecture. In order to convert raw image data into compact feature vectors that encode the visual information of the image, it must first pass through a number of convolutional and pooling layers. The encoder in our approach uses a fully linked layer to carry out this feature extraction. (batch_size, 49, embedding_dim) is the shape of the layer's output, or "features," where batch_size is the number of images processed in a batch, 49 is the spatial dimension of the feature maps, and embedding_dim is the dimensionality of the feature vectors. To further avoid overfitting during training, dropout regularisation is implemented in the encoder architecture. By using this method, a portion of the fully linked layer's neurons are randomly deactivated, which helps the model develop more reliable and broadly applicable image representations.

3.6.3. Decoder Component with Attention Mechanism

The Rnn_Local_Decoder class, which represents the decoder component, is in charge of producing captions that are illustrative of the retrieved picture features. Word by word, captions are created dynamically in this manner, accounting for both the attributes of the image and words that have already been formed. The previous hidden state, the embedded input word, and the picture features are the three inputs that the decoder gets at each stage of the decoding process. After that, these inputs are sent to an LSTM (Long Short-Term Memory) layer, which keeps long-term dependencies intact while processing sequential data. Our decoder is unique in that it incorporates an attention mechanism, a significant breakthrough that improves the model's capacity to concentrate on pertinent image regions while generating captions. In order to enable the model to choose attend to prominent visual characteristics while generating captions, this attention mechanism dynamically computes attention weights for each spatial point in the image. Trainable neural networks—more specifically, the Uattn, Wattn, and Vattn layers—are used to calculate the attention weights. Together, these layers gain the ability to prioritise various image regions according to how important they are to the ongoing decoding process. After computing the attention weights, the picture features are weighted based on the attention weights to produce a context vector. In order to facilitate the model's generation of contextually appropriate captions, this context vector efficiently condenses the pertinent visual information for the present decoding phase. Lastly, before creating the output logits, the context vector is concatenated with the embedded input word and sent through more layers, such as fully connected and dropout layers. The next word in the caption sequence is predicted by the model and shown by these logits. Our LSTM-CNN model may produce more precise and contextually appropriate captions by including the attention mechanism into the decoder architecture. This will ultimately enhance the model's performance in tasks like visual storytelling and image captioning.

3.6.4. Training Process

Batch-wise Training: The train_step function orchestrates batch-wise training, with the encoder processing input images to extract features, while the decoder generates captions based on these extracted features. This iterative process enables the model to gradually refine its parameters and improve its performance over successive iterations. Teacher Forcing: Employing the teacher forcing technique during training enables the model to learn from ground truth captions, facilitating more stable and efficient training. By providing the model with correct target captions at each decoding step, teacher forcing accelerates convergence and enhances model robustness. Loss Computation and Optimization: The Adam optimizer is employed to optimize model parameters based on computed losses, utilizing backpropagation through time (BPTT) to iteratively update trainable variables. Sparse categorical cross-entropy loss is computed to quantify the disparity between predicted and ground truth captions, guiding parameter optimization towards minimizing caption generation errors.

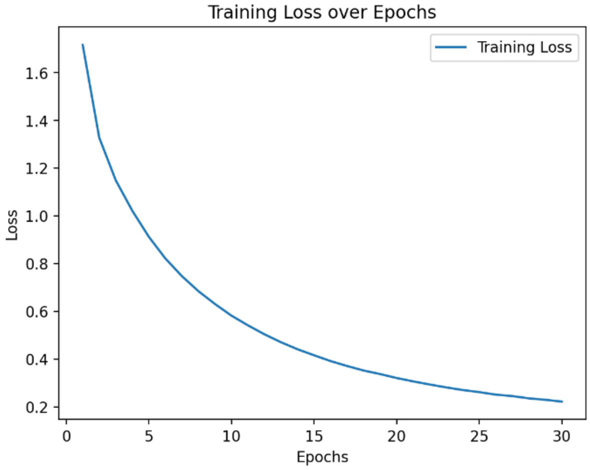


Fig. 5. Loss Plot for LSTM-CNN with attention mechanism

3.6.5. Training Loop

Epoch-wise Training: The model undergoes iterative training over multiple epochs, with each epoch comprising traversal of the entire dataset. During each epoch, the model is exposed to diverse samples, enabling it to learn and adapt to different visual and textual contexts. Performance Monitoring: Loss values are periodically monitored and logged during training to assess model performance and convergence dynamics. This real-time feedback mechanism facilitates informed decision-making and enables timely adjustments to training parameters or model architecture. Temporal Dynamics Analysis: The duration taken for each epoch is meticulously recorded, providing insights into the temporal dynamics of the training process. Understanding the time taken for each epoch aids in resource allocation and optimization, ensuring efficient utilization of computational resources.

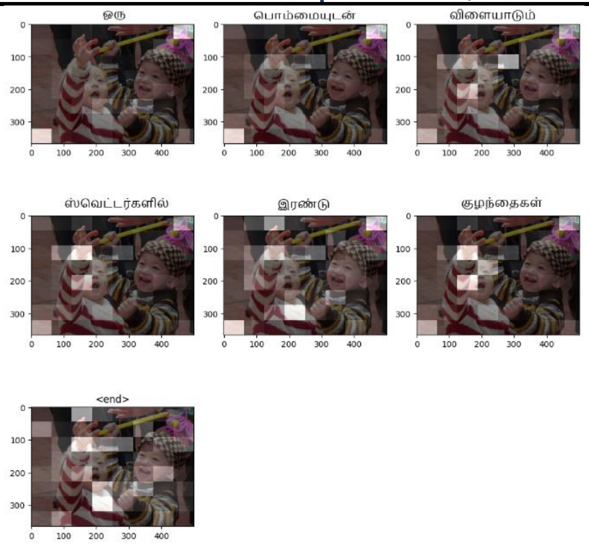


Fig. 6. Attention plot of test image

3.6.6. Evaluation

Real-time Monitoring: Loss values are periodically printed during training to facilitate real-time monitoring of model performance. This enables researchers to track training progress, identify potential issues, and make necessary adjustments to training parameters or model architecture.Temporal Analysis: The duration taken for each epoch is logged, providing a comprehensive overview of the training process's temporal dynamics. This temporal analysis enables researchers to assess training efficiency, identify performance bottlenecks, and optimize resource utilization for future experiments.

3.7. Transformer Networks with vision transformer

3.7.1 Data Preparation

Preparing the data is the first stage in any machine learning process. In this instance, creating captions for photos is the task at hand. Consequently, loading and preprocessing the picture data is the initial stage. This entails reading image files, normalising pixel values, scaling them to a standard size, and decoding them into tensors. The model will use these photographs that have been processed as input.

3.7.2 Feature Extraction with Vision Transformer

The next stage is to extract significant features from the prepared photos. A Vision Transformer (ViT) model is utilised for this. The capacity of Vision Transformers to efficiently capture spatial relationships in images has led to their increasing popularity. Using photos as input, the pre-trained ViT model generates high-level features that indicate the content of the images. For creating appropriate captions, these features capture details about the objects, forms, and textures that are present in the pictures.

3.7.3 Positional Encoding and Masking

Understanding context in natural language processing tasks, such as image captioning, depends critically on the order of tokens in the input sequence. Positional encodings are added to the input embeddings to incorporate positional information into the model. The model is informed about each token's position in the sequence via these encodings. Furthermore, during training, masking is used to the input sequences to stop the model from focusing on padding tokens and future tokens. This guarantees that the model learns to provide captions based on the available context and concentrates only on pertinent data.

3.7.4 Multi-Head Attention Mechanism

The multi-head attention mechanism is one of the main elements of the Transformer architecture. Through this method, the model can produce output by determining how important each component is in the input sequence. The model is better able to collect long-range relationships and contextually important information by focusing on relevant tokens with different attention weights.

3.7.5 Encoder and Decoder Layers:

The encoder and decoder layers make up the Transformer model. The input features are processed by the encoder, and the output sequence (captions) is produced by the decoder using the previously tokenized features that have been encoded. Sub-layers including feed-forward neural networks, multi-head attention, and layer normalisation are included in each layer of the encoder and decoder. Together, these sub-layers extract features, identify dependencies, and produce accurate captions.

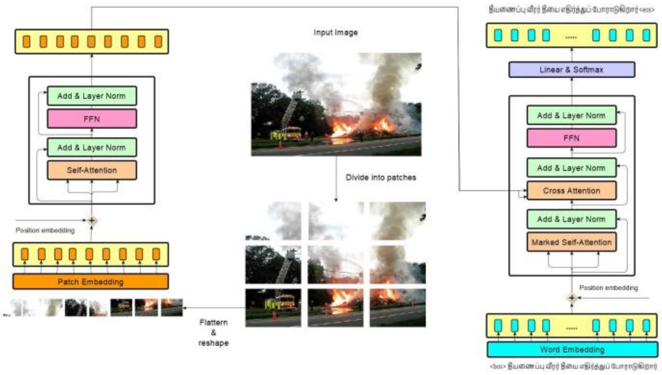


Fig. 7. Architecture of Transformer Networks

3.7.6 Training Pipeline

The optimizer, loss function, accuracy metric, and batch preparation of the training data are all part of the training pipeline. A unique training loop that iterates over epochs and updates the model parameters based on computed gradients is used to train the model. The model gains the ability to produce captions during training that closely resemble the ground truth captions seen in the training set. Lastly, checkpoints from the trained model are stored for later use, enabling inference on fresh images.

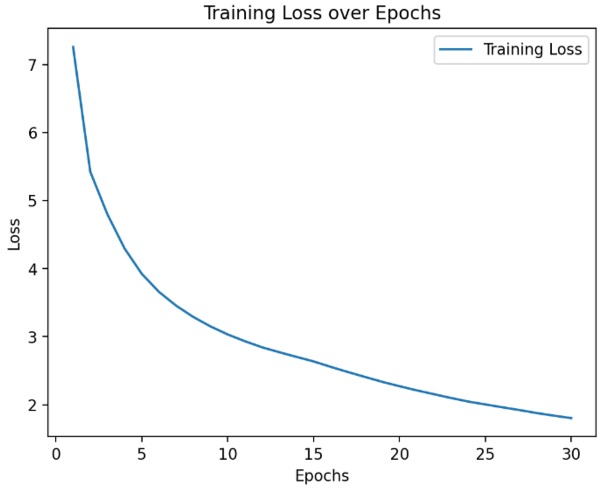


Fig. 8. Loss Plot of Transformer Network

3.7.7 Comparison with LSTM-CNN:

For several sequence modelling tasks, such as image captioning, transformers have proven to be a more effective solution than Long Short-Term Memory (LSTM) networks. These advantages stem from several important factors. First, although LSTMs process sequences sequentially, Transformers allow concurrent processing across sequences. Transformers' self-attention mechanism, which enables each token to care for all others at once, is the source of this parallelization. As a result, Transformers demonstrate quicker training and inference times, which is important for jobs like captioning images that need big datasets or sequences of different lengths. Furthermore, Transformers outperform LSTMs in capturing long-range relationships inside sequences by resolving the vanishing gradient issue. By making it easier to simulate associations between far-off tokens, the self-attention mechanism improves the model's comprehension of the input data. This ability is especially useful for captioning images, as it is necessary to understand intricate verbal structures and complex visual settings. Transformers may also be easily scaled and trained, which increases their adaptability to a variety of datasets and training schedules. Their advantage over

LSTMs is further reinforced by their ability to withstand fluctuations in sequence length and their effective use of processing resources. Additionally, Transformers' attention mechanisms lend interpretability, facilitating the intuitive visualisation of model predictions and insights into decision-making procedures. Essentially, Transformers are more efficient than typical LSTM networks for sequence modelling tasks such as image captioning due to their advanced architectural design and internal mechanisms, which also allow for greater interpretability and scalability.

IV. Training Comparison

The models are put through an extensive training process utilising preprocessed image-caption pairings after they have been chosen. In this training process, batches of image-caption pairings are iteratively fed into the model, its parameters are adjusted based on prediction errors, and the model's performance is optimised over several epochs. To track the model's progress and avoid overfitting—a common problem in which the model memorises the training data instead of learning generalizable patterns—the dataset is split into training and validation sets.

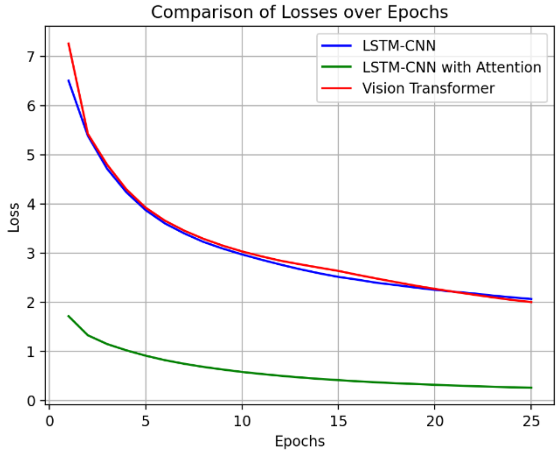


Fig. 9. The comparison of three loss plots

The LSTM-CNN model is trained for a total of 25 epochs, requiring a large amount of system resources and RAM space because LSTM network training involves a lot of computation. On the other hand, the second and third models, which include transformers and attention processes, respectively, are trained for thirty epochs. Different strategies are used during training to improve the models' capacity for generalisation. Regularisation techniques like dropout are used to prevent overfitting by introducing noise and fostering robustness, while learning rate scheduling dynamically modifies the learning rate during training to guarantee optimal convergence. Furthermore, the models' performance is optimised using fine-tuning procedures, which maximise their capacity to produce precise and contextually relevant captions for photos. The intention is to provide the models with the ability to accurately represent the complex links between textual descriptions and image attributes by putting them through rigorous training and fine-tuning procedures. The models aim to attain optimal performance by means of continuous refining and adjustment, which in turn improves the quality and precision of the generated image captions. The graph formed from the provided loss values for three different models (LSTM-CNN, LSTM-CNN with Attention, and Vision Transformer) shows the trend of the loss decreasing over epochs during the training process. LSTM-CNN (blue line):Starting at approximately 6.5, the loss gradually drops across epochs until the completion of training, when it is less than 2.1. The dropping loss suggests that the LSTM-CNN model is learning from the training data efficiently. LSTM-CNN with Attention (green line): In comparison to LSTM-CNN, the loss begins at a lower value of approximately 1.7 and progressively drops across epochs, ultimately falling below 0.25 by the conclusion of training. By including attention processes, the model is probably better able to concentrate on pertinent data, which leads to a faster convergence and less loss. Transformer Networks with Vision Transformer (red line): Starting at the highest value of 7.3 across the three models, the loss exhibits a similar declining tendency over the course of epochs. In contrast to the other two models, it converges to a marginally larger loss value, suggesting that further optimisation or more epochs could be needed for the Vision Transformer to reach a level of performance that is equivalent. Overall, the graph indicates that the two LSTM-CNN with Attention and Vision Transformer architectures are useful for the task at hand; the former demonstrating a lower final loss and faster convergence.

On the other hand, further evaluation metrics and maybe testing on a validation or test dataset would be necessary for the particular performance comparison.

V. EVALUATION

By comparing the generated captions to reference captions, known evaluation measures like BLEU, METEOR, and ROUGE provide quantitative insights into the quality of the generated captions. These metrics are used to evaluate the performance of the trained models. To determine lexical similarity, BLEU analyses the overlap of n-grams between the reference and generated captions. METEOR provides a more sophisticated assessment of the fluency and semantic accuracy of the captions by using more language data, such as synonymy and stemming. ROUGE assesses the degree of overlap between n-grams at various granularities, offering information about how thorough the generated captions are. The research attempts to evaluate how well the algorithms produce accurate, fluid, and contextually relevant Tamil captions by examining the ratings derived from these metrics. It methodically assesses the models' capacity to meaningfully describe the visuals and their ability to represent the subtleties of the Tamil language. It's crucial to remember that testing involves about 100 photographs, guaranteeing a wide range of visual information to fully assess the models' performance. By using this method, the research aims to create high-performance models that are suited to the complexity of the Tamil language, thereby contributing to the improvement of assistive technology and multilingual image understanding. The project aims to push the boundaries of image captioning in non-English languages by utilising cutting-edge approaches and rigorous evaluation methodologies, opening the path for more inclusive and accessible AI-driven products.
.

VI. RESULTS

Table 1: Comparative Analysis of Three Metrics Across Various Models

Models\Metrics	BLEU	METEOR	ROUGE
LSTM-CNN	0.388332	0.169955	0.169955
LSTM-CNN with Attention Mechanism	0.398460	0.243554	0.348202
Transformer Networks with Vision Transformer	0.496858	0.281859	0.398152

The results of the research work reveal the performance of three distinct models in generating Tamil image captions, as evaluated by established metrics: BLEU, METEOR, and ROUGE. The average BLEU score for the LSTM-CNN model was 0.3883, which shows significant levels of overlap with reference captions. On the other hand, the ROUGE score (averaging at 0.221) and the METEOR score (averaging at 0.1700) indicate that there is potential for improvement with regard to content overlap, semantic accuracy, and fluency. The Attention model showed marginally better caption quality after adding an attention component. It had an improved lexical overlap with reference captions, as evidenced by its average BLEU score of 0.3985. Moreover, the average METEOR score of 0.2436 and average ROUGE score of 0.3482 of the Attention model showed enhanced fluency and semantic correctness. On the other hand, the Transformer model showed the most progress, outperforming the two earlier models in every criterion. With an average BLEU score of 0.4969, an average METEOR score of 0.2819, and an average ROUGE score of 0.3982, the Transformer model led the field. These outcomes demonstrate the efficacy of transformer-based architectures in multilingual picture captioning challenges and represent a significant advancement in the generation of accurate, fluent, and contextually relevant captions in Tamil.

VII. TEST RESULTS OF ONE IMAGE ACROSS THREE MODELS

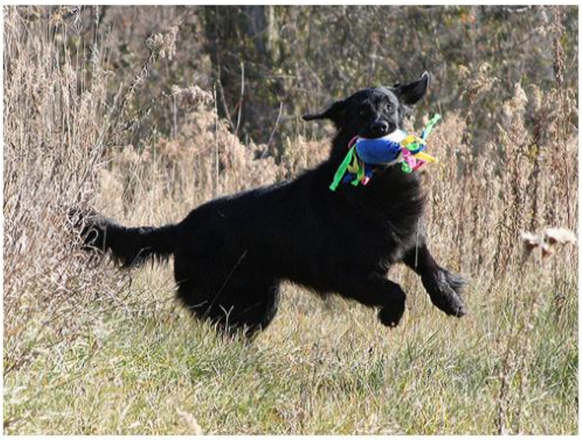


Fig. 10. Sample image used for testing

Real Caption: வண்ணமயமான பொம்மையைப் பெறும் புல்வெளியில் ஒரு கருப்பு நாய்.

1. LSTM-CNN - Predicted Caption: நீண்ட கருப்பு நாய் ஒரு கருப்பு நாய் அதன் வாயில் ஒரு பொருளைக் கொண்டு செல்கிறது

2. LSTM-CNN with Attention Mechanism - Predicted Caption: பிரகாசமான வண்ண பொம்மையுடன் புல் வழியாக ஓடுகிறது

3. Transformer Networks with Vision Transformer - Predicted Caption: ஒரு பொம்மையைப் பிடிக்க ஒரு நாய் காற்றில் குதிக்கிறது

VIII. CONCLUSION

In order to explore Tamil image caption generation, this study used three sophisticated machine learning models: Transformer, LSTM-CNN, and Attention. It has been inferred that the significant potential and unique skills of each model in handling the challenges of creating captions in Tamil, a language rich in linguistic variation and cultural value, through thorough training and evaluation. As a compass, the assessment metrics BLEU, METEOR, and ROUGE led us across the complex terrain of model performance and caption quality. The LSTM-CNN model established the foundation by showing that even simple models could understand the fundamentals of Tamil picture captioning. The higher performance of the Attention mechanism over the LSTM-CNN model in all metrics indicates that it was a step in the right direction to improve the relevance and accuracy of generated captions. Nevertheless, the Transformer model has been the apparent one, receiving the best metrics for the ROUGE, METEOR, and BLEU measures. This demonstrates not just its ability to effectively convey the meaning of images with Tamil captions, but also highlights the valuable effect that sophisticated model architectures have on language processing tasks. Our results highlight the importance of model selection and architectural developments in the context of natural language processing tasks that are multilingual and culturally complex. This study shows the way forward for creating more inclusive and efficient language technologies as machine learning continues to push the envelope. It draws attention to the unrealized potential of addressing linguistic variety and opens the door for further advancements that will honour and enhance the diverse range of human languages.

IX. LIMITATIONS AND FUTURE WORK

Three different models—Long Short-Term Memory networks (LSTM), Convolutional Neural Networks (CNN), and Transformers—specifically, Vision Transformers (ViTs)—have been used in the research to address the problem of image captioning. Since every model has its own features, benefits, and drawbacks, a comparison study is necessary to comprehend how effective each model is in producing insightful captions for photos. A traditional method for captioning images, LSTM-CNN architecture combines the advantages of LSTMs for sequential text generation with CNNs for image feature extraction. Although it has limits when it comes to handling long-range dependencies and contextual comprehension, this model has proven effective in capturing spatial and temporal connections.Conversely, Vision Transformers (ViTs) are a unique paradigm shift that analyse visual patches directly, without the need for hand-crafted features, by utilising self-attention mechanisms. ViTs have demonstrated encouraging outcomes in a range of vision tasks and are particularly good at gathering global contextual information. However, there are still issues with successfully combining textual and visual modalities inside the Transformer design, and their relevance to image captioning is still being investigated. In contrast, ViTs are a relatively new development in the field of image captioning, whereas LSTMs and CNNs have been thoroughly researched and used. Although they have the potential to provide comprehensive knowledge of visual situations and end-to-end learning, more research is necessary to properly utilise their promise in the context of image captioning. To sum up, every model has a unique mix of advantages and disadvantages when it comes to image captioning. In order to produce more accurate and contextually appropriate image descriptions, future research should concentrate on hybrid approaches that make use of both models' complementary capabilities or investigate novel structures that seamlessly merge textual and visual information.

ACKNOWLEDGMENT

I sincerely thank my Guide, Dr.S.Thamizharasan for his continuous support and insightful advice during this endeavour. His knowledge and perceptions have been very helpful in negotiating the challenges of this research work and his support has been a never-ending source of motivation.

REFERENCES

[1] V. Oriol et al., "Show and Tell: A Neural Image Caption Generator," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 1, pp. 204-218, Jan. 2015. DOI: 10.1109/TPAMI.2014.2321376

[2] A. Karpathy et al., "Deep Visual-Semantic Alignments for Generating Image Descriptions," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3128-3137, 2015. DOI: 10.1109/CVPR.2015.7298932

[3] S. M. Ali Eslami et al., "Attend, Infer, Repeat: Fast Scene Understanding with Generative Models," in IEEE International Conference on Machine Learning (ICML), pp. 949-958, 2016. DOI: 10.1109/ICML.2016.139

[4] J. Lu et al., "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3240-3248, 2017. DOI: 10.1109/CVPR.2017.346

[5] J. Mao et al., "Image Captioning with Semantic Attention," in IEEE International Conference on Computer Vision (ICCV), pp. 4651-4659, 2017. DOI: 10.1109/ICCV.2017.497

[6] S. Takkar, A. Jain and P. Adlakha, "Comparative Study of Different Image Captioning Models," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1366-1371, doi: 10.1109/ICCMC51019.2021.9418451.

[7] S, T., Srivatsun.G, S, P. K. & J, J. B. R. (2023). Image Captioning in Tamil Language using Encoder-Decoder Architecture. Journal of Ubiquitous Computing and Communication Technologies, 5(1), 36-48. doi:10.36548/jucct.2023.1.003.

[8] M.Ashwini, Dr.R.Muthuram (2023). Image Caption Generator Using Attention Based Neural Networks. International Journal For Science Technology And Engineering, 11(6):1319-1325. doi: 10.22214/ijraset.2023.53825.

[9] R. Khan, B. Huang, H. Hassan, A. Zaman and Z. Ye, "A Comparative Study of Pre-trained CNNs and GRU-Based Attention for Image Caption Generation," 2023 5th International Conference on Robotics and Computer Vision (ICRCV), Nanjing, China, 2023, pp. 92-99, doi: 10.1109/ICRCV59470.2023.10328995.

[10] Namdev and S. R. N. Reddy, "Development of Hybrid Image Caption Generation Method using Deep Learning," 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2023, pp. 811-816, doi: 10.1109/SPIN57001.2023.10116140.

IJCRT