IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

ExpressImage: Conveying images with captions

G. Indu[1], Darishetty Sai Varshini[2], Sane Nileesh[3], CH.Likhitha[4]

Asst. Professor, Computer Science and Engineering (Data Science) Institute of Aeronautical Engineering, Dundigal, Hyderabad[1]

Computer Science and Engineering (Data Science) Institute of Aeronautical Engineering, Dundigal, Hyderabad[2-4]

Abstract: With the exponential growth in the creation and sharing of images across online platforms, there is a pressing need to develop systems that enable machines to understand and generate descriptions for these images. While humans can easily comprehend visual content, automated image captioning systems are necessary to provide meaningful descriptions for use in various applications. Extracting semantic information from photos and expressing it in natural language is the aim of image captioning. This entails closely examining photos to pinpoint important details, important things, and the connections between them. Convolutional neural networks (CNNs), in particular, are utilized in deep learning techniques to extract these visual properties. A transformer-based model is then used to process these features and provide textual captions that make sense. The approaches for picture captioning are examined in this work, with a focus on the function of transformers and CNNs in automating the creation of descriptive captions. In order to progress fields like computer vision, artificial intelligence, and human- computer interaction, the study attempts to improve machines' capacity to comprehend and describe visual content.

Keywords: Image, Caption, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) , Long Short Term Memory(LSTM), Neural Networks.

I. INTRODUCTION

With the rapid proliferation of digital images generated from various online sources, such as social media, e-commerce platforms, and content-sharing websites, the demand for automated systems capable of interpreting and describing visual content has grown significantly. While it is easy for people to interpret images, creating machines that can mimic this comprehension is a difficult task. This gap is filled by automatic picture captioning, which translates visual information into plain language descriptions. This has practical implications in human-computer interaction, content-based image retrieval, and accessibility for the blind.

Current developments in deep learning, particularly the advent of Transformer models and Residual Networks (ResNet), have fundamentally altered computer vision and natural language processing. ResNet's deep architecture and residual connections allow it to extract rich and hierarchical picture attributes, which solves difficulties like the vanishing gradient issue that traditional CNNs have. But because to its self-attention mechanism, the Transformer model—which was first created for machine translation jobs—has shown to be remarkably successful for sequence- to-sequence activities. The model can recognize long-range relationships and produce more grammatically and contextually meaningful sentences thanks to this method.

The main drawbacks of earlier RNN-based models—such as their incapacity to manage lengthy sequences and their neglect of pertinent visual regions—are addressed by the adoption of the Transformer design. The model can dynamically shift its focus to different objects or regions in the image by utilizing the multi-head attention mechanism. This allows the model to produce captions that are more precise, in-depth, and contextually linked with the image content.

The quality and fluency of automatic image captions have been greatly enhanced by recent developments in deep learning, especially with the use of Convolutional Neural Networks (CNNs) for feature extraction and transformer- based models for sequence creation. CNNs are good at recognizing and encoding visual features, while transformers are good at modeling text's long-range dependencies so that descriptions can be produced that seem human.The amount of photographs created and disseminated on a daily basis in the digital age has increased dramatically across a variety of platforms, including social media, news websites, and e-commerce sites. Consequently, it is now more important than ever for robots to be able to not

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

just assess these images but also produce meaningful descriptions for them. The process of automatically translating visual content into meaningful written descriptions—known as automatic image captioning—has become indispensable in a number of fields, such as automated content generation, image-based search engines, and accessibility for the blind. Although machines are unable to comprehend an image's semantic content as well as humans do, this is a challenging task. It calls for advanced methods that can close the gap between NLP and visual perception. In order to provide a meaningful caption, image captioning needs to do more than just locate and identify the things in an image. It also needs to infer the relationships between these objects and their surroundings. In the past, retrieval-based systems or template-based techniques were used to address image captioning; however, these methods frequently lacked flexibility and accuracy. But the area has undergone a revolution with the introduction of deep learning. For extracting high-level visual features from images and integrating them with sequence-based models like Recurrent Neural Networks (RNNs) or, more recently, transformer models, modern image captioning systems usually rely on Convolutional Neural Networks (CNNs). CNNs are very good at recognizing and encoding the important visual elements of an image, like scenes, objects, and textures, whereas transformers are very good at creating word sequences that make sense and follow grammar rules. The implications of being able to automatically create captions for photos are extensive. It facilitates automatic content development for internet platforms, provides effective picture retrieval systems, and improves accessibility for visually impaired people by converting visual information into text. Furthermore, picture captioning is essential for surveillance, driverless cars, and healthcare—all fields where accurate interpretation of visual data is critical. A system that can precisely define abnormalities in an MRI or X-ray, for instance, may help medical practitioners make well- informed decisions.

Fig.1 person is walking on the side of cliff

II. LITERATURE REVIEW

A comprehensive Survey of Deep Learning for Image Captioning [1] Using a deep learning based approach, the image captions are produced after the photos are carefully inspected to identify important features, objects, and connections—all necessary for the process. The generated caption is then reviewed and edited based on the syntactic and semantic patterns, emphasizing CNN-LSTM combinations as a common technique -but also mentioning drawbacks like overfitting and difficulties with complex relationships in the image.

Image caption generation with high-level image features [2] provides a thorough analysis of current developments in machine learning methods for identifying malevolent insider threats. They talk about the advantages and disadvantages of using supervised, unsupervised, and semi-supervised machine learning techniques. The review emphasizes how important it is to have resilient, flexible systems that can deal with insider threats' ever-changing nature. Additionally, it assesses several algorithms and suggests future lines of inquiry to improve detection skills.

Show and Tell: A Neural Image Caption Generator [3] shown a technique that can generate plain-language descriptions of photos automatically. An LSTM and a CNN are combined in this model. While the LSTM creates a description by predicting one word at a time based on the picture features and already created words, the CNN is used to extract features from the image. The primary problems that led to the investigation of Transformer models were the constraints of fixed-length representations and challenges with lengthy sequences.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention [4] created a model with automated learning capabilities to recognize the contents of given photos. This model is trained via the application of backpropagation techniques. Convolutional methods are first employed to locate and retrieve the features. Initially, the features are located and extracted using convolutional techniques. After that, the features are processed by RNN, which uses LSTM to build the description word by word.

A region-based image caption generator with refined descriptions [5] The caption generator has a unique deep learning architecture based on regions. To offer the in-depth descriptions of the image, it employs an encoder-decoder, which is a language generator contained within two RNNs, an object detector, and RNN-based attribute prediction. The region-based method enhances image captions by focusing on specific areas of the image; nonetheless, it generates long sequences and has problems with vanishing gradient and ambiguity.

III. RESEARCH METHODOLOGY

One of the CNN architectures in the Residual Network Family is the ResNet18 model. There are eighteen deeply layered levels. The pretrained version of the network, trained on more than a million images, is stored in the ImageNet database.

It is capable of categorizing photos into 1000 groups. ResNet designs use residual learning, which is the process of adding shortcut or skip connections that omit a layer or layers. The network can learn residual mappings thanks to these skip connections, which facilitates the training of very deep networks.

The issue of disappearing gradients is resolved using residual learning, which also makes it possible to train deeper architecture. The vanishing gradient issue in deep networks is addressed via residual blocks, which are a basic component of the ResNet paradigm.

It presents skip connections, sometimes referred to as shortcuts. These connections cause the gradient to propagate more easily during training by skipping a few layers. This aids in model training without causing a drop in performance.
ResNet18 is extensively utilized in transfer learning for many computer vision tasks, having been trained on millions of images. As its name implies, it is a shallow network with eighteen layers, best suited for smaller datasets or less computationally demanding applications. It accepts input photos with a standard pixel size of 224 x 224.

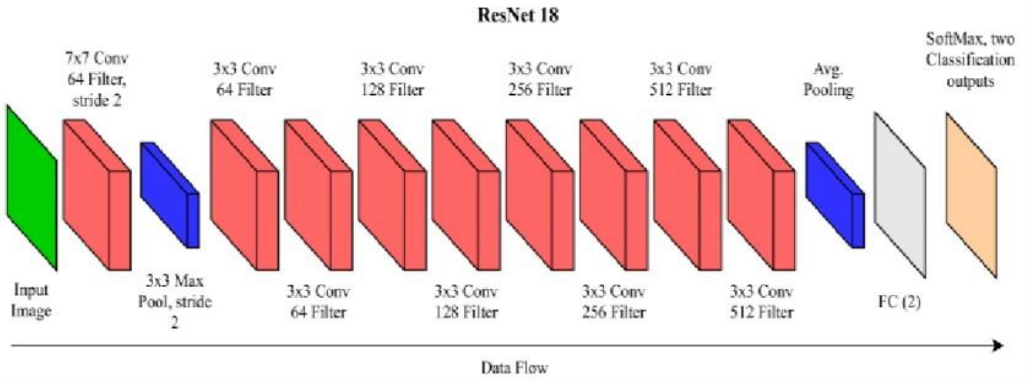

Fig. 2 ResNet18 Architecture

Another CNN which can be used is MobileNetV3. They are created with the goal of maximizing efficiency and accuracy for deployment in contexts with limited resources, such mobile or edge devices. To lower computing costs and model sizes, this architecture incorporates novel techniques such as squeeze-and- excitation modules, linear bottleneck layers, and depthwise separable convolutions. Moreover, it uses Neural Architecture Search (NAS) to automate architecture design and optimize it for power and performance. For real-time applications with constrained power and processing resources, MobileNetV3 has grown in popularity.

DOI: 10.17148/IJARCCE.2025.14429

Even though they are quite precise, traditional architectures like ResNet and VGG are too costly to compute and not suitable for most applications. Google released MobileNetV3 in 2019 with the express purpose of addressing these shortcomings by providing a highly effective model that yields competitive accuracy with minimal processing overhead. Squeeze-and-excitation (SE) modules, which model interdependencies between channels to recalibrate feature maps, are integrated into MobileNetV3. By suppressing less significant variables and concentrating on relevant ones, SE blocks increase the efficiency and accuracy of the model. The network can discover which features are most important for the current task by using this strategy. The more effective hard-swish activation is used by MobileNetV3 in place of the conventional ReLU activation mechanism. This activation function has a hard-swish definition. Further speedups without compromising model performance are possible with this activation function, which is computationally less expensive to implement and roughly matches the swish function. When comparing MobileNetV3 to more conventional architectures such as ResNet, a better balance among accuracy and efficiency is achieved. Using five times fewer parameters and ten times fewer FLOPs, MobileNetV3- Large achieves classification accuracy performance on the ImageNet dataset comparable to ResNet-50.

A. Data Pre-Processing

Data preprocessing is the critical step in our research aimed at enhancing insider threat detection. The main goal of these procedures is to guarantee quality, consistency, and relevance of raw data in order to prepare it for efficient modeling.

Data Preprocessing

- **Tokenization:**

Tokenization is the process of breaking each caption down into individual words or tokens once the caption text file has first been transformed into a dataframe.

- **Case Normalization:**

Then, in order to preserve consistency, every token is changed to lowercase.

- **Adding EOS tags:**

Adding special tokens ‘<start>’, ‘<end>’ to identify the beginning and end of each caption. This helps model learn when to start and stop generating captions.

- **Removing single character words:**

Removing single character words since they do not contribute anything meaningful..

- **Padding Sequences:**

Ensure all the captions are of equal length by adding extra ‘<
> ’ tokens up to maximum length of all captions. This is required because transformer models need fixed-length inputs**.**

- **Building vocabulary:**

Creating a list with all the words from captions. Using Python’s ‘collections’ module to count the frequency of each word in the dataset. Sorting the vocabulary based on word frequency in descending order to assign indices.

- **Mapping Words to Indices:**

Assigning a unique index to each word based on its frequency rank in the vocabulary. Creating Index-to-Word and Word-to-Index dictionaries to convert between words and their corresponding indices. Because transformer cannot understand the natural language.

- **Dataset Splitting:**

Segment the dataset into training, validation, and testing subsets

B. Model Training

First, a predetermined number of epochs is established. One full training dataset loop is known as an epoch. It is a crucial hyperparameter that controls how the model is trained. To solve the space issue, the training data is divided into smaller batches. The smaller batches can be simply fed to the model to train it. These components are referred to as batches. Additionally, epoch refers to the process of feeding all of the batches into the model at once.

The loss is calculated as the difference between the output target sequences and the actual output. Backpropagation can be done to update the model parameters using the optimizer. We monitor the total training loss and use a set of criteria to identify the best model.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

Data Preparation
Before training, the datasets underwent extensive preprocessing to enhance the quality of the training process.

Training Process
The training phase involved splitting the processed datasets into training and testing subsets. A common practice was to allocate 90% of the data for training and 10% for evaluation. The data is sent to ResNet18 encoder to take the features out of the pictures. The transformer architecture receives these features and applies them to the model.

Evaluation and Validation
Post-training, the models were evaluated using the reserved test set to assess their performance. Key metrics such as BLEU score are used to evaluate the model.

C. Transformers

In the year 2017, a paper Attention Is All You Need [7] was published which revolutionized the way natural processing tasks were done. A simple network architecture 'Transformer' was introduced which is based upon the attention mechanism

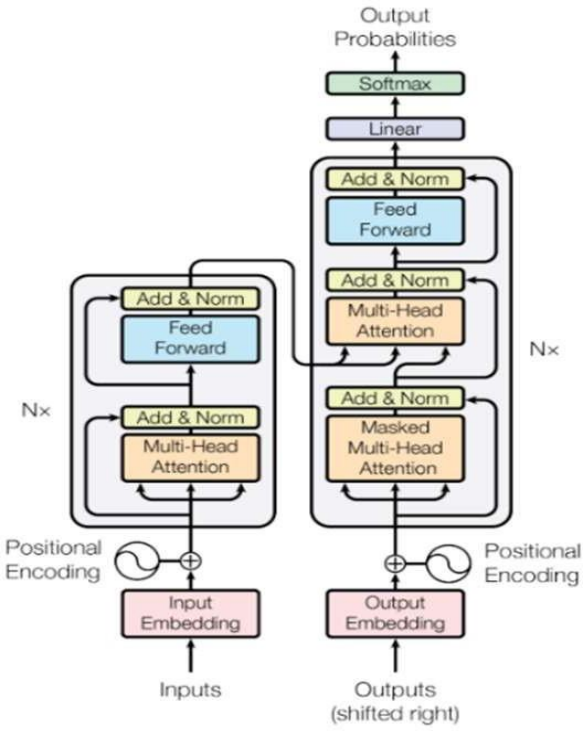

Fig. 3 Transformer Architecture

Encoder
The encoder extracts the features from an input sentence and these features are taken as input to the decoder to produce an output sentence. The input sentence is fed through each of the many encoder blocks that make up the encoder, and the output from the last encoder block is fed into the decoder, which also includes numerous decoder blocks.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

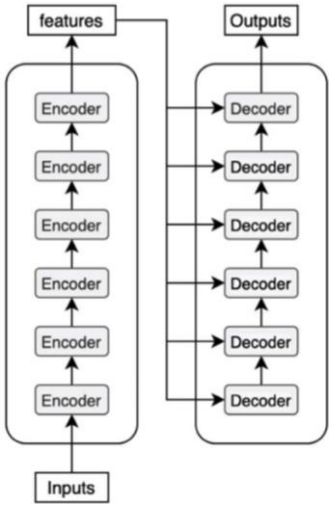

Fig. 4 Encoder and Decoder

Decoder

The decoder receives the extracted features, which are nothing more than the embedded vectors. The next decoder block's input is the previous output, and the decoder operates in an auto- regressive manner. However, since there is no input in the first decoder block, we start the translation by passing the <SOS> beginning-of-sentence to the decoder.

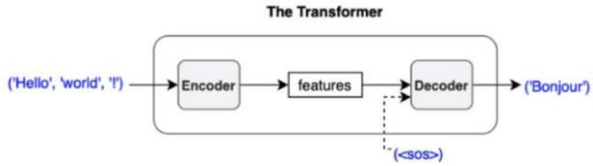

Fig. 5 Decoder

The decoder adds data to the embedding vector to aid in the creation of the first translated word. The decoder's output vector undergoes a linear transformation that modifies its dimension from the embedding vector size to the vocabulary size. Finally, the layer is transformed into probabilities via the softmax layer. It selects and creates the term based on the probability. Using a greedy method, it selects the term that has the highest probability. This initial output token is now supplied as an input to the subsequent decoder.

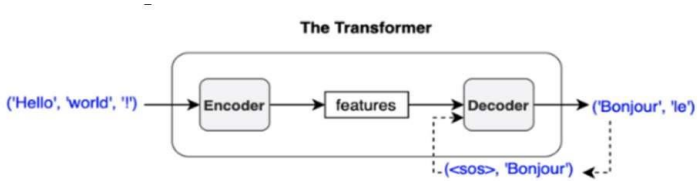

Fig. 6 Decoder The output is again given as input to the decoder.[2]

Again this input is converted into an embedding vector along with positional encoding and given to the decoder.(Output Embedding)

Attention Mechanisms for Multiple Heads and Masked Attention Mechanisms for Multiple Heads

A single attention score for a particular word in a single head attention mechanism is calculated by comparing every word in the phrase and adding them all up. Nevertheless, this process is repeated in parallel in numerous attention heads, each with a different set of parameters. By focusing on various context- related elements, each attention head attempts to capture

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

the various elements of the input sequence. The final output of a multi-headed attention layer is then created by concatenating and linearly transforming each attention head's outputs.

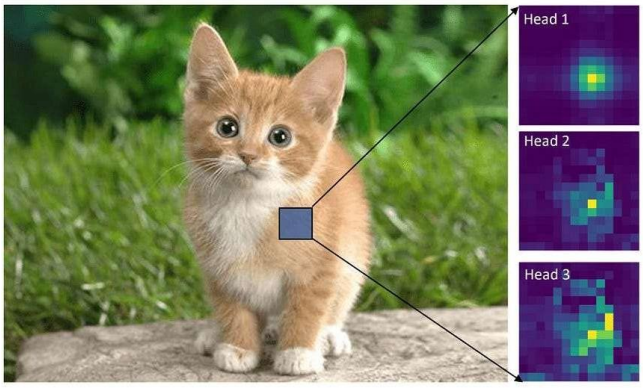

Fig. 7 Visualization of Attention weights on an image.

Feed Forward Network

The ReLU (rectified linear unit) layer plus another linear layer that processes each embedding vector independently using the same weights make up this network . The activation function, or ReLU, gives the model non-linearity so that it may learn intricate and delicate patterns that linear models cannot.

Residual Connections and Normalization

The current layers receive the prior embeddings from residual subjects,connections, which adds more information to the embedding. Additionally, a normalization layer that seeks to stabilize the embedding vectors' mean and standard deviation comes after each residual link. A fluctuating mean and standard deviation lead to unstable and sluggish training. The layer normalization was first introduced by the group led by Geoffrey Hinton.

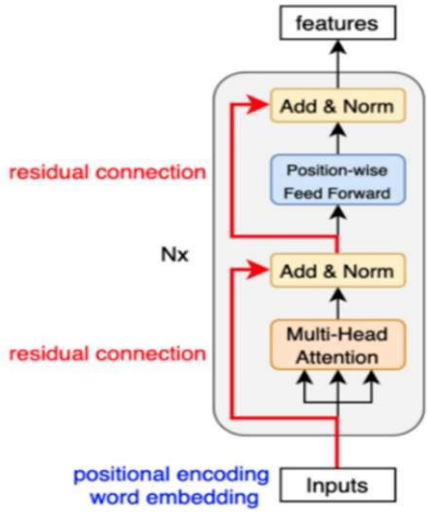

Fig. 8 Residual and normalization

IV. IMPLEMENTATION AND RESULTS

Involving systematic methodologies to validate the effectiveness of the model. This section outlines the steps taken to design, implement, and evaluate the experiments, ensuring robust and reproducible results.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

A. *Datasets overview*

Dataset Source:
(https://www.kaggle.com/datasets/adityajn105/flickr8k)

- A popular dataset for tasks like object recognition, image captioning, and visual-semantic analysis is Flickr8k. With eight thousand photos and five descriptive phrases per image, it's the perfect baseline for testing and training image-caption generating models. This dataset has been widely used to develop deep learning-powered visual comprehension in both commercial and academic research.

- **Dataset Composition**
- **Images**

Number of Images: 8,092 images in total.
Source: Images are collected from Flickr, an online photo- sharing platform.The dataset is diverse in terms of context and content because these photographs encompass a wide range of commonplace objects and events.

Content: Images in the dataset feature a variety of

- **People** (performing various actions such as playing, walking, etc.)
- **Animals** (dogs, horses, etc.)
- **Objects** (toys, vehicles, sports equipment)
- **Scenes** (urban, rural, indoor, outdoor)

- **Captions**

Number of Captions: 40,000 captions (5 captions per image).
Format: Five distinct captions in natural language are included with every image in the dataset. These subtitles, which highlight the main ideas and activities in the picture, were painstakingly annotated by people.

B. *Environmental Setup*

Data Preparation: The first step involved preprocessing the datasets to ensure they were clean, relevant, and suitable for analysis. Tokenization is carried out by dividing each caption into individual phrases or tokens in the first stage of data preparation, which involves first converting the caption text file into a dataframe. Case normalization is used to provide consistency, changing all tokens to lowercase. To help the model understand when to start and stop creating captions, special tokens like " <start>" and " <end>" are introduced to indicate the beginning and end of each caption. Words with a single character are eliminated from the dataset since they usually don't provide any significant information.

Padding is used to make sure that all captions are the same length. Sequences are made longer with extra "<>" tokens until they reach the maximum length of the caption. For transformer models, which need fixed-length inputs, this step is essential. Subsequently, the vocabulary will be expanded by creating a list of every word that appears in the captions and counting how often each word appears using Python's 'collections' module. Next, the vocabulary is arranged in descending order by word frequency, with each word being given a unique index.

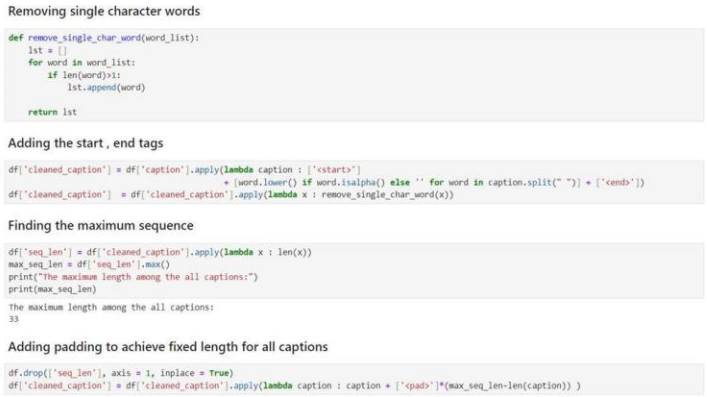

Fig. 9. Data Preparation

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

Models: We trained two distinct models one with ResNet18 being used as CNN and other one being MobileNetV3 for extracting the features of the input images.

- **ResNet18:** Utilizing the resnet18 layers for the extraction of image features.
- **MobileNetV3:** Using MobileNetV3 for the feature extraction.

C. *Experimental Execution*

Training the Models: Each model was trained separately on the relevant features extracted from the preprocessed datasets. A standardized training procedure was implemented. The data is separately split before the training into training data and validation data. The data is split in the ratio of 90% training data and 10% testing data randomly.

The model is placed in training mode for a predetermined number of epochs during the training process. We have set the number of epochs as 30 initially. The code initializes the count of processed words, as well as accumulators for training and validation losses, for every epoch. The model handles batches of picture embeddings and matching caption sequences during the training loop. It computes the gradients, changes the parameters of the model using an optimizer, and calculates the loss while applying a mask to disregard padding tokens. Once the model has been trained, the algorithm assesses its performance using a validation dataset, calculating losses once again and turning off gradient computations to maximize speed. The model that performs the best is saved if the validation loss is lower than in earlier epochs.

D. *Performance Evaluation*

Evaluation Metrics: Each model's performance was assessed using a comprehensive set of evaluation metrics, including precision, recall, F1-score and accuracy. These metrics provided valuable insights into the models' effectiveness in identifying malicious activities and maintaining a low false positive rate. But for a model such as image captioning models which is a generation task, the model generates a string of words, or captions. It is difficult to define accuracy because the output is a collection of words rather than a single label. And, Hence the same photograph may have more than one appropriate caption. This heterogeneity is explained by the BLEU, METEOR, or CIDEr metrics, which take synonyms and paraphrases into consideration.

- **BLEU:** One popular metric for assessing the quality of text produced by machine translation systems and natural language processing activities, such as captioning images, is the **BLEU (Bilingual Evaluation Understudy)** score. BLEU is a 2002 invention of Papineni et al. that calculates the overlap of the generated text (hypothesis) with one or more reference texts.

The generated and reference texts do not overlap when the BLEU score is 0; on the other hand, a score of 1 denotes perfect matches across all n-grams.

Model	BLEU SCORE
ResNet18	0.239457182
MobileNetV3	0.566227766

Fig. 10. BLEU scores

- **METEOR:** An automatic evaluation metric called METEOR (Metric for Evaluation of Translation with Explicit Ordering) is mostly used to score the quality of text created by machines, especially for jobs like image captioning and machine translation. METEOR, which was created in 2005 by Banerjee and Lavie, is a more sophisticated measure that takes synonymy, stemming, and paraphrase matching into consideration. It attempts to overcome some of the shortcomings of previous metrics, such BLEU.
- The METEOR score is a number between 0 and 1, where 0 denotes no overlap and 1 denotes a perfect match between the generated text (hypothesis) and the reference text. The metric calculates the score by evaluating the alignment between the generated and reference texts on a word-by-word basis.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

Model	METEOR
ResNet18	0.349457235
MobileNetV3	0.466295721

Fig. 11. METEOR scores

The loss value, which represents the model's prediction error, is plotted on the Y-axis and the number of epochs on the X-axis in a loss function plot.

- **Blue line:** Depicts the loss of training.
- **Orange line:** Depicts the validation loss.

- Overfitting may be indicated if the training loss dramatically drops while the validation loss either stays roughly constant or begins to rise. This indicates that while the model may be learning the training set too well, it may find it difficult to generalize to new data. A steady drop in both training and validation loss indicates that the model is generalizing well and learning efficiently. A model may be underfitted, which means it is not complex enough to reflect the underlying patterns in the data, if both the training and validation losses stay high or even rise.

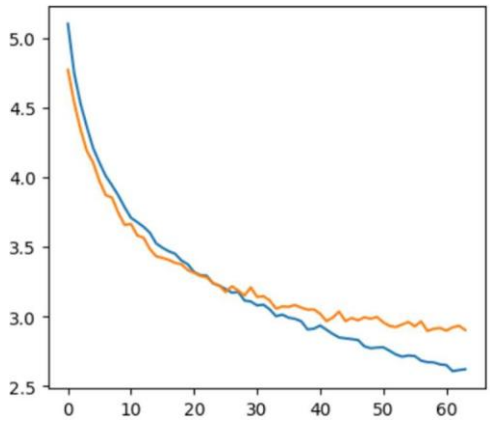

Fig. 11 Loss Function

- Masked Accuracy is a line graph that illustrates the performance of a machine learning model on a specific subset of data or with specific constraints. Plots often show two lines. First off, accuracy training for Masks indicates how accurate the model is on the training dataset, or the dataset that was utilized to train the model. Verification Masked Accuracy measures the model's performance on data that hasn't been seen yet, or validation data, which is a distinct set of data.

- The number of epochs (training iterations) is shown on the X-axis. The masked accuracy, a task-specific statistic, is shown on the Y-axis.

It likely measures the model's accuracy on a specific subset of the data or with certain constraints.

- **Blue line:** Depicts masked accuracy.
- **Orange line:** Depicts the validation masked accuracy.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14429

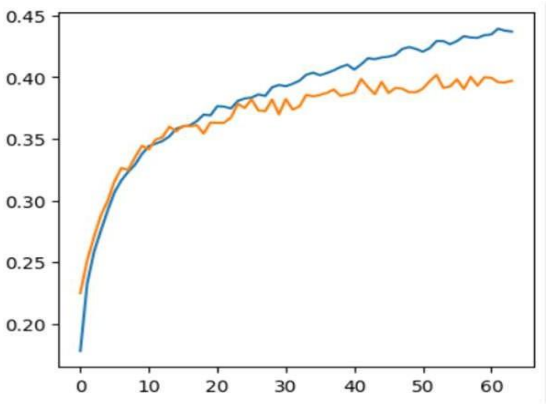

Fig. 12 Masked Accuracy

E. *Results*

Result :

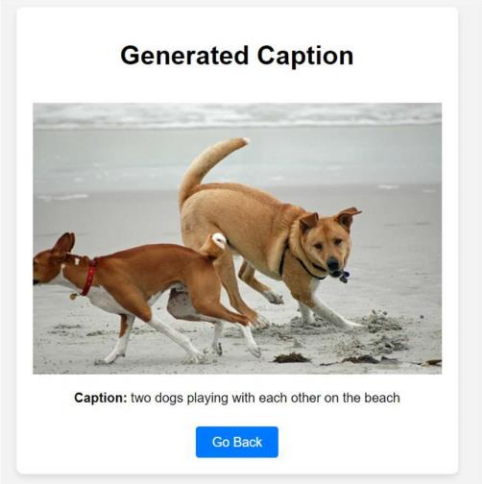

Fig. 13. Output1

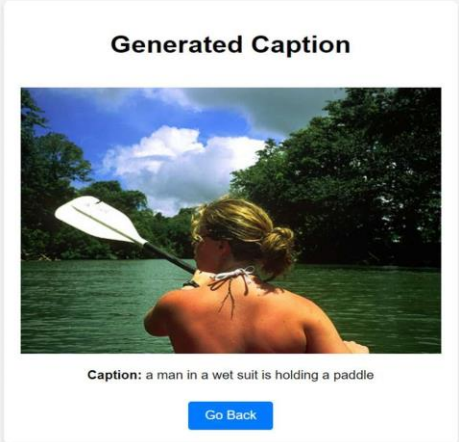

Fig. 14. Output2

DOI: 10.17148/IJARCCE.2025.14429

VI. CONCLUSION AND FUTURE SCOPE

Using deep learning architectures, we created two picture captioning models for this project: one based on ResNet-18 and the other on MobileNetV3. Convolutional neural networks, or CNNs, are effective at comprehending and interpreting visual content. This was demonstrated by the fact that both models were taught to produce meaningful captions for a variety of photographs. With its deep residual connections, the ResNet-18 model demonstrated exceptional ability to capture complex visual features, resulting in comparatively high caption generation accuracy. On the other hand, the MobileNetV3 model, which had speed and efficiency tuned, offered an amazing trade-off between computational cost and performance. Because of this, it was especially appropriate for circumstances involving restricted resources and real-time applications. When evaluation measures like BLEU and METEOR were used to compare the performance of the two models, it was found that the MobileNetV3 model performed exceptionally well in terms of processing speed and efficiency, while the ResNet-18 model scored marginally higher in terms of descriptive accuracy. This trade-off emphasizes how crucial it is to choose the right model depending on the demands of a given application, whether accuracy or computational economy is the top priority.

The picture captioning project, which will use ResNet-18 and MobileNetV3, has a lot of room to grow and improve in the future. Integrating attention mechanisms with sophisticated architectures such as Transformers is a potential approach that can enhance the contextual relevance of generated captions by enabling the model to concentrate on particular characteristics of the images. Furthermore, the utilization of pre-trained language models, such BERT or GPT, might improve the semantic comprehension of captions, resulting in more detailed descriptions. The generalization and performance of the model can be further enhanced by growing the dataset to include larger and more varied collections of photos as well as multimodal data that includes metadata and user-generated content. Incorporating user input techniques could also make captioning experiences more personalized by allowing descriptions to be tailored to the tastes of specific users. By adding accessibility features and user interaction, the MobileNetV3 model may be used in mobile and embedded systems by optimizing it for real- time applications. Lastly, investigating cross-modal applications could increase the usefulness of the created models. Examples of these include image retrieval based on text descriptions or video captioning. We may greatly improve the precision, applicability, and effectiveness of picture captioning systems by pursuing these developments, opening the door for creative uses in a variety of fields.

ACKNOWLEDGMENT

Required resources are provided by the Department of CSE(DS), Institute of Aeronautical Engineering, Hyderabad, India for this paper's research study and related work.

REFERENCES

[1]. Hossain, MD Zakir, et al. "A comprehensive survey of deep learning for image captioning." *ACM Computing Surveys (CsUR)* 51.6 (2019): 1-36.
[2]. Ding, Songtao, et al. "Image caption generation with high-level image features." *Pattern Recognition Letters* 123 (2019): 89-95.
[3]. Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
[4]. Xu, K. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.
[5]. Kinghorn, Philip, Li Zhang, and Ling Shao. "A region-based image caption generator with refined descriptions." *Neurocomputing* 272 (2018): 416-424.
[6]. L. Lin, S. Zhong, C. Jia and K. Chen, "Insider Threat Detection Based on Deep Belief Network Feature Representation," 2017 International Conference on Green Informatics (ICGI), Fuzhou, China, 2017.
[7]. Vaswani, A. "Attention is all you need." *Advances in Neural Information Processing Systems* (2017).
[8]. https://kikaben.com/transformers-encoder-decoder/
[9]. https://s3-us-west-2.amazonaws.com/openai-assets/research- covers/language unsupervised/language_understanding_paper.pdf
[10]. https://python.plainenglish.io/image-captioning-with-an- end-to-end-transformer-network-8f39e1438cd4