International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

From Pixels to Words: A Deep Learning Approach to Image Captioning

Isha Panchal[1], Dr. Jalpa Shah[2]

[1]P.G. Student, Department Of Computer Engineering, Silver Oak University, Ahmedabad, India.
[2]Associate Professor, Department Of Computer Engineering, Silver Oak University, Ahmedabad, India.

Abstract:
Image captioning, a crucial task in computer vision and natural language processing (NLP), aims to generate meaningful textual descriptions for images. Traditional models use an encoder-decoder framework, where convolutional neural networks (CNNs) extract image features, and sequence models generate captions. However, conventional CNN-based approaches often lack efficiency in feature extraction. To address this, we propose a novel image captioning model integrating EfficientNetB0 as the feature extractor with a Transformer-based encoder-decoder architecture. The Transformer-Encoder, equipped with Multi-Head Attention, refines image feature representations by capturing both global and local dependencies. The Transformer-Decoder consists of two self-attention layers: Self-Attention_1 focuses on previously generated words, ensuring linguistic coherence, while Self-Attention_2 dynamically attends to the refined image features, enabling the model to emphasize relevant visual details at each decoding step. Additionally, an adaptive attention mechanism further optimizes image feature utilization for caption generation. We evaluate our model on the Flickr 8k dataset, demonstrating superior performance. Our results highlight the effectiveness of combining EfficientNetB0 with a Transformer-based encoder-decoder model, achieving improved caption accuracy while maintaining computational efficiency.

Index Terms: Image Captioning, CNN, EfficientNetB0, Deep Learning, Transformer, Multi-Head Attention, Self-Attention, Feature Extraction, Flickr 8k Dataset

I. INTRODUCTION

Image captioning. Image caption generation involves com- comprehending the visual scenes captured in an image and the subsequent generation of the correct sentences that describe them. Image captioning is a technology that combines computer vision (CV) and natural language processing (NLP) to generate descriptive and contextually relevant textual descriptions for images. [1] The system takes an image as its input. Computer vision techniques are applied to extract meaningful features from the image. These features represent objects, scenes, or other relevant information within the image. NLP models generate descriptive captions based on the extracted image features. The captioning model considers the context of the image to produce coherent and contextually relevant descriptions.[14] This involves understanding relationships between objects, recognizing actions, and interpreting the scene. The generated captions are often evaluated using metrics such as BLEU (Bilingual Evaluation Understudy), ROUGE (Recall-Oriented Understudy for Gisting Evaluation), CIDEr (Consensus-based Image Description Evaluation), METEOR (Metric for Evaluation of Translation with Explicit Ordering) [1].

Deep learning, a transformative branch of artificial intelligence (AI), has revolutionized the way machines perceive, process, and interpret data. Unlike traditional machine learning methods that rely heavily on feature engineering, deep learning models automatically extract hierarchical features from raw data by leveraging multi-layered artificial neural networks. This paradigm shift has led to remarkable advancements in fields such as computer vision, natural language processing (NLP), speech recognition, and autonomous systems.[11]

At its core, deep learning is driven by neural networks with numerous layers, commonly referred to as deep neural networks (DNNs). These networks consist of an input layer, multiple hidden layers, and an output layer.[13] Each layer progressively refines the data representation, allowing the network to learn complex patterns and relationships. The advent of high-performance computing hardware (e.g., GPU and TPU), large-scale datasets, and improved training algorithms has propelled deep learning to the forefront of AI research.

II.NLP TECHNIQUES

Text Preprocessing Text data is often noisy and unstructured. Preprocessing techniques ensure that the input is clean and ready for analysis. Tokenization: Breaking down text into individual words, sentences, or subwords (e.g., "Natural Language Processing" → ["Natural", "Language", "Processing"]).Stopword Removal: Removing common words (e.g., "the," "is") that do not add significant meaning.Stemming and Lemmatization: Reducing words to their root forms (e.g., "running" → "run"). Normalization: Lowercasing, handling punctuations, and correcting misspellings.[17]

Language Understanding Machines extract meaning and context from text through techniques like: Part-of-Speech (POS) Tagging and identifying grammatical roles (e.g., noun, verb).[21] Named Entity Recognition (NER): Identifying entities such as names, dates, and locations.Syntactic Parsing: Analyzing the grammatical structure of sentences.Semantic Analysis: Understanding the meaning of words, phrases, and context.

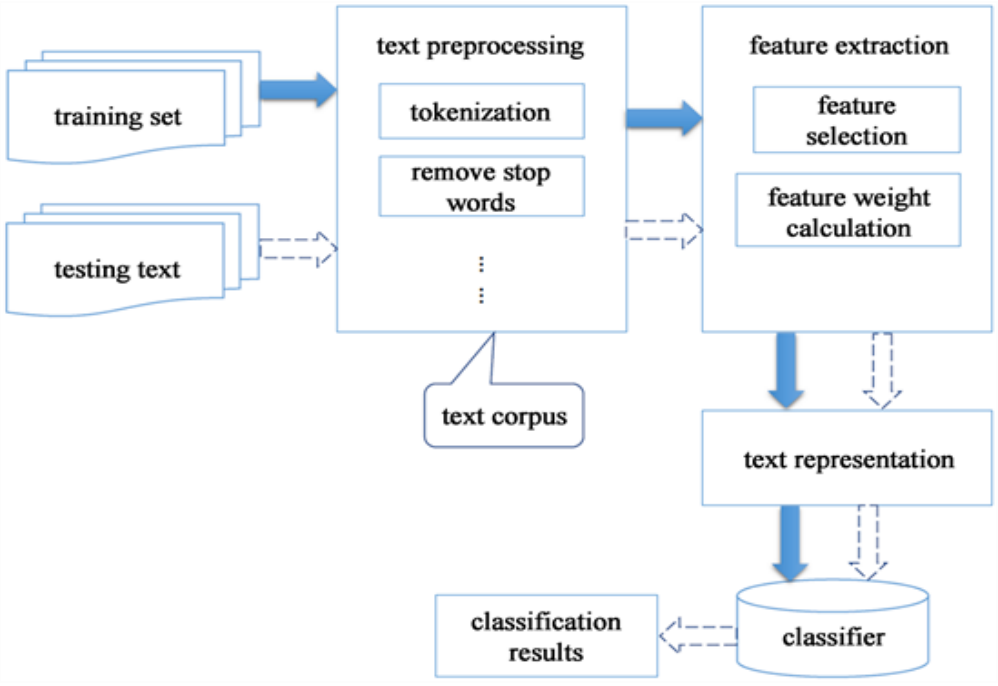


Fig:1 Preprocessing for Text Classification

Language Generation NLP systems generate coherent, contextually accurate text based on given inputs. Text Summarization: Producing concise summaries of long texts. Dialogue Generation: Building chatbots and conversational agents.[20]Text Completion: Predicting the next word or phrase in a sequence. Machine Translation Converting text or speech from one language to another, leveraging models like Transformer-based architectures (e.g., Google Translate)

Image captioning is a multidisciplinary task at the intersection of computer vision and natural language processing (NLP), aimed at generating descriptive textual captions for images. By combining the ability to analyze visual content with language generation models, image captioning enables machines to understand and articulate the semantics of visual data in human language. [2]

Though NLP effectively extracts and analyzes raw text data, such as text-based requirement documents, to assist in design specification, natural language, inherent complexity, and variability pose challenges in accurately interpreting the data.[23] In this paper, we explore the integration of NLP with SysML to automate the generation of system models from input textual requirements.

This figure illustrates the **text classification pipeline**, which consists of multiple stages, including text preprocessing, feature extraction, text representation, and classification. The process begins with input data, which includes a **training set** and **testing text**. The training set consists of labeled textual data used to train the classification model, while the testing text comprises unseen data used to evaluate the model's performance. These datasets undergo **text preprocessing**, a crucial step in Natural Language Processing (NLP) that prepares raw text for analysis. This step includes **tokenization**, which breaks down the text into smaller units such as words or phrases, and **stop-word removal**, which eliminates common but non-informative words like "and," "the," and "is" to enhance processing efficiency. The preprocessed text forms a **text corpus**, which serves as a structured dataset for further analysis. Once preprocessing is complete, the text undergoes **feature extraction**, which is responsible for transforming textual data into numerical representations that can be processed by machine learning algorithms. This stage involves **feature selection**, where the most relevant words or phrases are identified, and **feature weight calculation**, where numerical values are assigned to words based on their importance in the text (e.g., Term Frequency-Inverse Document Frequency (TF-IDF) or word embeddings).

The extracted features are then used to generate a **text representation**, which converts raw text into a structured format suitable for classification.The processed text representation is fed into a **classifier**, which is a machine learning model designed to categorize text into predefined classes. The classifier learns patterns from the training data and applies them to classify new, unseen text. Finally, the system produces **classification results**, which indicate the predicted category of the input text. The classifier can iteratively improve its performance by refining the feature selection process and optimizing its learning algorithm

III. Image Captioning System Using Deep Learning Approaches

Image captioning is a multidisciplinary task that combines techniques from computer vision, deep learning, and natural language processing. It requires models to extract meaningful features from images and then generate grammatically correct and semantically meaningful sentences to describe them. Key components of an image captioning system include an image encoder, a text decoder, and an attention mechanism [12]. By employing an attention mechanism, the model can selectively focus on specific parts of the image, leading to more accurate and contextually relevant captions [3].

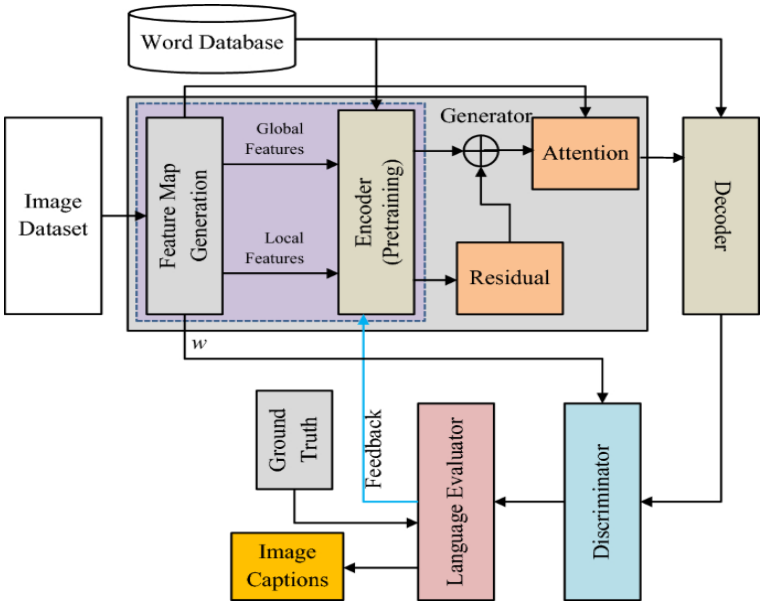


Fig:2 Image Captioning System Using Deep Learning Approaches

Most image captioning systems use an encoder-decoder framework, where an input image is encoded into an intermediate representation of the information in the image, and then decoded into a descriptive text sequence.[19] The most popular benchmarks are Flicker 30k, Flicker 8k, and MS COCO, and models are typically evaluated according to a BLEU or CIDER metric.

Recent advances in image captioning have been driven by the development of deep learning models [18], particularly convolutional neural networks (CNNs) [19] and recurrent neural networks (RNNs) [20]. These models are effective in extracting features from images and generating accurate and fluent captions. Nevertheless, the RNN's challenges in gradient propagation prompted the integration of Long Short-Term Memory (LSTM) [11]. LSTMs were employed to interpret extracted image features, transforming them into captions by generating a cohesive string of words [22].

A significant milestone in the field of image captioning was the incorporation of attention mechanisms into the encoder-decoder architecture [23]. Inspired by their effectiveness in machine translation and object detection, attention models empowered captioning systems to selectively focus on pertinent aspects of input images during caption generation. This breakthrough not only substantially enhanced the quality of generated captions but also became a standard component in numerous state-of-the-art image captioning models [24].

The utility of attention mechanisms transcends various captioning models, as evidenced by their widespread adoption. The advent of attention mechanisms laid the groundwork for the emergence of self-attention, which subsequently facilitated the development of multi-head attention within the Transformer model. The Transformer model revolutionized the abstraction of multi-head self-attention operations into a unified module, enabling the stacking of these modules to attain the requisite non-linearity and representational capacity for modeling intricate functions. Originally applied to machine translation, these advancements were later extended to the realm of image processing. Since its inception in 2017 [8], the Transformer model has emerged as a cornerstone framework underpinning many contemporary state-of-the-art models. The Attention on Attention (AoA) mechanism has shown significant promise in the field of image captioning, contributing to advancements in how visual features are processed and used to generate textual descriptions of images [22].

IV. APPLICATIONS OF IMAGE CAPTIONING

Image captioning makes visual content more accessible to individuals with visual impairments. It provides textual descriptions of images and helps them understand the content. Image captions are used in content-based image retrieval systems, enabling users to find images based on textual descriptions rather than keywords. Image captioning is widely used on social media platforms.[1] It allows users to add context to their photos and helps index and search content. Search engines can use image captions to improve the accuracy of image searches.[5]Users can input text to find images relevant to their query. On platforms where user-generated content is shared, image captions can be analyzed for content moderation, ensuring that inappropriate or harmful content is filtered out. In the medical field, image captioning can be applied to medical imaging to provide detailed descriptions of scans and images for better patient understanding and communication among healthcare professionals. Image captioning is used in e-learning and educational materials, providing explanations for visual content to enhance learning experiences.[1] It aids in computer vision applications, such as self-driving cars, robotics, and drones, by providing descriptions of the environment. Image captioning assists in organizing personal photo collections by automatically generating descriptions for images.

V. PROPOSED WORK

The proposed image captioning system integrates convolutional neural networks (CNNs) with transformers to generate descriptive captions for images. The CNN extracts visual features, which are then processed by a transformer encoder. [10]The transformer decoder utilizes these encoded features, incorporating both visual and positional information, to generate meaningful captions.

This system consists of three key components:

1. **Feature Extraction (CNN)** – An EfficientNetB0 model, pre-trained on ImageNet, extracts essential visual features from images.
2. **Transformer Encoder** – Processes extracted features using multi-head self-attention, layer normalization, and feed-forward networks.
3. **Transformer Decoder** – Generates captions sequentially using self-attention, encoder-decoder attention mechanisms, and positional embeddings.
4.

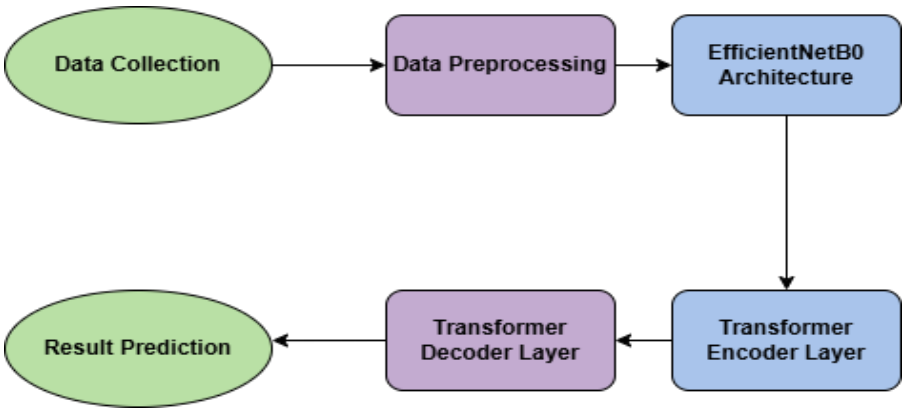


Fig:3 Proposed Work

The entire model is implemented within the ImageCaptioningModel class, which manages training and evaluation processes. It includes a custom training loop to optimize captioning performance across multiple captions per image, incorporating loss calculation and accuracy tracking. [9] By leveraging a

transformer-based architecture, this system provides a flexible and efficient framework for generating descriptive image captions, allowing for experimentation and fine-tuning to enhance performance.

VI. Phases of the Encoder-Decoder Model

To generate a caption for a given image, two key inputs are required: **image data** and **text data**. Previous research has demonstrated that combining CNNs with RNNs or Transformers can significantly enhance image description quality. In our proposed system, we employ an **encoder-decoder architecture** that leverages **EfficientNetB0** for image feature extraction and a **Transformer-based decoder** for generating captions. The model consists of three main phases:[4]

1. **Feature Extraction**
2. **Text Preprocessing**
3. **Language Modeling**

Before feeding image and text data into the model, both undergo pre-processing and transformation. The following sections describe how these steps are implemented in our approach.

Feature Extraction: The image captioning process starts with feature extraction, where the high-dimensional RGB image is transformed into a latent space representation. Initially, the image is converted into a vector format before being fed into the neural network.[7] For this purpose, we utilize **EfficientNetB0**, a pre-trained CNN model, to extract meaningful image features. EfficientNetB0 has demonstrated superior performance with fewer parameters compared to traditional models like VGG-16 or ResNet. The last layer of the CNN is removed, and we extract features from the second-last layer, representing the image in a fixed-size vector format. Feature extraction is the first step in the image captioning pipeline. The goal is to convert high-dimensional image data into a compact and meaningful feature representation that can be processed by the Transformer-based caption generator.

1. Image Normalization & Resizing: The input image is resized to 224×224 pixels (the required input size for EfficientNetB0). Pixel values are normalized to the range [0, 1] for stable training.
2. Feature Extraction using EfficientNetB0: We use EfficientNetB0, a lightweight and powerful pre-trained CNN (trained on ImageNet) for feature extraction. The final classification layer is removed, and features are extracted from the second-last layer. The extracted feature vector is of size 1280 dimensions.
3. Transformation into Latent Space: The extracted feature vector is passed through a fully connected (FC) layer to transform it into a lower-dimensional representation. This latent representation is used as input to the Transformer encoder.

Text Preprocessing: Text preprocessing is essential to ensure the raw captions are in a format suitable for training the model. It involves cleaning, tokenizing, and encoding captions into a numerical form.[5] Text preprocessing is crucial for generating high-quality captions. Before feeding text into the model, we perform the following steps:

1. Data Cleaning: Convert all text to lowercase to maintain consistency. Remove punctuation, special characters, and numeric values. Eliminate unnecessary words that do not contribute meaningfully to captions. Example: Original Caption: "A Big Dog is Running! 123" Processed Caption: "A big dog is running"
2. Tokenization & Vocabulary Creation: The dataset is tokenized into individual words. A word-to-index dictionary is created, where each unique word gets an integer ID. A maximum sequence length is determined based on the longest caption in the dataset.

3. Adding Start & End Token : Each caption is enclosed within <start> and <end> tokens to indicate sentence boundaries.Example: Before: "a dog is running in the snow" After: "<start> a dog is running in the snow <end>"
4. Padding & Sequence Encoding: Captions are converted into sequences of integers using the word-to-index mapping. Shorter sequences are padded with zeros to match the maximum length.

Language Modeling with Transformers: In the final phase, preprocessed text and extracted image features are fed into the captioning model.[8] Unlike conventional models using LSTMs or GRUs, our system leverages a Transformer-based decoder, which offers several advantages: In this phase, both image features (from EfficientNetB0) and text sequences (preprocessed captions) are used to train a Transformer-based decoder to generate captions.[6]

1. Encoding Image Features: The 1280-dimensional image feature vector extracted by EfficientNetB0 is passed through the Transformer encoder. The encoder applies multi-head self-attention and feed-forward layers to refine these features.
2. Generating Captions with a Transformer Decoder: The Transformer decoder receives two inputs: [11] Image features from the encoder and tokenized text (caption words as input sequences). The decoder uses multi-head attention to focus on relevant visual features while generating each word. A positional encoding mechanism helps retain word order. The decoder predicts words one at a time, conditioning each prediction on previously generated words.
3. Caption Generation using Beam Search: Instead of choosing the most likely word at each step (greedy decoding), we use beam search to explore multiple possibilities.[6] Beam search keeps track of the top k best sequences, reducing errors and improving fluency.
4. Final Caption Output: The sequence of predicted words is converted back into natural language.[3] The <start> and <e*nd*> tokens are removed Example: Predicted Output: [1, 2, 3, 4, 5, 6, 7, 8, 9], Decoded Caption: "A dog is running in the snow"

VII. METHODOLOGY

Convolutional Neural Networks: Famous artificial neural network architectures for object and image recognition and classification include convolutional neural networks.[21] To recognize text from an image document as input and provide the user with editable text. Deep learning thus makes use of CNN to categorize the objects in an image. Convolutional Neural Networks (CNNs) can analyze complex objects and patterns because they have an input layer, an output layer, numerous hidden layers, and thousands of parameters. It sub-samples the input using convolution and pooling techniques before using an activation function. The performance of the CNN model is evaluated by recognizing the image.[12] The input image's size and the output shape are similar. Horizontal or diagonal edges are fundamental features that the first layer typically extracts. Machine recognition of handwritten texts has been in research for pattern recognition. The following layer receives this output and searches for more difficult features like corners and multiple edges. Figure 3 depicts the architecture of CNN. [14] The network is also capable of recognizing increasingly complex elements, along with objects, etc. which consist of six convolutional layers in sequential order with ReLU serving as the activation function and the maxpool layer following, which are used in the research.

It has been convincingly shown that CNNs can produce a rich representation of the input image by embedding it into a fixed-length vector, such that this representation can be used for a variety of vision tasks like object recognition, detection, and segmentation [8]. Hence, image captioning methods based on

encoder-decoder frameworks often use a CNN as an image encoder. The RNN network obtains historical information through continuous circulation of the hidden layer, which has better training capabilities and can perform better than mining deeper linguistic knowledge such as semantics and syntax information implicit in the word sequence [19]. For a dependency relationship between different location words in historical information, a recurrent neural network can be easily represented in the hidden layer state. In image captioning task based on encoder-decoder framework, the encoder part is a CNN model for extracting image features It can use models such as AlexNet, VGG (VGG-16, VGG-19), GoogleNet (Inception v1), InceptionNet (v2, v3, v4, Inception-ResNet), ResNet (ResNet-50, ResNet-101, ResNet-152, ResNeXt), DenseNet (121, 169, 201, 264), Xception, MobileNet (V1, V2, V3), EfficientNet (B0–B7, V2-S/M/L), and NASNet (Mobile, Large) for feature extraction. These pre-trained models, optimized for accuracy and efficiency, play a crucial role in deep learning-based image captioning by extracting meaningful visual features from images.[23] for feature extraction in image captioning tasks.

These CNN models, pre-trained on large-scale datasets like ImageNet, offer different trade-offs between accuracy, efficiency, and computational cost. AlexNet was one of the first deep CNN architectures, while VGG introduced deeper networks with uniform 3×3 convolutions. Inception-based models enhanced efficiency using inception modules, with Inception-ResNet integrating residual learning.[25] ResNet revolutionized deep networks with residual connections to tackle vanishing gradients. DenseNet improved gradient flow through dense connectivity. [19] Xception extended InceptionNet by utilizing depthwise separable convolutions. MobileNet introduced lightweight models optimized for mobile and edge devices.[22] EfficientNet used compound scaling to optimize accuracy and efficiency, with EfficientNetV2 improving speed and training efficiency. NASNet leveraged neural architecture search (NAS) to automatically discover high-performing CNN structures. These models provide various options for achieving high-performance feature extraction in deep learning-based image captioning systems.[9]

Image captioning is a challenging task that combines computer vision and natural language processing to generate meaningful textual descriptions of images. Our approach utilizes a deep learning-based encoder-decoder framework, where EfficientNetB0 serves as the feature extractor (encoder) and a Transformer-based model functions as the decoder for caption generation. This combination enables computational efficiency while maintaining high accuracy by leveraging deep feature extraction and self-attention mechanisms for language modeling[20]. The encoder is responsible for extracting high-level visual representations from images. We employ EfficientNetB0, a lightweight and efficient convolutional neural network (CNN) pre-trained on ImageNet. Unlike traditional CNN architectures such as ResNet152, EfficientNetB0 achieves higher accuracy with fewer parameters, making it an ideal choice for feature extraction in image captioning. Before being passed to the encoder, images are resized to (3, 224, 224) and normalized to [0, 1] to ensure consistency in data representation. To optimize storage and retrieval, preprocessed images are stored in an HDF5 (H5) file, reducing disk I/O overhead during training. During processing, the final classification layer of EfficientNetB0 is removed, and the extracted features are taken from the second-last layer, producing a (7 × 7 × 1280) feature map. This output retains spatial information through the 7 × 7 grid, while the 1280 feature channels capture deep semantic representations of the image.0[9]

To convert these extracted visual features into a sequential format suitable for text generation, a Transformer-Encoder is used. The Transformer-Encoder refines the feature representations by leveraging self-attention mechanisms, which allow the model to focus on different parts of the image dynamically. This ensures that important visual elements contribute effectively to the final caption. The Transformer-

Decoder then generates textual descriptions by attending to both the refined image features and previously generated words in a sequential manner.[15] Instead of relying on recurrent structures like LSTMs, the Transformer-Decoder processes information in parallel, leading to improved learning efficiency and better contextual understanding. Additionally, text processing is handled using TransformerEncoderBlock and Positional Embedding layers, which help encode textual information and establish semantic relationships between the image features and the generated text. These layers enhance the model's ability to understand and integrate spatial and contextual dependencies, resulting in coherent and contextually accurate captions. By combining EfficientNetB0 for feature extraction and a Transformer-based architecture for language generation, our model provides a highly accurate, efficient, and scalable solution for automatic image captioning. This system is particularly useful for applications such as assisting visually impaired individuals, content-based image retrieval, and automatic image annotation in large-scale multimedia databases.[11]

EfficientNetB0: We have developed a deep learning model for image captioning based on an encoder-decoder framework using EfficientNetB0 as the feature extractor and a Transformer-based architecture for caption generation. [16] EfficientNetB0 is a lightweight and highly efficient convolutional neural network (CNN) that serves as an advanced feature extractor for image captioning tasks. As part of the EfficientNet family, EfficientNetB0 was introduced by Tan and Le (2019) and is designed using Neural Architecture Search (NAS) to optimize accuracy and computational efficiency. [7]Unlike traditional CNNs such as ResNet or VGG, which scale arbitrarily by increasing depth or width, EfficientNet employs compound scaling to proportionally balance depth (number of layers), width (number of channels per layer), and resolution (input image size). This approach significantly enhances feature extraction while minimizing computational costs, making EfficientNetB0 an ideal choice for image captioning systems.

In the context of image captioning, EfficientNetB0 is used as the encoder in an encoder-decoder framework, where it extracts deep semantic features from input images before passing them to a Transformer-based decoder for caption generation.[17] Pre-trained on ImageNet, EfficientNetB0 effectively captures high-level representations, such as object details, spatial relationships, and contextual information, which are crucial for generating meaningful captions. The final classification layer of EfficientNetB0 is removed to obtain a ($7 \times 7 \times 1280$) feature map, where the 7×7 spatial grid preserves positional information, and the 1280 channels encode deep semantic features. These extracted features are then flattened and transformed into sequential embeddings, which serve as input tokens for the Transformer-Encoder, refining them using self-attention mechanisms.

By integrating EfficientNetB0 as the feature extractor, image captioning models achieve superior performance in terms of accuracy, efficiency, and generalization. Compared to traditional CNNs like ResNet152, EfficientNetB0 offers higher accuracy with fewer parameters, making it well-suited for resource-constrained environments such as mobile and embedded systems. This combination of EfficientNetB0 with a Transformer-based decoder ensures that generated captions are coherent, contextually accurate, and semantically rich, improving applications in assistive technology, image retrieval, and automatic annotation systems.[10]

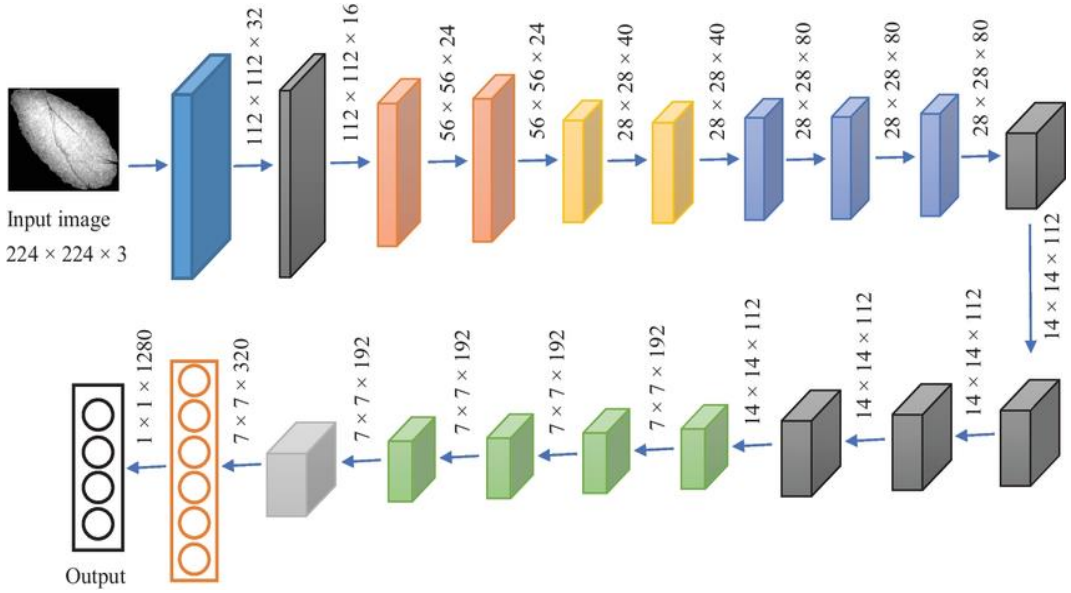


Fig: 4 The architecture of the EfficientNetB0 Model

Transformer Encoder-Decoder: The Transformer architecture serves as the backbone for sequence-to-sequence tasks in image captioning. It comprises two primary modules: the Transformer-Encoder and the Transformer-Decoder. The encoder processes extracted image features, while the decoder generates textual captions based on these features. This architecture efficiently models dependencies within input data, ensuring accurate and meaningful caption generation.

The Transformer-Encoder, which is the second part of our encoder, plays a crucial role in refining the image features extracted from EfficientNetB0. It consists of multiple layers, where each layer contains Multi-Head Self-Attention and Feed-Forward Networks (FNN). The Multi-Head Self-Attention mechanism allows the model to focus on different parts of the image simultaneously, enabling a more comprehensive understanding of the visual context.[10] The Feed-Forward Networks (FNN) process the outputs of the self-attention layers, further refining the feature representations to make them more meaningful for caption generation. Since the Transformer architecture does not inherently recognize the order of input sequences, positional encoding is applied to the extracted feature vectors to retain spatial relationships. Each of the 49 feature vectors (corresponding to the 7 × 7 spatial locations in the feature map) is assigned a positional encoding vector to encode location-based information. We use sine functions for even indices and cosine functions for odd indices to generate these positional encodings.[19] This process ensures that the Transformer-Encoder captures not only the extracted visual features but also their spatial arrangement, ultimately improving the contextual understanding of the image before passing it to the Transformer-Decoder for caption generation.

The self-attention mechanism in the encoder computes attention scores by calculating the dot-product between the query (Q), key (K), and value (V) matrices.[18] The scaled dot-product attention mechanism is applied as follows: where is the dimension of the key vectors This formulation ensures that attention weights remain numerically stable and do not grow excessively large.[16] To enhance feature representations, the encoder also utilizes layer normalization, which standardizes input distributions and improves training stability. Dropout regularization is applied to prevent overfitting and enhance model generalization

The Transformer-Decoder serves as the final stage in the captioning process, generating text descriptions based on the encoded image representations. It is composed of multiple layers, each containing Masked Multi-Head Self-Attention, Adaptive Multi-Head Attention, and Feed-Forward Networks (FNN).[20] The Masked Multi-Head Self-Attention mechanism ensures that the model only attends to previous words when predicting the next word in the caption, preserving the autoregressive nature of the decoding process. The Adaptive Multi-Head Attention mechanism enables the decoder to selectively focus on the most relevant regions of the image when generating each word, ensuring that the generated caption is semantically aligned with the visual content. The Feed-Forward Networks (FNN) process the attention outputs to further refine the textual representations before passing them to the next layer.[23]

For word embeddings, we use pre-trained word vectors such as GloVe, which convert words into meaningful numerical representations, improving the model's ability to generate coherent and contextually relevant captions. Similar to the transformer encoder, positional encoding is applied to the text inputs in the decoder to preserve sentence structure and maintain word order.[21] Caption generation in the Transformer-Decoder relies on computing correlations between the query (Q), key (K), and value (V) vectors. The dot-product operation between Q and K determines how much attention each word should pay to different parts of the image, and the result is scaled to maintain numerical stability. The softmax function is then applied to compute the probability distribution over different regions of the image, and the weighted sum with V is performed to obtain the final attended feature representation.[9]

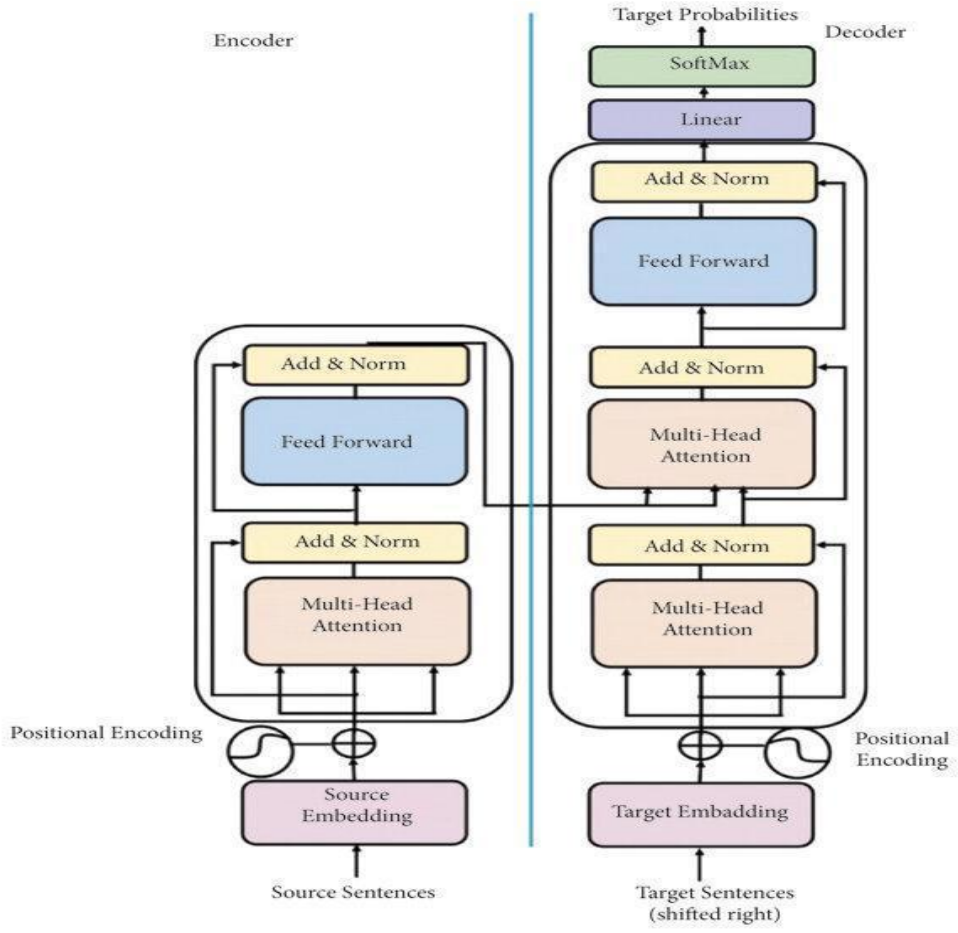


Fig: 5 Transformer Encoder-Decoder

The decoder in a Transformer-based image captioning model integrates two primary attention mechanisms: self-attention and adaptive attention. The self-attention mechanism analyzes relationships between words, ensuring coherence in sentence formation.[15] Its masked nature ensures that only previously generated words influence the next word prediction. The adaptive attention mechanism dynamically determines when and where to utilize image features, incorporating an adaptive gating mechanism that selectively enables or disables the use of image features based on contextual needs. The gate activates when predicting content-related words and restricts visual influence for non-visual words, ensuring linguistic accuracy. Additionally, an adaptive gating function modulates the influence of visual features in the decoder. At each time step, this function is computed using the hidden state of the decoder, the attended image feature, and learnable parameters, selectively controlling the contribution of visual information to the word-generation process.

[10] Multi-Head Self-Attention plays a fundamental role in both the encoder and decoder blocks of the Transformer. It allows the model to assign different levels of importance to various elements within the input sequence, enabling it to capture dependencies among diverse elements within an image or sentence, attend to multiple positions simultaneously, and improve contextual understanding. This mechanism significantly enhances the accuracy and relevance of generated captions by allowing the model to focus on pertinent visual and textual details.

Layer normalization and Feed-Forward Networks (FNN) are integral components of both encoder and decoder blocks, contributing to improved learning efficiency and stability.[15] Layer normalization standardizes the input to each layer, stabilizing the training process by reducing internal covariate shifts and preventing drastic fluctuations in feature activations, leading to faster convergence and improved learning performance. FNNs introduce non-linearities that allow the model to capture intricate patterns and relationships in data, enhancing learned representations and making them more expressive and capable of extracting meaningful insights from input data.

Once the Transformer-Decoder processes all attention layers, the output is passed through a fully connected layer with a softmax activation function to compute the probability distribution over the vocabulary. At each time step, the word with the highest probability is selected, constructing the caption word by word until an end token is generated, signaling completion. The final probability distribution over the vocabulary at each step is computed using learnable parameters, ensuring accurate and contextually relevant caption generation.[16]

VIII. Evaluation Metrics

BLEU (Bilingual Evaluation Understudy): is a metric used to assess the similarity between the generated caption and reference captions by calculating the n-gram precision.[25] It quantifies how closely the generated caption aligns with the reference captions.BLEU is a precision-based metric that measures how many words (or n-grams) in the generated caption match the reference captions. It is calculated using **n-grams**, where BLEU-1 considers unigrams (single words), BLEU-2 considers bigrams (two-word sequences), and so on.

Interpretation of BLEU scores:

1. BLEU-1: Measures the match of single words (unigrams). It indicates basic word accuracy.
2. BLEU-2: Measures two-word sequences (bigrams), capturing some contextual accuracy.
3. BLEU-3 & BLEU-4: Consider longer n-grams, reflecting fluency and context.
4. Higher BLEU scores mean better alignment with human-generated captions.

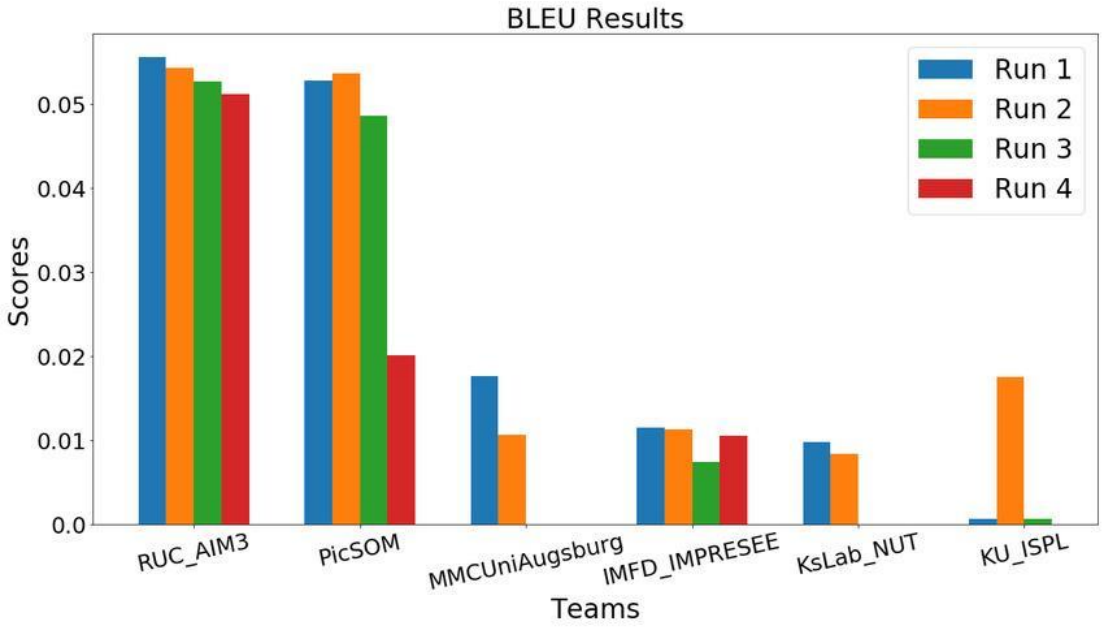


Fig: 6 BLEU Metrics

METEOR: (Metric for Evaluation of Translation with Explicit ORdering): METEOR is another metric based on n-gram precision but also considers recall and aligns words based on their meanings. It accounts for synonyms and paraphrases, making it suitable for evaluating diverse captions .[10] Unlike BLEU, METEOR considers **synonyms, stemming, and word order** for a better semantic match. It uses a harmonic mean of **precision and recall**, giving more weight to recall than BLEU. It captures meaning better than BLEU since it accounts for synonyms. Higher METEOR values indicate better-quality captions

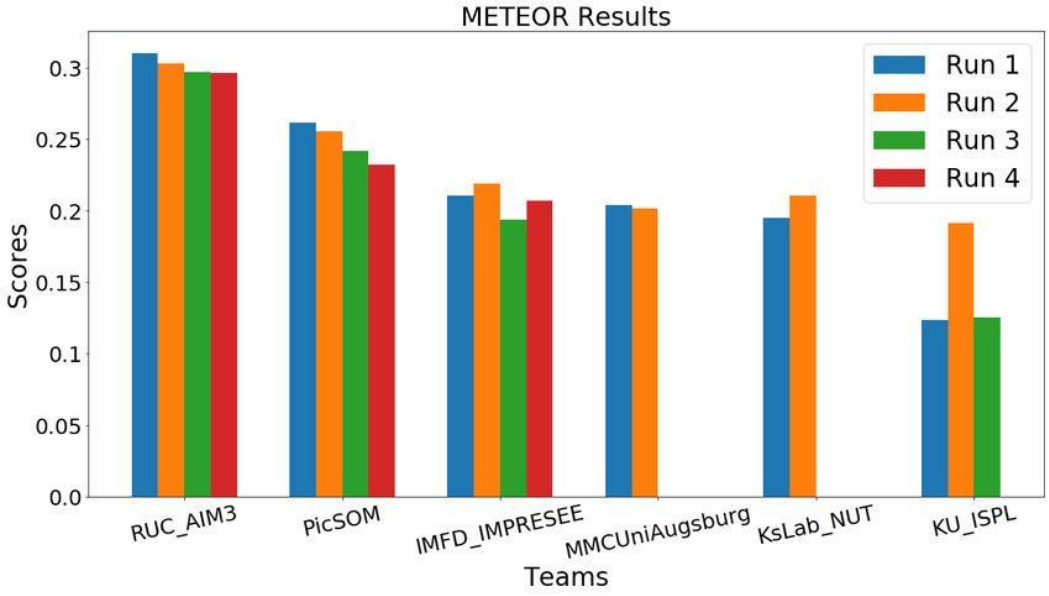


Fig: 7 METEOR Metrics

SPICE (Semantic Propositional Image Caption Evaluation) : is an advanced metric used to assess the quality of image captions by focusing on their semantic content rather than just word overlap. Unlike BLEU, which measures n-gram precision, SPICE evaluates how well a generated caption represents the

relationships between objects in an image. It does this by converting both reference and generated captions into scene graphs—structured representations containing objects, attributes, and relationships. For example, for an image of a dog sitting on a red couch, the scene graph might include elements like {dog, couch, red, sitting-on(dog, couch)}. SPICE then calculates precision, recall, and F1-score by comparing the generated scene graph with the reference scene graph, ensuring that the model captures meaningful relationships rather than just matching words. This makes SPICE particularly useful as it aligns closely with human judgment, handling paraphrasing and synonyms more effectively than traditional metrics. However, it comes with some limitations, such as being computationally expensive and less sensitive to fluency and grammar. Despite these drawbacks, SPICE remains a valuable evaluation tool, especially when assessing how well a model understands the semantic structure of an image.

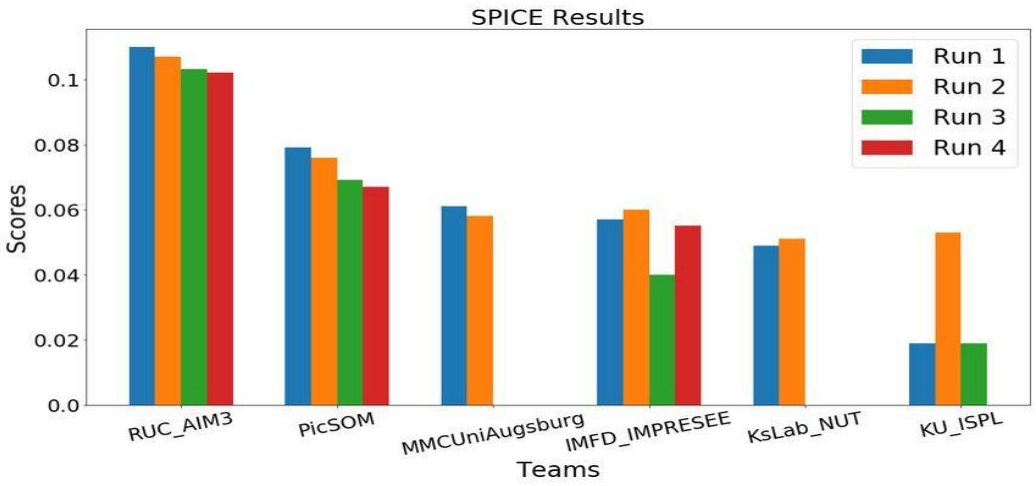


Fig: 8 SPICE Metrics

CIDEr: (Consensus-based Image Description Evaluation): [16] calculates the agreement between the generated caption and reference captions. It evaluates the similarity of the generated caption with all the reference captions, taking into account the diversity of human-provided captions.CIDEr evaluates how well a generated caption aligns with multiple reference captions by computing TF-IDF (Term Frequency-Inverse Document Frequency) weights. It emphasizes words that are important and distinctive for an image. It ensures captions capture relevant details of the image. Higher CIDEr scores indicate more meaningful and accurate captions.

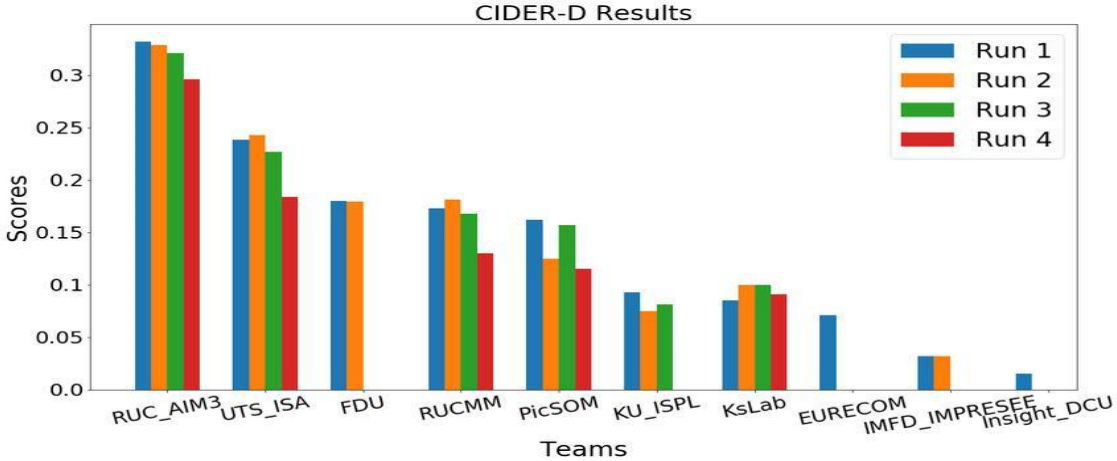


Fig: 9 CIDER-D Metrics

IX. CAPTIONS GENERATIONS RESULT

The evaluation of image captioning models using the **Flickr 8k dataset** involves analyzing the generated captions' accuracy, relevance, and fluency compared to human-annotated ground truth captions. The **Transformer-based Encoder-Decoder model** effectively utilizes self-attention and cross-attention mechanisms to generate contextually accurate and semantically rich captions. The results of caption generation can be assessed using both qualitative and quantitative metrics.

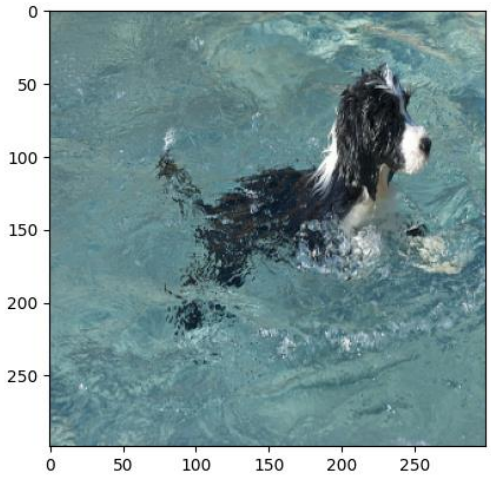
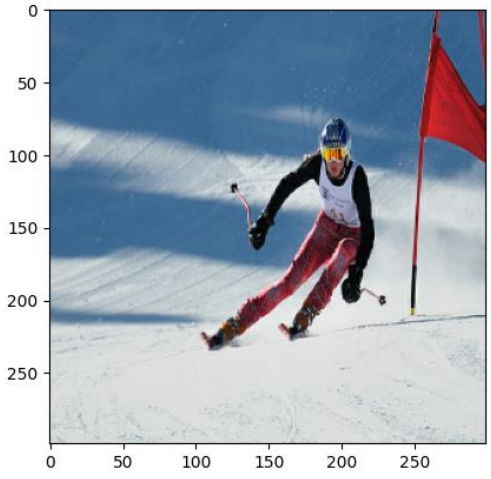
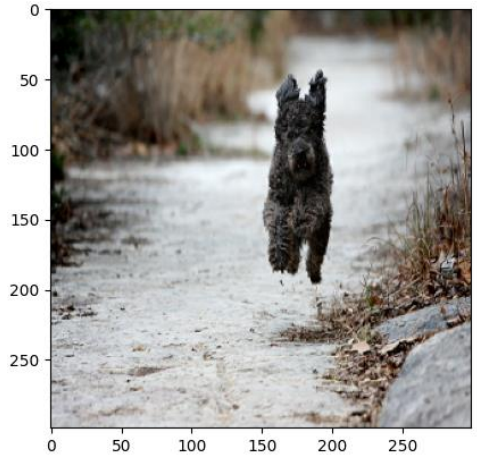
	
Predicted Caption: a black and white dog is swimming in the water	**Predicted Caption: a man wearing a helmet riding a bicycle**
	
Predicted Caption: a snowboarder is riding a ski lift	**Predicted Caption: a black dog is running through the snow**

Fig:10 Captions Generation Results

For quantitative evaluation, automated metrics compare generated captions against ground truth captions. BLEU measures n-gram overlap for word accuracy, while METEOR considers synonyms, stemming, and

word order for a more human-like evaluation. ROUGE evaluates recall-based n-gram matches, CIDEr focuses on sentence-level similarity, emphasizing meaningful words, and SPICE analyzes scene graph structures, considering object attributes and relationships. Leveraging self-attention and cross-attention mechanisms, the Transformer-based model enhances visual-text alignment, resulting in higher CIDEr and SPICE scores than RNN-based models.[17]In qualitative analysis, the generated captions are visually inspected to determine their semantic correctness and contextual relevance to the given images. A high-quality caption should correctly identify the main objects and actions in the image, maintain grammatical accuracy and natural sentence flow, avoid hallucinations (incorrect object or scene descriptions), and be concise yet informative, similar to human annotations. The Transformer model, with its multi-head self-attention and cross-attention mechanisms.

X. Experimental Results & Analysis

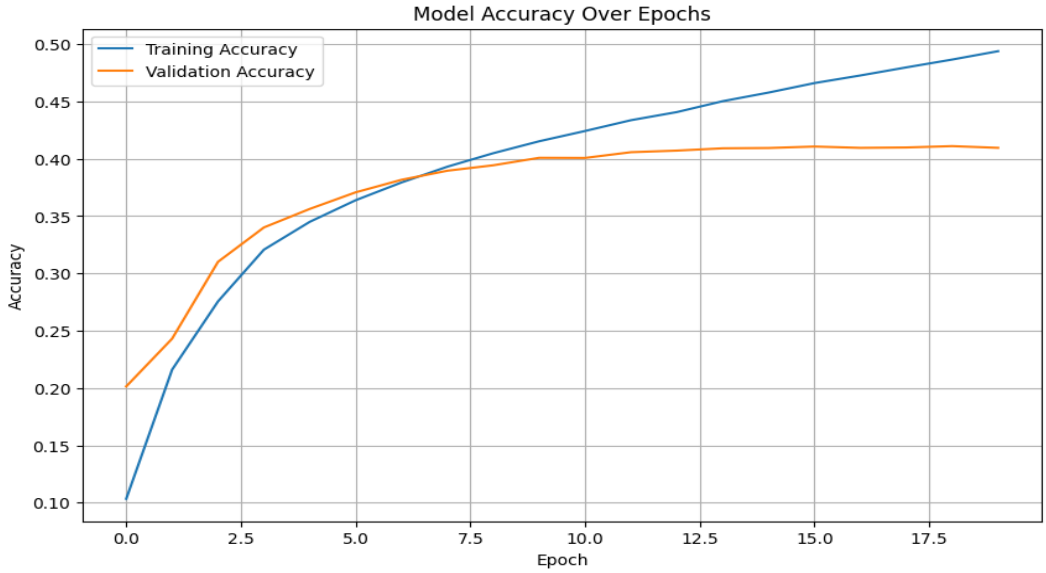


Fig:11 Model Accuracy Over Epochs

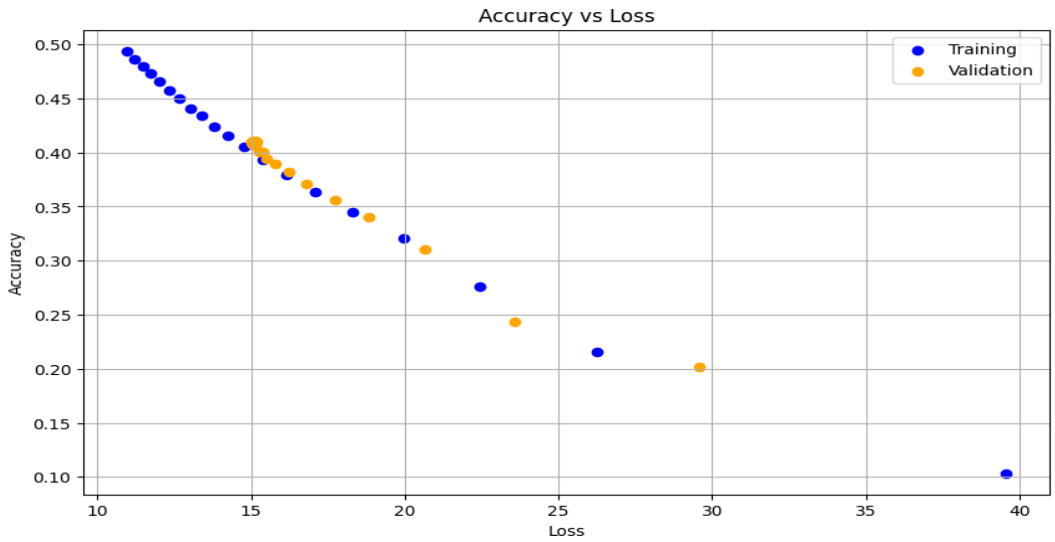


Fig:12 Accuracy vs Loss

XI. Major Challenges

The research paper "From Pixels to Words: A Deep Learning Approach to Image Captioning" highlights several challenges in developing an efficient and accurate image captioning system. One major challenge lies in feature extraction efficiency, as traditional CNN-based models often struggle to capture both global and local dependencies in images. Although EfficientNetB0 is used to address this issue, optimizing feature extraction for better caption accuracy remains a complex task. Another significant challenge is the complexity of natural language processing (NLP) since generating contextually relevant and grammatically coherent captions requires advanced NLP techniques. The variability in natural language descriptions further complicates the process, making it difficult to maintain consistency across different captions. Additionally, training Transformer-based models present computational challenges, as these architectures require large-scale data, high computational power, and fine-tuning of self-attention mechanisms to achieve optimal performance.

Moreover, handling ambiguity in image descriptions is a persistent issue, as objects and scenes in images can have multiple interpretations, leading to inconsistencies in generated captions. The model needs to ensure linguistic coherence while avoiding hallucinations, where incorrect objects or actions are described. Another key difficulty is related to evaluation metrics and subjectivity. While automated metrics like BLEU, ROUGE, CIDEr, and METEOR are commonly used to assess caption quality, they may not fully capture human-like understanding and contextual relevance, necessitating subjective evaluation for real-world applications. Lastly, there is a trade-off between computational efficiency and accuracy. Transformer-based models, while highly effective, are computationally expensive, making it challenging to balance processing power with the quality of generated captions. Addressing these challenges is crucial for improving image captioning models and making them more practical for real-world applications.

XII. Conclusion and Future Scope

Image captioning is a challenging yet crucial task at the intersection of computer vision and natural language processing, enabling machines to describe visual content in human language. In this research, we introduced an EfficientNetB0 and Transformer-based image captioning model, leveraging self-attention mechanisms and adaptive feature utilization to generate accurate and semantically rich descriptions. The model was evaluated on the Flickr 8k dataset, demonstrating improved caption accuracy compared to traditional CNN-RNN architectures. Our findings highlight the effectiveness of combining lightweight CNN feature extraction with a Transformer-based encoder-decoder, ensuring a balance between computational efficiency and linguistic coherence. Through both quantitative (BLEU, METEOR, CIDEr, ROUGE) and qualitative evaluations, the proposed approach exhibited enhanced performance in terms of fluency, contextual relevance, and visual-text alignment. Despite these advancements, several challenges remain, including the high computational cost associated with Transformer-based architectures, the difficulty in capturing complex scene semantics beyond object detection, and the limitations of current evaluation metrics, which may not fully reflect human perception. Future research should explore multimodal fusion techniques, low-complexity transformer variants, and human-in-the-loop evaluation strategies to further refine image captioning performance. Additionally, integrating domain-specific datasets and real-world applications, such as assistive technology, autonomous driving, and medical imaging, could significantly expand the impact of automated image captioning. In conclusion, this study contributes to the ongoing evolution of deep learning-based image captioning by demonstrating the potential of combining EfficientNetB0 with Transformer-based attention mechanisms. The proposed

model sets a foundation for further research into scalable, efficient, and context-aware captioning systems, bringing us closer to bridging the gap between visual perception and natural language understanding in artificial intelligence.

While the proposed image captioning model demonstrates significant improvements in caption accuracy and efficiency by leveraging EfficientNetB0 and a Transformer-based encoder-decoder, several areas remain open for future exploration and enhancement.

1. Enhanced Feature Extraction Techniques: Although EfficientNetB0 provides a balance between accuracy and computational efficiency, exploring other vision models, such as EfficientNetV2, Vision Transformers (ViTs), or Swin Transformers, could further enhance feature extraction capabilities. Combining self-supervised learning (SSL) methods for feature extraction might improve performance on limited datasets without requiring extensive labeled data. Exploring hybrid architectures that integrate convolutional and transformer-based models (e.g., ConvNext) could optimize feature extraction for better contextual understanding.

2. Multimodal Learning for Richer Contextual Understanding: Incorporating scene graphs or object detection models (e.g., Faster R-CNN, YOLO) could provide richer semantic understanding by identifying objects, relationships, and spatial arrangements in images. Leveraging multi-modal transformers such as CLIP (Contrastive Language-Image Pretraining) or BLIP (Bootstrapped Language-Image Pretraining) could enhance the model's ability to align textual and visual features more effectively. Investigating the use of audio-visual captioning by integrating speech or background sounds into the captioning process could make descriptions more comprehensive.

3. Optimizing Computational Efficiency for Real-Time Applications: Knowledge distillation or pruning techniques can reduce the number of parameters in the Transformer model while maintaining accuracy, making the model lighter. Investigating low-rank approximations and quantization methods can further optimize the memory footprint and computational efficiency, allowing deployment on edge devices or mobile applications. Exploring transformer alternatives like Linformer or Performer could reduce the quadratic complexity of self-attention, making the model scalable for real-time applications.

4. Improving Caption Diversity and Semantic Accuracy: Current models sometimes generate repetitive or overly generic captions. Future work could focus on diversity-promoting techniques, such as contrastive learning or reinforcement learning-based reward functions. Leveraging commonsense reasoning models, such as COMET (Commonsense Transformers), could help generate captions that reflect real-world knowledge rather than relying solely on visual patterns. Fine-tuning the language decoder with large-scale pre-trained models, such as GPT-based architectures, could improve grammatical fluency and contextual relevance.

5. Expanding Training and Evaluation on Larger and More Diverse Datasets: While the model performs well on the Flickr 8k dataset, future experiments could extend training to larger datasets such as MS COCO, Conceptual Captions, or LAION-5B to enhance generalization. Investigating domain-specific datasets (e.g., medical imaging datasets, satellite images, or industrial inspection datasets) can tailor the model for specialized applications. Conducting cross-dataset generalization studies will help assess the robustness of the model across different image domains and captioning styles.

6. Refining Evaluation Metrics with Human-Centric Assessment: Existing evaluation metrics (BLEU, CIDEr, METEOR) rely on n-gram overlap, which may not fully capture human-level caption quality. Future research can integrate: Semantic similarity metrics (e.g., BERTScore) to evaluate meaning rather than exact word matches. Human-in-the-loop evaluations using crowdsourced assessments to measure

fluency, coherence, and relevance. Explainability and interpretability tools, such as attention visualization techniques, to understand how the model attends to image regions when generating captions.

7. Domain-Specific Adaptations and Real-World Applications: Adapting the model for assistive technologies for visually impaired individuals, where captions can be integrated into screen readers. Extending image captioning capabilities for autonomous vehicles to describe real-time driving scenes and enhance situational awareness. Implementing personalized caption generation by incorporating user preferences, sentiment analysis, or contextual metadata from previous interactions. Applying image captioning in healthcare by generating detailed captions for radiology images, ultrasound scans, or MRI reports, aiding doctors in diagnostics.

XIII. REFERENCES

1. Hadi, M., Safder, I., Waheed, H., Zaman, F., Aljohani, N. R., Nawaz, R., & Hassan, S. U. (2024). A transformer-based Urdu image caption generation. *Journal of Ambient Intelligence and Humanized Computing, 15*, 3441–3457. https://doi.org/10.1007/s12652-024-04824-9
2. Budhathoki, R., & Timilsina, S. (2023). Image Captioning in Nepali Using CNN and Transformer Decoder. *Journal of Engineering and Sciences, 2*(1), 41–50
3. Mulyawan, R., Sunyoto, A., & Muhammad, A. H. (2023). Pre-trained CNN Architecture Analysis for Transformer-Based Indonesian Image Caption Generation Model. *International Journal on Informatics Visualization, 7*(2), 487–493.
4. Zhang, X., Xuan, J., Yao, C., Gao, Q., Wang, L., Jin, X., & Li, S. (2022). A deep learning approach for orphan gene identification in moso bamboo (Phyllostachys edulis) based on the CNN + Transformer model. *BMC Bioinformatics, 23*, 162. https://doi.org/10.1186/s12859-022-04702-1
5. Mulimani, D., Patil, P., & Chaklabbi, N. (2023). Image Captioning using CNN and Attention-Based Transformer. In S. J. Nanda & R. P. Yadav (Eds.), *Data Science and Intelligent Computing Techniques* (pp. 157–166). Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-14
6. Haidarh, M., & Zhang, S. (2024). Image Captioning Based on Convolutional Neural Network and Transformer. *Conference Paper*, Taiz University & Northeastern University. Retrieved from ResearchGate
7. Dong, X., Long, C., Xu, W., & Xiao, C. (2021). Dual Graph Convolutional Networks with Transformer and Curriculum Learning for Image Captioning. *arXiv preprint arXiv:2108.02366*.
8. Liu, W., Chen, S., Guo, L., Zhu, X., & Liu, J. (2021). CPTR: Full Transformer Network for Image Captioning. *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.*
9. Wang, Y., Xu, J., & Sun, Y. (2022). End-to-End Transformer Based Model for Image Captioning. *Proceedings of the AAAI Conference on Artificial Intelligence.* Raju, K. L., Rayidu, V., Surendra, P., & Satish, V. S. (2024). Image Captioning using CNN and Transformers. *International Journal of Advanced Research in Computer and Communication Engineering, 13*(4), 476–482. https://doi.org/10.17148/IJARCCE.2024.13469
10. Ayush Kumar Poddar, Rajneesh Rani. Hybrid Architecture using CNN and LSTM for Image Captioning in Hindi Language. Procedia Computer Science, Vol. 218, 2023, pp. 686–696. DOI: 10.1016/j.procs.2023.01.049

11. M. Sinthujaa, Chirag Ganesh Padubidri, Gaddam Sai Jayachandra, Mudduluru Charan Teja, Golthi Sai Pavan Kumar. Extraction of Text from Images Using Deep Learning. Procedia Computer Science, Vol. 235, 2024, pp. 789–798. DOI: 10.1016/j.procs.2024.04.075
12. Md Rakibul Hasan. Transformer and Convolutional Neural Network Hybrid Approaches in Medical Image Classification, Caption Generation, and Retrieval Processes. Master's Thesis, Morgan State University, May 2024
13. Wei Liu, Sihan Chen, Longteng Guo, Xinxin Zhu, Jing Liu. CPTR: Full Transformer Network for Image Captioning. arXiv preprint, 2021
14. Taraneh Gandhi, Hamidreza Pourreza, Hamidreza Mahyar. Deep Learning Approaches on Image Captioning: A Review. arXiv preprint, 2023.
15. Huda A. Al-Muzaini, Tasniem N. Al-Yahya, Hafida Benhidour. Automatic Arabic Image Captioning Using RNN-LSTM-Based Language Model and CNN. International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 9, No. 6, 2018, pp. 67–77. DOI: 10.14569/IJACSA.2018.090611
16. Azhar Jamil, Saif-Ur-Rehman, Khalid Mahmood, Monica Gracia Villar, Thomas Prola, Isabel De La Torre Diez, Md Abdus Samad, Imran Ashraf. Deep Learning Approaches for Image Captioning: Opportunities, Challenges, and Future Potential. IEEE Access, 2024. DOI: 10.1109/ACCESS.2024.3365528
17. Asmaa A. E. Osman, Mohamed A. Wahby Shalaby, Mona M. Soliman, Khaled M. Elsayed. Novel Concept-Based Image Captioning Models Using LSTM and Multi-Encoder Transformer Architecture. Scientific Reports, Vol. 14, 2024, Article No. 20762. DOI: 10.1038/s41598-024-69664-1
18. Aditya Akundi, Joshua Ontiveros, Sergio Luna. Text-to-Model Transformation: Natural Language-Based Model Generation Framework. Systems, Vol. 12, 2024, Article No. 369. DOI: 10.3390/systems12090369
19. Dhomas Hatta Fudholi, Royan Abida N. Nayoan. The Role of Transformer-based Image Captioning for Indoor Environment Visual Understanding. International Journal of Computing and Digital Systems, Vol. 12, No. 1, 2022, pp. 479-488. DOI: 10.12785/ijcds/120138
20. Yue Ming, Nannan Hu, Chunxiao Fan, Fan Feng, Jiangwan Zhou, Hui Yu. Visuals to Text: A Comprehensive Review on Automatic Image Captioning. arXiv preprint, 2024.
21. Shuang Liu, Liang Bai, Yanli Hu, Haoran Wang. Image Captioning Based on Deep Neural Networks. MATEC Web of Conferences, Vol. 232, 2018, Article No. 01052. DOI: 10.1051/matecconf/201823201052
22. Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 4, 2017, pp. 652-663. DOI: 10.1109/TPAMI.2016.2587640
23. Yugandhara A. Thakare, Kishor H. Walse. A Review of Deep Learning Image Captioning Approaches. Journal of Integrated Science and Technology, Vol. 12(1), 2024, p. 712. DOI: URN:NBN: science in.just.2024.v12.712
24. Palash, M. A. H., Nasim, M. A. A., Saha, S., Afrin, F., Mallik, R., & Samiappan, S. (2021). Bangla Image Caption Generation through CNN-Transformer based Encoder-Decoder Network. *arXiv preprint arXiv:2110.12442*