Project ID: 2024/CSE/3/06

Image Captioning using VGG-16 Deep Learning Model

A PROJECT REPORT
Submitted By

Vanshika Singh
(2000270100180)
Satyankar
(2000270100137)
Yuvika Singh
(2000270100197)
Vaibhav Tiwari
(2000270100178)

Under the Guidance of
Mr. Vishal Jayaswal
Assistant Professor
(Dept. of Computer Science and Engineering)

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering

to

Department of Computer Science & Engineering
AJAY KUMAR GARG ENGINEERING COLLEGE, GHAZIABAD
DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

Declaration

We hereby declare that the work presented in this report entitled **"Image Captioning using Deep Learning"**, was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors / sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors / sources.

We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

Name :Vanshika Singh
Roll No. : 2000270100180

Name : Satyankar
Roll No. : 2000270100137

Name : Yuvika Singh
Roll No. : 2000270100197

Name : Vaibhav Tiwari
Roll No. : 2000270100178

Certificate

This is to certify that the report entitled **"IMAGE CAPTIONING USING VGG-16 DEEP LEARNING MODEL"** submitted by Vanshika Singh (2000270100180), Satyankar (2000270100137), Yuvika Singh (2000270100197) and Vaibhav Tiwari (2000270100178) to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by him/her under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Vishal Jayaswal	Dr. Shashank Sahu
Assistant Professor, CSE	Professor-In-Charge, CSE
AKGEC, Ghaziabad	AKGEC, Ghaziabad

Place: Ghaziabad
Date:

Acknowledgements

We are profoundly grateful to Ajay Kumar Garg Engineering College for providing this wonderful opportunity and the conducive environment necessary for undertaking this research project. The institution's commitment to fostering research and innovation has been instrumental in bringing this project to fruition.

We would like to extend our deepest gratitude to our guide, **Mr. Vishal Jayaswal**(Assistant Professor - CSE Department), whose insightful guidance and unwavering support have been pivotal throughout the course of this research. His expertise in the field and his encouragement have greatly enriched this work.

Special thanks to the **Prof. Anu Chaudhary (Head of the Computer Science and Engineering Department)**, for his leadership and support. His dedication to academic excellence and her encouragement have greatly facilitated the progress of this research.

We are especially thankful to **Mr. Shashank Sahu (Professor-in-Charge, CSE Department)**, whose invaluable advice and continuous encouragement have been critical to the successful completion of this project. His dedication to guiding students and his wealth of knowledge have significantly shaped the direction and outcome of our research.

Finally, we are grateful to our families and friends for their constant encouragement and understanding during the course of this research. Their support has been a source of strength and motivation. This research would not have been possible without the collective efforts and support of all these individuals and institutions. Thank you for your invaluable contributions.

Abstract

Image captioning is a method that creates captions from images. It makes use of computer vision, natural language processing, and deep learning. The field of image captioning has evolved significantly in recent times, thanks to the application of both conventional and sophisticated deep learning approaches. This study uses two well-known benchmark datasets, Flickr8k and Flickr30k, to examine the use of the VGG-16 convolutional neural network (CNN) for producing descriptive captions. A recurrent neural network (RNN) is integrated to generate coherent and contextually relevant captions after the VGG-16 model is utilized for feature extraction. Human-provided references and generated captions are compared for quality using standard assessment metrics such as BLEU. The findings have applications in the areas of content indexing, assistive technology for the visually impaired, and enhancing user interfaces on image-centric systems. The comparison of the Flickr8k and Flickr30k datasets sheds light on the challenges posed by the different datasets and provides guidance for future image captioning research. A list of references, a synopsis of the key findings, and suggestions for future research topics are included in the paper's conclusion.

List of Figures

List of Tables

Table of Content

Chapter 1

Introduction

The advent of deep learning has significantly revolutionized various domains of artificial intelligence, among which image captioning stands out as a prominent application. Image captioning involves the automatic generation of textual descriptions for given images. This interdisciplinary challenge marries the complexities of computer vision and natural language processing, requiring systems to not only recognize and interpret the visual content but also to express it in coherent and contextually appropriate language. Over the past decade, deep learning techniques have demonstrated remarkable success in advancing the capabilities of image captioning systems, driving both academic research and practical applications forward.

Historically, the problem of image captioning has been approached using a variety of techniques ranging from template-based methods to sophisticated probabilistic models. Early methods primarily relied on manually crafted rules and were limited by their rigidity and inability to generalize across diverse datasets. With the advent of machine learning, especially deep learning, the landscape of image captioning has transformed. The motivation behind this shift is multifaceted. Deep learning models, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown unprecedented prowess in visual recognition and sequence generation tasks, respectively. This synergy has inspired researchers to develop integrated models that can learn to generate descriptions from images in an end-to-end manner, bypassing the need for extensive feature engineering and rule-based systems.

At the heart of modern image captioning systems lie two key components: the encoder and the decoder. The encoder, typically a CNN,

processes the input image and extracts high-level visual features. These features are then fed into the decoder, often an RNN or a variant such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs), which generates the corresponding textual description. This encoder-decoder architecture is analogous to the sequence-to-sequence models used in machine translation, adapted to handle the image-to-text modality.

Model Architecture and Training: The VGG-16 model, a convolutional neural network (CNN) developed by the Visual Geometry Group at the University of Oxford, is renowned for its performance and simplicity. Its architecture consists of 16 weight layers, including 13 convolutional layers and 3 fully connected layers. Each convolutional layer uses small receptive fields (3x3) and applies ReLU (Rectified Linear Unit) activation functions, which introduce non-linearity into the model. The network also includes max-pooling layers to reduce the spatial dimensions of the feature maps, thereby decreasing the computational load and improving the model's robustness to spatial variations in the input images.

In the context of image caption generation, the VGG-16 model serves as the encoder that extracts high-level visual features from input images. These features are then fed into a language model, typically an LSTM (Long Short-Term Memory) network, which generates the corresponding captions. The overall architecture is known as an encoder-decoder model, where the VGG-16 acts as the encoder and the LSTM functions as the decoder.

The LSTM network is particularly suited for sequential data tasks due to its ability to maintain long-term dependencies. It consists of memory cells and gates (input, output, and forget gates) that regulate the flow of information. This structure allows the LSTM to effectively capture the temporal dependencies in the sequence of words in the captions, ensuring coherence and grammatical accuracy.

Attention Mechanism: One significant advancement in image caption generation is the introduction of attention mechanisms. Attention mechanisms allow the model to focus on specific parts of the image while

generating each word of the caption. This is particularly useful for complex images with multiple objects and activities.

The attention mechanism can be implemented in various ways, such as soft attention and hard attention. Soft attention generates a weighted sum of the feature vectors from different parts of the image, where the weights represent the relevance of each part to the current word being generated. Hard attention, on the other hand, makes a discrete choice about which part of the image to focus on. Soft attention is differentiable and can be trained using gradient descent, whereas hard attention typically requires reinforcement learning techniques.

Training Procedure: Training the model involves optimizing the parameters to minimize a loss function, which measures the difference between the generated captions and the actual captions. Commonly used loss functions include cross-entropy loss and sequence-to-sequence loss. Cross-entropy loss evaluates the difference between the predicted probability distribution of the next word and the true distribution, while sequence-to-sequence loss considers the entire sequence of words in the generated caption.

Transfer Learning: Transfer learning is another important aspect of training deep learning models. It involves using a pre-trained network, such as VGG-16, as a starting point and fine-tuning it on the specific dataset for image captioning. This approach leverages the knowledge gained from training on a large dataset, such as ImageNet, and applies it to the target task, significantly reducing the amount of data and time required for training.

1.7 Traditional vs. Deep Learning Methods" Traditional methods for image captioning often rely on handcrafted features and rule-based approaches. These methods typically involve extracting features using techniques such as Scale-Invariant Feature Transform (SIFT) or Histogram of Oriented Gradients (HOG) and then using these features to generate captions based on predefined templates or rules.

Retrieval-Based Methods: Retrieval-based methods involve selecting the most relevant caption from a pre-existing database of captions

based on the similarity between the input image and the images in the database. This similarity can be measured using various metrics such as Euclidean distance or cosine similarity. Retrieval-based methods can produce high-quality captions when the input image closely matches an image in the database. However, they are limited by the diversity of the database and cannot generate novel captions for new or unique images.

Template-Based Methods: Template-based methods generate captions by filling in predefined templates with the appropriate content. For example, a template might be "A [object] is [action] in the [location]," which can be filled in with words like "dog," "running," and "park" to generate the caption "A dog is running in the park." These methods rely on accurate detection and classification of objects and actions within the image. While template-based methods can produce grammatically correct captions, they often lack the flexibility and creativity needed for more diverse and complex images.

Deep Learning Methods: Deep learning methods, on the other hand, leverage large amounts of data and powerful neural network architectures to learn complex representations and generate more accurate and diverse captions. Convolutional Neural Networks (CNNs) are used to extract high-level visual features from the images, while Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks generate the captions based on these features.

Encoder-Decoder Models: The encoder-decoder architecture is a common framework for deep learning-based image captioning. The encoder, typically a CNN such as VGG-16, extracts a fixed-length feature vector from the input image. This feature vector is then passed to the decoder, an RNN or LSTM, which generates the caption word by word. The encoder-decoder model is trained end-to-end using large datasets of images and corresponding captions.

Attention Mechanisms: Attention mechanisms have further improved the performance of deep learning-based image captioning models. By allowing the decoder to focus on specific parts of the image when generating each word, attention mechanisms help the model capture finer details and produce more accurate and contextually relevant captions.

This approach mimics the human visual attention process, where we focus on different parts of a scene when describing it.

Transformer Models: Transformer models, which use self-attention mechanisms to process the entire image and caption simultaneously, have set new benchmarks in image captioning performance. These models are able to capture long-range dependencies and generate more coherent and contextually appropriate captions. Transformers have been particularly successful in natural language processing tasks and have shown great promise in image captioning as well.

Challenges and Recent Advances: Despite the progress, several challenges persist in the field of image captioning. One major challenge is the generation of captions that are not only accurate but also rich, diverse, and contextually appropriate. Capturing subtle details, understanding context, and ensuring linguistic coherence remain open problems. Moreover, the evaluation of image captioning models poses its own set of difficulties. Commonly used metrics such as BLEU, METEOR, and CIDEr, while useful, do not always align perfectly with human judgment, necessitating the development of more sophisticated evaluation methodologies.

Regularization Techniques: To prevent overfitting, regularization techniques such as dropout and batch normalization are employed. Drop out involves randomly setting a fraction of the input units to zero during training, which helps in preventing the model from relying too heavily on any particular features. Batch normalization normalizes the inputs of each layer to have zero mean and unit variance, which speeds up training and improves the model's stability.

Data Augmentation: Data augmentation techniques such as rotation, scaling, and cropping are often used to increase the diversity of the training data and improve the model's robustness. By artificially expanding the dataset, these techniques help the model generalize better to new, unseen images.

Future Directions: The practical applications of image captioning are vast, spanning assistive technologies for visually impaired individuals,

automated content generation, and image indexing for search engines, to name a few. As the field continues to evolve, the integration of image captioning with other AI systems, such as conversational agents and augmented reality, presents exciting new possibilities.

Looking forward, future research is likely to focus on improving the contextual understanding of images, developing more robust and interpretable models, and enhancing the ability of these systems to generate captions that are not only syntactically correct but also semantically rich. The integration of multimodal data and the exploration of unsupervised and semi-supervised learning techniques will also be crucial in advancing the state-of-the-art in image captioning.

In conclusion, the field of image captioning using deep learning stands at a dynamic intersection of technological innovation and practical application. As deep learning models continue to evolve, the prospects for more accurate, versatile, and human-like image captioning systems grow ever brighter, heralding a future where machines can seamlessly interpret and describe the visual world.

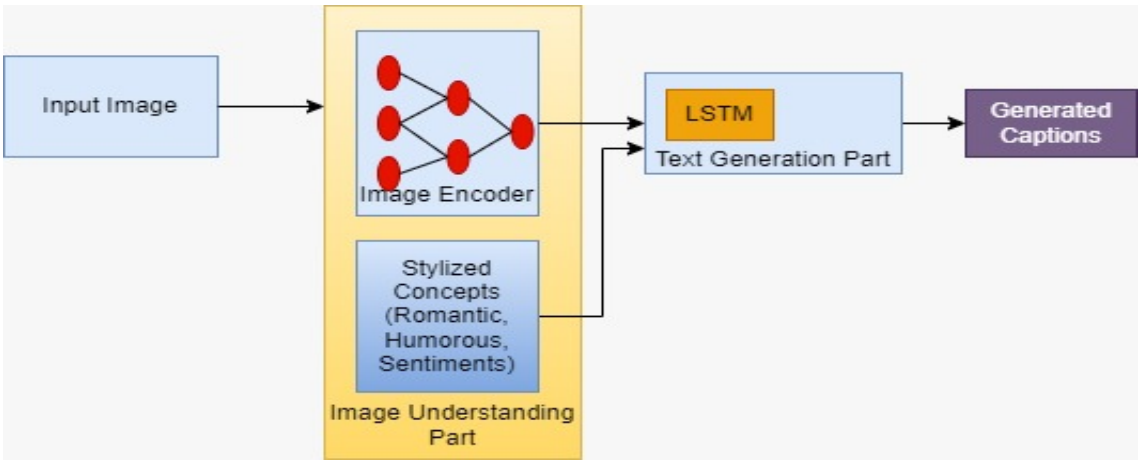


Figure 1.1: Basic Architecture

In fig 1.1, we are explaining the process behind the model. Process includes steps i.e. image features are obtained using convolutional neural network, [7] then attributes are extracted from visual features and multiple captions are created by language model and at last captions that are generated.

Chapter 2

Literature Review

This study examines various approaches to creating captions for images, including template-based, retrieval-based, and creative approaches. While retrieval-based methods draw captions from pre-existing ones, template-based approaches create captions using predetermined templates with unfilled slots. Deep machine learning-based techniques can be used to produce original captions from visual and multimodal domains. With the aid of these techniques, each image's caption can be produced with greater semantic accuracy[14]. Learning methodologies including supervised learning, reinforcement learning can be used and systems that are based on deep learning can be categorized using learning. Using a straightforward encoder-decoder architecture or compositional[15] architecture, captions can be produced for an entire scene or specific areas of an image. Some techniques make use of attention mechanisms, semantic ideas, and various image-description styles.

2.0.1 Traditional Approaches

Prior to the invention of artificial neural networks (ANNs)[1], thresholding and clustering algorithms, which are essentially unsupervised techniques, constituted the majority of segmentation and semantic segmentation approaches. Traditional semantic segmentation techniques typically take less time to compute models. Additionally, compared to the current era of deep learning and artificial neural networks, the majority of the approaches require less data.

Template-based and retrieval-based and techniques are the main approaches employed by early researchers to allow a computer with visual learning abilities to generate a coherent and understandable language

for the given image.[12]

Retrieval-Based Image Captioning

Storage of numerous image-caption pairs in a corpus is the basic goal of retrieval-based image captioning. It begins by using similarity comparison to find images in the query corpus that are comparable to the input image. The caption for the provided image is chosen based on the best annotation caption among the related candidates of the retrieved photos.

In order to characterize a given image, computers can use annotations from previously obtained photos. To choose the closest statement, they can use the Tree-F1 rule and the nearest neighbor rule. Global similarity is calculated by Ordonez et al., and the caption of the matching image is sent. Based on word order and syntactic details, Socher et al. concentrate on actions and subjects but ignore noise. To lessen visual estimating noise, Mason and Chiarniak suggest a nonparametric[11] density estimation method. Sun et al. use bi-directional retrieval to output captions after filtering text terms based on visual discrimination, organizing them into concepts.

The retrieval-based approaches select the most semantically related sentences or phrases from the database to create the description of the provided image. The automatically generated captions are typically syntactically sound, expressive, and closely related to natural language.One of the obvious problems is that the generated sentences keep the syntax of their expression and expression style due to excessive reliance on the annotation database, but there are other drawbacks as well.

Additionally, it limits the caption to statements or phrases that already exist and cannot be changed to fit new items or settings. The subtitles that are generated in some situations can even have nothing to do with the image that is being used.

Template-Based Image Captioning

Another model that was applied in early image captioning projects is Template Based. This approach created captions in a fairly limited way, both syntactically and semantically. For the purpose of creating captions, we previously employed fixed templates with a number of fixed

blank spots. Prior to using these extractions to fill in the gaps of those predetermined templates, the many properties, features, actions, and surrounds of the objects are first recognized. Consequently, the template is now appropriate for the given image. Here are a few template-based strategies: One method involved filling in the blanks in the templates using a triplet comprising a scene, an object, or an action. First, object identification techniques were used to estimate the items and scenes, then pretrained language models[1] were used to identify the verbs, prepositions, and situations to create a meaningful phrase.

Another approach renders image contents using Conditional Random Field (CRF). The objects, their properties, and their geographical relationship were all represented by the graphs' nodes. These are used to fill in any blank spaces in the templates, completing their descriptions of the image.

Unlike retrieval models, which produce grammatically incorrect sentences, template-based models produce descriptions that are considerably more pertinent to images. However, this method's two main drawbacks are that it employs predefined templates and is unable to produce captions of varying length. These inflexible standards also make the automated descriptions less organic when compared to handwritten captions by humans.

2.0.2 Neural Network-Based Captioning

Comparing deep learning to other machine learning techniques and algorithms, image captioning has greatly improved. These techniques require a substantial quantity of training data, comprising images and caption labels, and are mostly used in supervised environments. These astounding results have been attained by applying a variety of models, such as autoencoders ,generative adversarial networks (GAN)[3], convolutional neural networks (CNN), recurrent neural networks, artificial neural networks (ANN), and combinations of these.

Encoder-decoder framework

An encoder-decoder structure is commonly used for picture captioning jobs, in which words and images are controlled by separate models. While recurrent neural networks (RNNs) are used as decoders to

process extracted features concurrently with text labels, convolutional neural networks (CNNs) are used as encoders to extract features. CNNs help anticipate words by assisting with object identification, localization, and interactions in conjunction with long-term dependencies from a recurrent network cell.

Various CNN-based encoders have been suggested to offer a more thorough and reliable extraction of objects and their interactions from pictures. In order to fuse and embed semantics from different encoders and provide meaningful textual representations and descriptions, a novel recurrent fusion network (RFNet) was presented. In order to reduce the computational time required for picture captioning jobs, a combination of two CNN models was investigated. A concept-based phrase ranking technique was added to the CNN-LSTM model to improve visual description with less manual annotation needed. Using a generative adversarial network (GAN) trained on a binary vector, the image's sentiment was labeled.[9]

- **Encoder:** Using the pre-trained VGG-16 convolutional neural network (CNN) as a feature extractor is the first step in the encoding process for picture captioning using VGG-16. Obtaining the VGG-16 model's pre-trained weights is the first step; this model is usually trained on extensive image classification datasets such as ImageNet[9]. Pixel values are subsequently standardized and input photos are preprocessed to comply with the model's input size specifications. By feeding the input images through the network until they reach a chosen intermediate layer—typically the final fully connected layer before the classification layer—the VGG-16 model is used to extract high-level features from the images. Condensed representations of the image material are provided by these extracted features, which also capture important patterns and structures.

- **Decoder:** One kind of RNN (recurrent neural network) architecture called Long Short-Term Memory (LSTM) was created to solve the vanishing gradient issue with conventional RNNs. LSTMs are particularly useful for jobs involving sequences, such natural language processing, since they are good at collecting and maintaining long-term dependencies in sequential data.

Long Short-Term Memory (LSTM) networks play a crucial role as decoders in the image captioning environment, producing logical and contextually appropriate textual descriptions for visual content. The LSTM handles decoding after an encoder—typically a Convolutional Neural Network (CNN)—extracts useful representations from input images. Because LSTMs can capture and maintain long-term relationships in sequential data, they are especially well-suited for this task. This is because they can solve problems like the vanishing gradient problem that traditional recurrent neural networks encounter.

Using the word that were previously generated and the concealed state from the previous phase as input, the LSTM predicts the probability distribution across the vocabulary for the subsequent word in the sequence at each decoding step.
Until an end token is produced or the predetermined maximum sequence length is achieved, this process iteratively continues.

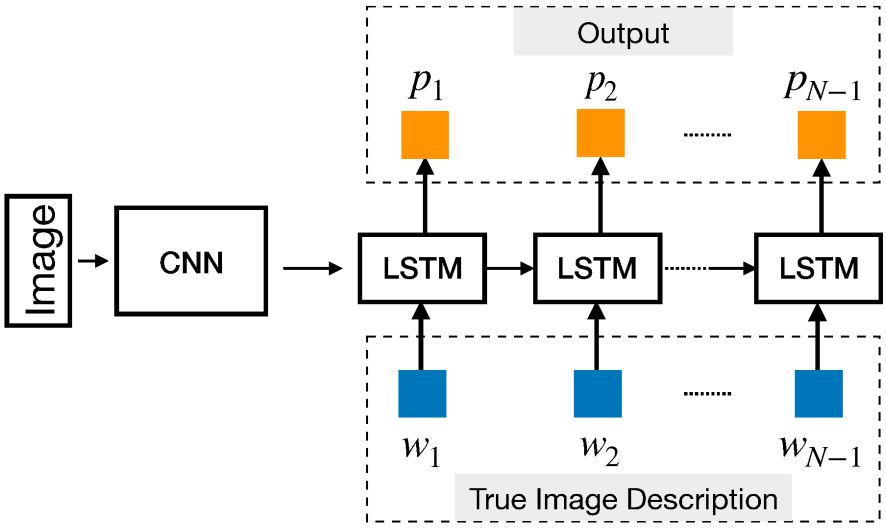


Figure 2.1: LSTM as a decoder

The LSTM's variables are fine-tuned through training to reduce the difference between the ground truth captions and the expected sequence. By using this decoding method, the model is able to provide linguistic coherence and contextuality to the generated captions, which enhances the ability of image captioning systems to comprehend and describe visual content.

The first step in employing LSTM as a decoder for image captioning is to extract visual characteristics from an image using a convolutional neural network that has already been trained, such as VGG-16. The LSTM decoder is initialized by these features, after which it processes the caption's word embeddings[16] one at a time. By capturing contextual information, the LSTM's hidden state makes meaningful predictions for every word. The model learns by reducing the discrepancy between words that are anticipated and those that occur during training. Until a termination token is anticipated or the maximum length is achieved, caption creation keeps going. The efficacy of LSTM in capturing sequential relationships and generating coherent language output is demonstrated by the model's inference process, which uses the learnt visual features to produce descriptive captions for input photos.

2.0.3 Deep Learning

A branch of machine learning called "deep learning" uses multi-layered neural networks, or "deep neural networks," to automatically learn data representations that are hierarchical. Deep learning is a key component that has enabled amazing progress in the field of picture captioning. Convolutional neural networks, or CNNs, are frequently used to extract features from images, identifying patterns and spatial hierarchies in the visual information. These CNN-based elements form the basis for comprehending the visual context of the image. Deep learning is a subset of machine learning that uses multi-layered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain.

Recurrent Neural Networks (RNNs) or, more recently, Transformer models are frequently used to produce captions that are both logical and pertinent to the context. RNNs make it possible to include sequential dependencies in language, which enables the model to represent the temporal organization of a sentence.

Conversely, transformer models have shown to be effective at understanding long-range dependencies and are gaining popularity due to their capacity for parallelization. End-to-end training is made easier by the deep learning framework, which enables the model to simultaneously

learn language and visual representations. This improves image captioning systems' overall performance and makes it possible for the model to generalize well to a variety of unknown input. Thus, the combination of deep learning and picture captioning has made it possible to generate automatic description creation that is more sophisticated and context-aware. This has a wide range of applications, from image indexing and retrieval systems to content accessibility.

Based on their fundamental structures, we have categorized and organized the various frameworks, methodologies, and approaches that have been widely used in recent research efforts. [6]

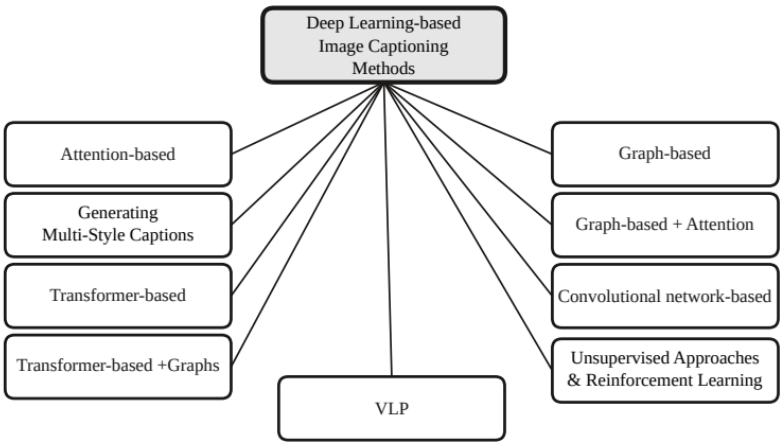


Figure 2.2: Taxonomy of image captioning methods

2.0.4 Attention-Based Methods

Attention-based techniques draw inspiration from human attentional patterns and the way the eye focuses on visuals. When seeing an image, humans are more likely to focus on its most prominent features. The same idea is used by attention-based systems[8]. As part of the training process, the model is shown "where to look at." To understand attention-based techniques, imagine a sequential decoder where the word "c" has a context vector under it, in addition to the output and internal state of the preceding cell.

Vector c represents the weighted sum of the hidden states of the encoder.

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{2.1}$$

The encoder state in input j is denoted by hj in the previous statement, and the "amount of attention" that output i needs to provide to input j is denoted by aij. Aij is derived by calculating softmax over the attention amounts denoted by e for the inputs and output i.

$$a_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{2.2}$$

$$e_{ij} = f(S_{i-1}, h_i) \tag{2.3}$$

2.0.5 Transformer-Based Methods

Originally developed for natural language processing applications, transf ormer-based models have become widely used in many other fields because of their ability to effectively parallelize computations and capture long-range dependencies. The Transformer architecture has been modified to support the collaborative modeling of textual and visual data in the context of picture captioning.[2] The following provides a thorough overview of the image captioning process using Transformer-based models:
An encoder-decoder structure is the foundation of the Transformer architecture. In image captioning, the decoder creates a logical and contextually appropriate caption, while the encoder examines the input image and extracts pertinent visual information.

Transformers' self-attention mechanism enables the model to accurately capture global dependencies by weighing various input sequence components while making predictions. Typically, a pre-trained Convolutional Neural Network (CNN)[5] is used by the encoder in image captioning jobs to extract visual information from the input image. After that, a sequence is created from these visual characteristics, which are then fed into the Transformer encoder. Word by word, however, the decoder is in charge of creating the caption. The Transformer decoder uses a combination of visual information from the encoder and contextual information from the words that come before it in the caption to forecast each word based on the context that it has encoded.

One advantage of Transformer-based models in image captioning is their ability to handle the inherent sequential and hierarchical structure of natural language. Transformers are particularly good at identifying word relationships within sentences, which guarantees that the resulting captions are both contextually consistent and semantically understandable. Transformers' parallelization capabilities also help to shorten training and inference periods.

Transformer-based models have been shown to be effective in image captioning across a range of benchmarks, exhibiting their capacity to produce precise and contextually relevant captions. The Transformer architecture's use of large-scale pre-trained models, like BERT or GPT, has improved image captioning systems' overall performance. Transformers are a potent option for pushing the boundaries of picture captioning applications due to their versatility in multimodal jobs, where both textual and visual input must be taken into account.

2.0.6 CLIP- Guided language modelling

With each generation phase, the goal is to steer the LM in the direction of the desired visual direction. Alignment with the provided image and preservation of language qualities are the goals of the guidelines.CLIP's embeddings for images and text share the same space, enabling direct comparisons between the two modalities. This is accomplished by training the model to bring related images and texts closer together while pushing unrelated ones apart Through CLIP[12], which evaluates token relatedness to an image and modifies the model accordingly, the first goal is accomplished. Regularizing the objective to resemble the initial target output is the second aim.

The table given below depicts the evolution of various methodologies in image caption generation research over the years. It highlights significant advancements and innovative approaches from 2015 to 2023, showcasing the progression and diversification of techniques in this field.

Table 2.1: Related Work

S.No.	Year	Author's Name	Approach	Name of paper
1.	2015	*Vinyals*	CNN-RNN framework	Show and Tell
2.	2015	*K Xu*	Attention Mechanism	Show, Attend and Tell
3.	2017	*Rennie*	Reinforcement Learning	Actor Critic Sequence Training
4.	2017	*J Li*	Adversarial Training	Adversarial Learning for Neural Dialogue Generation
5.	2018	*Parmar*	Transformer Based Mechanism	Image Transformer
6.	2019	*Liunian Harold Li*	Multimodal Approaches	Unified Vision-Language Pre-training
7.	2020	*Himanshu*	Retrieval Based Approach	Image captioning: a comprehensive survey
8.	2021	*Guerin*	Zero-shot and Few-shot Learning	Meta-Learning for Low-Resource Image Captioning
9.	2021	*Mokady*	CLIP Based Approach	Clipcap: Clip prefix for image captioning
10.	2023	*Wang*	CLIP Based Approach	Efficient image captioning for edge devices

Chapter 3

Datasets and Models

3.1 Datasets

For deep learning to fully understand patterns and optimize the number of parameters required for gradient convergence a large volume of training data is required.To facilitate evaluation and testing, it is standard procedure to partition the dataset into training, validation, and testing sets. Because of this separation, the model can be trained on one subset of the data, validated on another to adjust hyperparameters and track performance, and then tested on an entirely different subset to evaluate its generalization abilities. In order to prepare a solid and organized dataset for the ensuing training of picture captioning models, it is imperative that the preprocessing and data loading processes are carried out carefully. As a result, the following datasets are readily available and were created especially for the task of semantic segmentation.

- **Flickr8k:**

  A key dataset in the field of computer vision and processing of natural languages, Flickr8k is especially useful for image captioning. The dataset, which consists of 8,000 unique photographs taken from the Flickr site, is carefully labeled with five human-generated descriptions per image, providing a strong basis for training and testing image captioning algorithms. This dataset, which offers a wealth of image-caption pairs for training, is essential to the creation of these models. Flickr8k is used by researchers and practitioners to teach models the complex relationships between written descriptions and visual information. After training, models are then subjected to a thorough evaluation process on the Flickr8k data set to make sure they can produce captions on their own that are both seman-

tically meaningful and similar to the references created by humans. It is compact in size. So, on budget laptops or desktop computers, the model may be trained with ease.It includes eight hundred and ninety-two JPEG photos of various sizes and shapes. 6000 of which are used for development, 1000 for testing, and 1000 for training. includes text files that describe the trainset and testset. Flickr8k.token.txt has 40460 captions total—five captions for each image.

- **Flickr30k:**

  Like its predecessor, Flickr8k, Flickr30k is an extensive data set created to further image captioning research. It is composed of 31,000 photos taken from the photo-sharing website Flickr, each with five captions created by humans. This creates a huge corpus of various visual settings and activities. The dataset is divided into training, validation, and test sets, facilitating model development and performance evaluation. Flickr30K's images capture a wide range of scenes and activities, from everyday life to more specific scenarios, making it a valuable resource for developing algorithms that understand and describe visual content. Captions provide detailed information about objects, actions, attributes, and relationships within the images. As a standard benchmark for image captioning, Flickr30K supports significant advancements in machine learning and deep learning, enabling meaningful comparisons between different models.

  Larger training and assessment sets are needed, and this dataset fills that demand, enabling the creation of more reliable and scalable image captioning algorithms. A deeper comprehension of the complex interrelationships between descriptive language and visual content is provided by the various captions included for each image. The collection consists of 31,783 photos showing people going about their daily business. Every picture has a caption that describes it. Flickr30k is used to comprehend visual media (picture) that match a language expression (picture description). This dataset is frequently utilized as a reference standard for image descriptions based on sentences.

- **MS COCO**

  MS COCO, short for Microsoft Common Objects in COntext, stands as a cornerstone dataset in the realm of computer vision research. Featuring a vast compilation of over 200,000 images, it encompasses a broad spectrum of scenes, objects, and activities captured from diverse sources like Flickr. Each image within the dataset is meticulously annotated, providing invaluable resources for various computer vision tasks. These annotations include bounding boxes delineating objects of interest alongside categorical labels, facilitating tasks such as object detection and localization. Moreover, MS COCO offers pixel-level segmentation masks, enabling precise instance segmentation. Additionally, each image is paired with human-generated captions, offering detailed textual descriptions of the depicted scenes, thus serving as fertile ground for training and evaluating image captioning models. With 80 object categories covering everyday objects, animals, people engaged in diverse activities, and abstract concepts like sky and water, MS COCO presents a rich and diverse array of visual data, making it an indispensable asset for advancing computer vision algorithms and techniques.

- **CamVid**

  CamVid, short for Cambridge-driving Labeled Video Database, is a prominent dataset extensively used in the field of computer vision, particularly for high-resolution image segmentation tasks. It comprises 101 meticulously annotated images, each boasting a resolution of 960×720 pixels. These annotations categorize the images into 32 distinct object classes, facilitating precise segmentation and recognition tasks.

  One unique aspect of CamVid is the inclusion of the "void" class. This class is designated for regions within the images that do not clearly belong to any of the other predefined object classes. The "void" class is essential for handling ambiguous areas, occlusions, or regions with undefined objects, thereby enhancing the robustness of segmentation models.

  The dataset is further enriched with RGB class values, ranging from 0 to 255, for each object class. These values provide standardized color coding for the classes, which is crucial for visualizing and differentiating the segmented regions. Researchers and practition-

ers leverage CamVid to train, validate, and benchmark segmentation algorithms, making it an invaluable resource for advancing autonomous driving, urban scene understanding, and related computer vision applications.

- **Cityscapes**

  The Cityscapes dataset is an extensive and meticulously curated collection of urban scene images, captured from 50 different cities during various seasons, thus providing a rich and diverse array of environmental conditions and urban settings. These images were derived from original video footage, resulting in a dataset that is invaluable for tasks involving scene understanding and semantic segmentation. Cityscapes includes approximately 20,000 images with coarse annotations and 5,000 images with fine annotations, the latter being exceptionally detailed to ensure high accuracy for training and evaluation purposes.

  The dataset is structured around 30 distinct label classes, which are grouped into eight comprehensive categories: people, buildings, objects, void, cars, flat surfaces, nature, and sky. The 'people' category encompasses pedestrians and riders, capturing the human elements within the urban landscape. 'Buildings' includes various structures such as houses and offices, reflecting the architectural diversity of cities. The 'objects' category covers a range of urban items like poles, traffic signs, and streetlights. The 'void' class is used for areas that do not fit into the other categories, such as regions outside the cityscape, reflections, or ambiguous parts of the image. 'Cars' encompasses all types of vehicles including cars, trucks, buses, and motorcycles. 'Flat surfaces' include roads, sidewalks, and other planar elements essential to urban infrastructure. The 'nature' category captures natural elements like trees, bushes, and grassy areas within the city, while the 'sky' category represents the sky above the urban environment.

  The primary purpose of the Cityscapes dataset is to facilitate the detailed segmentation and understanding of complex urban scenes, making it a critical benchmark for assessing the performance of semantic segmentation algorithms. Its comprehensive annotations and the diversity of urban environments covered make Cityscapes an indispensable resource for researchers and developers. This dataset

significantly contributes to advancements in fields such as autonomo us driving, smart city development, and various applications requiring precise urban scene interpretation and understanding.

- **PASCAL VOC**

  The PASCAL Visual Object Classes (VOC) dataset is one of the most widely used benchmarks in the field of computer vision, particularly for semantic segmentation. Introduced in 2005, this dataset has become a cornerstone for evaluating and developing algorithms in various vision-related tasks. The PASCAL VOC dataset consists of 21 predefined classes, including both object and background categories, enabling comprehensive and detailed scene analysis.

  This dataset supports a wide range of tasks beyond just semantic segmentation. It is also utilized for human layout analysis, action classification, object detection, and image classification. The versatility of PASCAL VOC makes it a valuable resource for researchers aiming to develop and test models that can understand and interpret complex visual scenes.

  The dataset is divided into three main subsets for training, validation, and testing, with each subset containing a substantial number of images. Specifically, the training set includes 1,464 images, the validation set comprises 1,449 images, and the test set contains 1,456 images. This division ensures a robust framework for model training and evaluation, allowing for meaningful performance comparisons.

  Since its inception, the PASCAL VOC dataset has been used annually in open competitions, fostering continuous innovation and improvement in the field. These competitions have spurred the development of advanced techniques and models, significantly advancing the state of the art in computer vision. The dataset's long history and ongoing relevance underscore its importance and influence in the research community. Overall, the PASCAL VOC dataset is a pivotal resource that has driven significant progress in semantic segmentation and related computer vision tasks. Its extensive use in competitions and research has established it as a benchmark for evaluating the capabilities of new algorithms and techniques in understanding and interpreting visual data.

- **ADE20K**

  The ADE20K dataset is a comprehensive and widely utilized dataset designed for scene parsing and object detection tasks. It includes a total of 25,210 images, meticulously divided into 20,210 training images, 2,000 validation images, and 3,000 test images. This large and diverse collection of images makes ADE20K particularly well-suited for developing and evaluating models in scene understanding and segmentation.

  One of the standout features of the ADE20K dataset is its detailed annotations. Each image is accompanied by segmentation masks that delineate the boundaries of various objects within the scene. These masks are further enhanced with part segmentation masks, which provide even finer granularity by breaking down objects into their constituent parts. This level of detail is invaluable for training models to recognize not only whole objects but also their specific components.

  Additionally, ADE20K provides a text file for each image containing extensive metadata. This file includes information on the object classes present in the image, helping to identify instances of the same class and distinguish between different objects. The text file also offers a description of the image's contents, providing contextual information that can be used to improve scene understanding algorithms.

  Moreover, ADE20K's annotations include three-channel images, wh ich are standard RGB images that provide color information essential for many computer vision tasks. The inclusion of RGB images ensures that models can leverage color data alongside segmentation masks to improve accuracy and performance in object detection and scene parsing.

  In summary, the ADE20K dataset is a richly annotated, versatile resource that supports a wide array of computer vision tasks. Its extensive and detailed annotations, combined with a large and diverse set of images, make it an ideal benchmark for training and evaluating models in scene parsing, object detection, and comprehensive scene understanding. This dataset continues to play a crucial role in advancing the field of computer vision by providing a robust foundation for developing sophisticated algorithms and models.

- **Synthia** SYNTHIA, the Synthetic Collection of Imagery and Annotations, is an innovative and meticulously crafted dataset comprised of synthetic images generated from a high-resolution virtual city. This dataset is specifically designed to advance research in computer vision, particularly in semantic segmentation, object detection, and scene understanding. With a total of 13,407 training images, each image is meticulously annotated with segmentation masks, ensuring precise and consistent labeling that can be challenging to achieve with real-world data.

  The dataset includes thirteen predefined label classes that cover a broad range of typical urban elements, such as buildings, vehicles, vegetation, roads, sidewalks, fences, sky, and pedestrians, among others. These classes encapsulate various urban features, providing a comprehensive framework for developing models that need to recognize and interpret complex urban scenes. The high-resolution nature of the SYNTHIA images ensures that even small details are visible, which is crucial for training models to accurately segment fine-grained features in urban environments.

  Furthermore, the synthetic origin of SYNTHIA allows for extensive customization and control, enabling researchers to experiment with different conditions and parameters. This capability is particularly beneficial for developing and testing autonomous driving systems, where understanding and responding to a wide range of urban scenarios is critical. In summary, SYNTHIA is a comprehensive and versatile dataset that provides high-resolution, meticulously annotated synthetic images from a virtual urban environment. Its detailed and diverse content, combined with the advantages of synthetic data generation, makes it an invaluable resource for advancing research in semantic segmentation, object detection, and urban scene understanding.

3.2 Models

We use pre-trained models in deep learning because they offer significant advantages in terms of efficiency and performance. Training deep learning models from scratch requires substantial computational resources, large datasets, and considerable time.

Pre-trained models, having already been trained on extensive datasets, enable transfer learning, allowing new models to be fine-tuned for specific tasks quickly and with less data. This results in improved performance, as the pre-trained models have already learned to recognize patterns and features from their initial training. Additionally, pre-trained models help achieve better generalization and robustness when applied to new tasks, making them more effective in handling various data types and reducing the need for large, labeled datasets. Overall, the use of pre-trained models significantly accelerates the development process and enhances the quality of deep learning applications.

- **VGG16:**
  Architecture: VGG16 is a Convolutional Neural Network (CNN) that consists of 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers, culminating in a softmax layer. Each convolutional layer uses small receptive fields (3x3 filters) and follows a pattern of increasing depth (number of filters) while reducing spatial dimensions via max-pooling.

  Use in Image Captioning: In image captioning, VGG16 serves as the encoder part of the encoder-decoder architecture. It extracts high-level features from images, which are then passed to the decoder (usually an LSTM or GRU) to generate descriptive captions.

- **Inception (GoogLeNet)**:
  Architecture: Inception, also known as GoogLeNet, introduced the inception module, which applies multiple convolutional and pooling operations with different filter sizes simultaneously and concatenates their outputs. The original Inception (GoogLeNet) model has 22 layers deep and includes auxiliary classifiers to combat the vanishing gradient problem.

  Use in Image Captioning: Inception models are used as feature extractors in image captioning systems. They capture diverse spatial hierarchies and complex patterns in images. The extracted features are then fed into a sequence model to generate captions.

- **Xception**:
  Architecture: Xception stands for "Extreme Inception" and improves on the inception architecture by using depthwise separable

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312

Total params: 134,260,544
Trainable params: 134,260,544
Non-trainable params: 0

Figure 3.1: Layers of VGG 16 model

convolutions instead of standard convolutions in the inception modules. This modification results in a more efficient model with 36 convolutional layers structured into 14 modules, each with residual connections.

Use in Image Captioning: Xception is used to encode images into feature vectors in image captioning tasks. Its efficient design helps in extracting detailed image features while maintaining computational efficiency. These features are then used by a decoder to produce textual descriptions.

- **ResNet (Residual Networks)**:
  Architecture: ResNet, particularly ResNet50 and ResNet101, uses residual blocks to enable training of much deeper networks. Each block includes identity mappings and shortcut connections that skip one or more layers. This architecture helps in mitigating the vanishing gradient problem, allowing models to go as deep as 152 layers.

  Use in Image Captioning: ResNet models are powerful feature extractors in image captioning frameworks. The residual connections allow for the preservation of gradients, leading to more effective feature extraction. These features are passed to a decoder for caption generation.

- **DenseNet (Dense Convolutional Network)**:
  Architecture: DenseNet, including variants like DenseNet121 and DenseNet169, connects each layer to every other layer in a feed-forward fashion. Each layer receives inputs from all preceding layers, enhancing feature reuse and gradient flow. The dense connections reduce the number of parameters while improving performance.

  Use in Image Captioning: DenseNet models extract detailed and richly connected features from images. In image captioning, these features help in generating more accurate and descriptive captions by providing the decoder with comprehensive image information.

- **EfficientNet**:
  Architecture: EfficientNet models, ranging from EfficientNet-B0 to EfficientNet-B7, use a compound scaling method to balance network depth, width, and resolution systematically. This approach results in a family of models that achieve high performance with fewer parameters and lower computational cost.

Use in Image Captioning: EfficientNet models are used as encoders in image captioning due to their high efficiency and strong performance. They provide high-quality feature representations that enhance the caption generation process..

- **MobileNet**:
  Architecture: MobileNetV1, MobileNetV2, and MobileNetV3 are lightweight models designed for mobile and resource-constrained environments. They use depthwise separable convolutions (MobileNetV1), inverted residuals and linear bottlenecks (MobileNetV2), and a combination of the two with further optimizations (MobileNet V3).

  Use in Image Captioning: MobileNet models are preferred in scenarios where computational resources are limited. They efficiently extract features from images, which can then be used by lightweight decoders to generate captions.

- **Inception-ResNet**:
  Architecture: Inception-ResNet combines the inception modules with residual connections, blending the advantages of both architectures. It includes deep inception modules for capturing diverse features and residual connections to maintain gradient flow during training.

  Use in Image Captioning: Inception-ResNet models serve as powerful feature extractors in image captioning systems. The combination of inception modules and residual connections ensures robust feature extraction, aiding in the generation of high-quality captions.

Chapter 4

Evaluation Metrics

We assess the automatically generated captions to make sure they accurately describe the provided image. Some popular metrics for evaluating image captioning in machine learning are listed below.

- **BLEU(BiLingual Evaluation Understudy):**

  When comparing the number of matched n-grams in the model's prediction to the actual data, the BLEU metric counts them. As a result, the mean number of computed grams is used to determine precision, while the caption label's shortness penalty is used to determine recall.[17] The BLEU (Bilingual Evaluation Understudy) score is a commonly used statistic to assess the caliber of text created by machines, including captions for images. When it comes to captioning images, the BLEU score evaluates how similar a generated caption is to one or more reference captions that have been annotated by humans. There are multiple steps in the calculation. Tokenization of the generated and reference captions into individual words or n-grams is the first step. The precision, which indicates the ratio of overlapping n-grams[18] in the generated caption to the entire number of n-grams, is next calculated for each n-gram length independently. The geometric mean of the precision scores over the various n-gram lengths is then used to get the overall precision. Brevity penalty (BP) is used to correct for any biases favoring shorter captions.

  The difference between the generated caption's length and the average length of the reference captions, divided by the generated caption's length, is the shortness penalty. This is computed as the exponential of the minimum of 1. Multiplying the overall precision

by the brevity penalty yields the final BLEU score. The BLEU score can be expressed mathematically as follows:

$$BLEU = BP \times \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log(precision_n)\right) \tag{4.1}$$

where BP is the brevity penalty, N is the maximum n-gram length considered, and precision-n[14] is the precision for n-grams. This comprehensive formula captures both precision and brevity aspects, providing a numerical representation of the quality of machine-generated captions in image captioning tasks.

- **ROUGE (Recall-Oriented Understudy for GistingEvaluation):**

  Description: ROUGE is another popular metric, particularly used in the context of summarization and translation. Unlike BLEU, which focuses on precision, ROUGE emphasizes recall—the ability to capture all the relevant information from the reference captions. It measures the overlap of unigrams, bigrams, and higher-order n-grams between the reference and predicted sequences.

  Variants: The most commonly used variants of ROUGE include ROUGE-N, which counts the overlap of n-grams; ROUGE-L, which measures the longest common subsequence; and ROUGE-S, which evaluates skip-bigrams (pairs of words allowing for gaps).

- **METEOR (Metric for Evaluation of Translation with Explicit Ordering):**

  Description: METEOR was developed to address some of the limitations of BLEU. It evaluates machine-generated text by considering both precision and recall, and it incorporates a penalty for incorrect word order. METEOR aligns the generated caption with the reference caption by considering synonyms, stemming, and paraphrasing, making it more flexible and semantically aware than BLEU.

  Calculation Steps:

  1. Exact Matches: Initially, METEOR looks for exact word matches Stem and Synonym Matches: It then considers matches based on word stems and synonyms, allowing for a broader range of acceptable matches.

2. Chunking: The aligned words are grouped into chunks, and a penalty is applied based on the number of chunks, encouraging longer contiguous matches.
3. Score Calculation: The final METEOR score is computed as a harmonic mean of precision and recall, adjusted by a fragmentation penalty based on the alignment chunks. between the generated and reference captions.

- **SPICE:**

  SPICE (Semantic Propositional Image Caption Evaluation) is a relatively new metric designed to assess the quality of image captions by focusing on the semantic relationships between the reference caption and the generated caption. Unlike traditional metrics such as BLEU, ROUGE, or METEOR, which primarily rely on n-gram overlaps, SPICE evaluates the alignment of semantic content, providing a more meaningful measure of caption quality. It employs a graph-based approach where captions are parsed into scene graphs that represent objects, attributes, and their relationships within a scene. These scene graphs offer a structured representation that details the entities present in an image, the attributes of these entities, and the relationships between them. For example, in an image caption like "A dog chasing a ball," the scene graph would include nodes for "dog" and "ball," an attribute indicating that the "dog" is "chasing," and an edge representing the action connecting the "dog" node to the "ball" node. This focus on semantic content rather than surface-level word matches allows SPICE to provide a more nuanced evaluation of caption quality, measuring how well the generated caption captures the essential details and relationships described in the reference caption. Consequently, SPICE is particularly useful for scenarios where the generated text should faithfully represent the underlying meaning and structure of the reference text, offering insights into the semantic fidelity of generated descriptions and guiding the development of more accurate and meaningful captioning systems.

Chapter 5

Implementation

5.1 Preprocessing and Data Loading

The image captioning process involves preprocessing and data loading to convert unprocessed image and text data into a machine learning model. Preprocessing involves cropping, normalization, and data augmentation techniques. Text preprocessing prepares written captions, while data loading organizes processed photos and captions into a training format. Batching speeds up the training process, while data shuffles prevent erroneous correlations. A data pipeline manages data loading and preprocessing on-the-fly during training.

Fig 5.1 shows the flow diagram of image captioning process , in this an image dataset is given as input to CNN(encoder) for feature extraction , after that caption sequences are generated using LSTM (decoder) [13] and then resulting captions are generated.

The image captioning process involves preprocessing and data loading to convert unprocessed image and text data into a machine learning model. Preprocessing involves cropping, normalization, and data augmentation techniques. Text preprocessing prepares written captions, while data loading organizes processed photos and captions into a training format. Batching speeds up the training process, while data shuffles prevent erroneous correlations. A data pipeline manages data loading and preprocessing on-the-fly during training.

- Load the Flickr8k dataset, which is made up of pictures with captions.

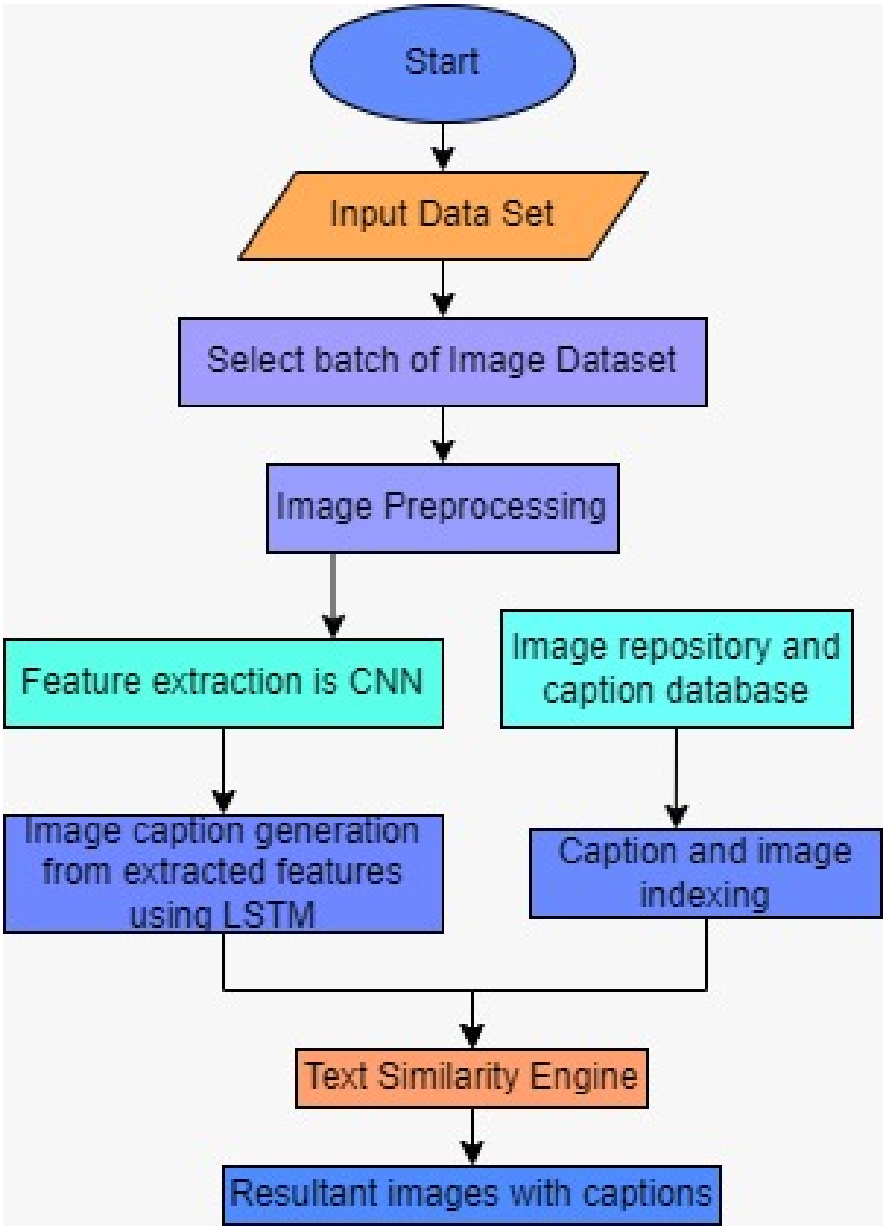


Figure 5.1: Flow Diagram

5.1.1 DataPreprocessing

- Data Cleaning: As we see all image captions are available in the Flickr 8k.token file of the Flickr8k text folder. If you analyze this file carefully, you can drive the format of image storing, each image and caption separated by a new line and carry 5 captions numbered from 0 to 4 along with.
  - load_fp(filename)
  - mg_capt(filename)
  - txt_cleaning(descriptions)
  - txt_vocab(descriptions)
  - save_descriptions(descriptions, filename)
- VGG-16 Feature Extraction:
  - Use a pre-trained VGG-16 model that was initially trained on ImageNet to extract high-level features from the images.
  - The output of the final convolutional layer should be retained while the VGG-16 fully connected layers are removed. This output results in the visual representation.
  - For the purpose of captioning photos, the VGG-16 model collects high-level information from the pictures. Rich representation of the input image is provided by the last convolutional layer of VGG-16, which preserves spatial information. Semantic aspects that include objects, forms, and patterns found in the image are recorded by VGG-16. These characteristics are essential for comprehending the visual context and producing insightful captions.
- Loading dataset for model training: A file named "Flickr8k.trainIma ges.txt" is present in our Flickr8k test folder. This file carries a list of 6000 image names that are used for the sake of training.

  Functions required to load the training datasets:
  - load_photos(fname)
  - load_clean_descriptions(fname, image)
  - load_features(photos)

- Tokenizing the vocabulary: Machines are not familiar with complex English words so, to process model's data they need a simple numerical representation. That's why we map every word of the vocabulary with a separate unique index value. An in-built tokenizer function is present in the Keras library to create tokens from our vocabulary. We can save them to a pickle file named "tokenizer.p".

- Create a data generator: For training the model as a supervised learning task we need to feed it with input and output sequences. Total 6000 images with 2048 length feature vector and the caption represented as numbers are present in our training sets. It's not possible to hold such a large amount of data into memory so we are going to use a generator method that will yield batches

5.2 Model Architecture

- Construct an image captioning model architecture by fusing a language model for caption generation with the image attributes from VGG-16.

- To turn the words in the captions into dense vectors, use an embedding layer.

- As the decoder, use an LSTM (Long Short-Term Memory) network to produce the captions' consecutive words.

5.3 Training with Flickr8k

The Flickr8k dataset, consisting of 8,000 photos with captions, is used to train a model for image captioning. Data preprocessing involves cleaning and tokenizing captions, shrinking images, and normalizing pixel values. A Convolutional Neural Network extracts features, while tokenization and padding prepare captions for input. An encoder-decoder architecture with LSTM or GRU[7]cells is used, with sequence-based loss functions and optimization approaches.

5.4 Refine the model using Flickr30k

- Adjust the model using the Flickr30k dataset. To do this, load the weights from the Flickr8k-trained model and carry on with the new

dataset's training.

- By fine-tuning, the model may be made to work with a larger dataset and recognize a wider range of patterns in the image-caption pairs.

5.5 Retuning the Model on Flickr8k

- Adjust the model once more using the Flickr8k dataset. This step aims to improve generalization by maximizing the model's output on the original dataset. We can see Table II shows images and a comparison between reference caption of the dataset and predicted captions by our model.

5.6 Testing and Evaluation

- Analyze the trained model's performance with metrics like BLEU, METEOR, etc., or on a different validation set.

- The accuracy of translations produced by machines is gauged by the BLEU (Bilingual Evaluation Understudy) score. Based on word sequences (n-grams) that overlap between the proposed and reference translations, it determines precision. Brevity Penalty takes into consideration translations that are shorter. The geometrical median of n-gram precisions[10], shortened for clarity, is the BLEU score. Higher scores denote higher quality translation; scores range from 0 to 1.

5.7 Optimization and adjustments

- Model Complexity: Modify the LSTM decoder's and other model elements' complexity. A model that is too simple could have trouble capturing complicated relationships, whereas a model that is too complex could result in overfitting.

- Training epochs and Batch Size: Try different batch sizes and epoch counts. In order to prevent overfitting or underfitting, keep an eye on both training and validation performance across epochs.

- Assessment Measures: To obtain a more thorough grasp of caption quality, take into account utilizing additional assessment metrics in addition to BLEU[4], such as METEOR, CIDEr, and ROUGE.

- Data Augmentation: Use methods such as picture flipping, rotation, and scaling to add more images to the training dataset. The resilience and generalization of a model can be strengthened through augmentation.

- Hyperparameter Search: Look for the best possible hyperparameters by methodically examining various architectural parameters, LSTM units, [19] and embedding dimensions. Either a random or grid search may be used in this investigation.

- Error Analyzation: Analyze model errors in test or validation sets using error analysis. Recognize the different kinds of errors the model commits and modify the training plan accordingly.

In this chapter, we provide a detailed explanation of how data was collected for the research and the methods used for data analysis. This section outlines the process, tools, and techniques employed to gather and analyze the data relevant to our study.

5.8 Data Collection Methods

The Flickr8K dataset represents the model training of image caption generators. The dataset is downloaded directly. The downloading process takes some time due to the dataset's large size(1GB). In the image below, you can check all the files in the Flickr8k text folder. The most important file is Flickr 8k.token, which stores all the image names with captions. 8091 images are stored inside the Flicker8k Dataset folder and the text files with captions of images are stored in the Flickr8k text folder.

5.9 Data Analysis Techniques

5.9.1 Quantitative Analysis

- **Statistical Methods**: Descriptive statistics to summarize dataset characteristics.

- **Performance Metrics**: Using metrics like BLEU, METEOR, and CIDEr to evaluate captioning model performance.
- **Comparative Analysis**: Comparing different models and their results quantitatively.

5.9.2 Qualitative Analysis

- **Human Evaluation:** Conducting surveys or employing annotators to assess caption quality.
- **Content Analysis**: Analyzing captions for creativity, coherence, and relevance
- **Error Analysis**: Identifying common errors made by the model and understanding their causes.

5.9.3 Libraries used

- **OS:** The os library in Python provides a way of interacting with the operating system. It offers various functionalities like file and directory operations, environment variables, and process management.
  Some Method Names: getcwd, listdir, mkdir, rmdir, remove, rename

  Usage in Project: The os library helps in managing datasets, creating directories (using mkdir), storing generated captions or models, and navigating through the file system.

- **pickle:** Python's pickle module is a popular format used to serialize and deserialize data types. This format is native to Python, meaning pickle objects cannot be loaded using any other programming language.

  Usage in Project: pickle is utilized to save and load trained models, tokenizer objects, or any other Python objects related to the caption generation process.

- **pandas:** The pandas is a powerful Python library for data manipulation and analysis. It provides data structures like DataFrame and Series, along with functions to manipulate, clean, and analyze data.

Usage in Project: pandas helps in loading and preprocessing metadata associated with images, such as image descriptions, image IDs, and annotations. It provides efficient data structures like DataFrames to handle and organize this information. Uses in image caption generation include cleaning, analyzing, filtering, and visualizing the data.

- **numpy:** The numpy is a fundamental package for scientific computing with Python. It provides support for multi-dimensional arrays and matrices, along with a wide range of mathematical functions to operate on these arrays efficiently.

  Usage in Project: numpy is used for converting images into arrays for processing, performing mathematical operations on image features extracted by neural networks, or manipulating data structures related to captions and embeddings.

- **graphviz:** The graphviz is a Python interface to the Graphviz graph visualization software. It allows you to create and render graphs and visualize relationships between entities.

  Usage in Project: graphviz is useful for visualizing neural network architectures or any graph-based representations involved in the project. It helps in understanding the structure of the models used for image feature extraction or caption generation, facilitating model design and debugging.

- **pydot:** The pydot is another Python interface to Graphviz, specifically focused on creating, parsing, and manipulating DOT language graphs.

  Usage in Project: pydot is used for creating, parsing, and manipulating graph representations. It can aid in visualizing and understanding the connections between different components of the image captioning system, such as the relationship between image features and generated captions.

- **tensorflow:** The tensorflow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and community resources for building and deploying machine learning models.

Usage in Project: tensorflow provides a robust framework for building and training neural networks, including models for image feature extraction and text generation. It is used to implement and train deep learning models like convolutional neural networks (CNNs) for image feature extraction or recurrent neural networks (RNNs) for generating captions.

- **tqdm:** The tqdm is a Python library that adds progress bars to loops and iterators. It provides a simple way to monitor the progress of operations, making it especially useful for tasks that take a long time to complete.

  Usage in Project: tqdm is used to monitor the progress of time-consuming operations during training or evaluation of the image captioning models. It provides progress bars that indicate the completion percentage of tasks, making it easier to track the training process and estimate remaining time.

- **PIL (Python Imaging Library):** PIL is a library in Python used for opening, manipulating, and saving many different image file formats. It provides powerful image processing capabilities and is widely used in various applications.

  Usage in Project: PIL provides functionalities for opening, manipulating, and saving image files. It is used to preprocess images before feeding them into the neural network for feature extraction, including tasks such as resizing, cropping, or normalizing images to ensure compatibility with the model's input requirements.

- **io:** The io module provides Python's main facilities for dealing with various types of I/O. It allows handling of streaming data, such as reading from or writing to files, strings, and memory buffers.

  Usage in Project: io is used for handling streaming data, such as reading images from memory buffers or writing generated captions to files. It facilitates the loading and processing of image data, as well as managing input and output streams for communication between different components of the system.

- **gc (Garbage Collector):** The gc module in Python provides an interface to the garbage collector for automatic memory management. It helps manage memory by recycling objects that are

no longer needed, thereby preventing memory leaks and improving performance.

Usage in Project: gc helps in managing memory usage and preventing memory leaks. Since deep learning models and large datasets can consume a significant amount of memory, enabling the garbage collector can ensure efficient memory management and prevent potential performance issues due to memory constraints.

- **Keras:** Keras, essential for image caption generation, simplifies model construction and training with its user-friendly interface. It seamlessly integrates CNNs and RNNs, allowing rapid experimentation and prototyping. Compatible across hardware setups and frameworks like TensorFlow, Keras streamlines model development. In our Flickr dataset project, Keras facilitated model definition, training, and evaluation, underscoring its pivotal role in advancing captioning research.

Chapter 6

Results and Findings

6.1 Quantitative Results

6.1.1 BLEU Scores

BLEU (Bilingual Evaluation Understudy) Score is a metric used to evaluate the quality of text generated by comparing it to one or more reference texts. In the context of image caption generation, the BLEU Score measures the precision of n-grams (contiguous sequences of n words) in the generated captions against the reference captions. The calculation involves several steps:

1. Tokenization: Breaking down both the generated and reference captions into tokens (words or n-grams).

2. N-gram Precision: Calculating the precision for various n-grams (unigrams, bigrams, trigrams, and four-grams) by comparing the n-grams in the generated captions to those in the reference captions.

3. Clipping: Applying a clipping mechanism to limit the count of each n-gram to the maximum count found in any single reference caption, preventing the model from being overly rewarded for repetitive n-grams.

Table 6.1: BLEU Scores for Datasets

Dataset	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
Flickr8k	0.53	0.30	0.11	0.17
Flickr30k	0.53	0.27	0.16	0.08
Flickr8k trained on Flickr30k	0.56	0.34	0.22	0.13

4. Geometric Mean: Combining the precision scores of different n-grams using a weighted geometric mean, resulting in a final BLEU score that typically ranges from BLEU-1 (unigrams) to BLEU-4 (four-grams).

To address the issue of short generated captions, a brevity penalty (BP) is applied, which is calculated based on the length ratio between the generated and reference captions. The final BLEU Score is the product of the geometric mean of the precisions and the brevity penalty. This metric provides an objective way to assess the performance of image caption generation models by comparing them to human-generated reference captions, ensuring a balance between accuracy and length.

6.1.2 Epoch V/s Accuracy Graph

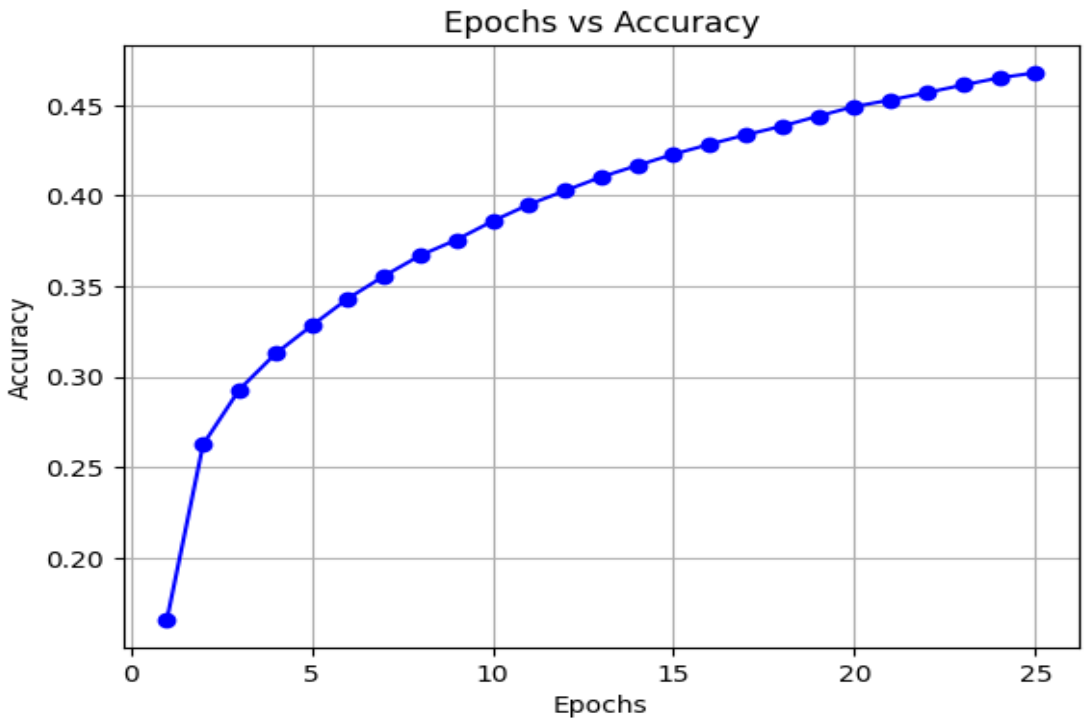


Figure 6.1: Graph for Flicker8k

The accuracy graphs show the model's performance over the training epochs for the Flickr 8k and Flickr 30k datasets:

- Epoch vs. Accuracy: This graph plots the number of epochs (x-axis) against the model's accuracy (y-axis). The progression of the graph indicates how well the model is learning:

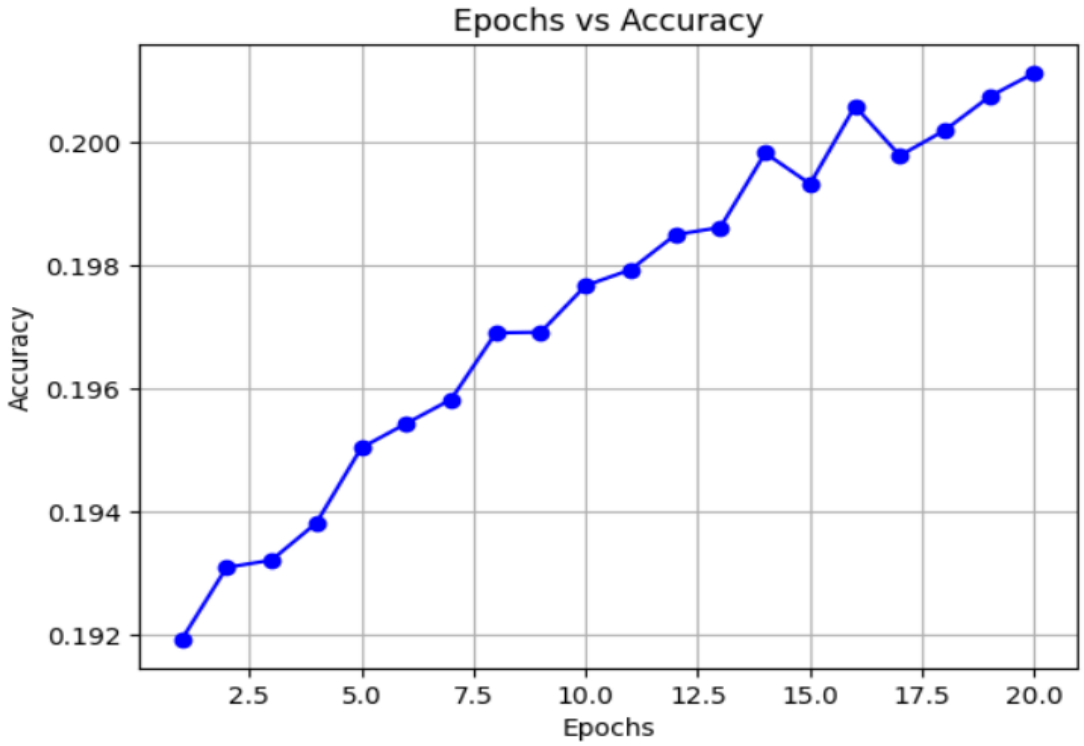


Figure 6.2: Graph for Flickr8k trained on Flickr30k

1. Early Epochs: Both training and validation accuracies are low, reflecting the model's initial learning phase.
2. Mid Epochs: Accuracies improve as the model learns from the data.
3. Later Epochs: The model may show signs of overfitting, where the training accuracy continues to increase while validation accuracy stabilizes or decreases.

It provides valuable insights into the training dynamics and performance of the image caption generation model using the VGG-16 architecture with Flickr 8k and Flickr 30k datasets. This graph plots the number of epochs on the x-axis against the accuracy on the y-axis, illustrating how the model learns over time. In the initial epochs, both training and validation accuracy are low, reflecting the model's early learning stage. As training progresses, the accuracy increases, indicating that the model is effectively learning the patterns in the data. For the Flickr 8k dataset, the graph may show quicker convergence due to the smaller dataset size, but it may also exhibit early overfitting, where training accuracy continues to rise while validation accuracy decreases. In contrast, the Flickr 30k dataset, being larger and more diverse, may demonstrate a

more gradual and sustained improvement in accuracy, with a later onset of overfitting. This comparative analysis highlights the importance of dataset size and diversity in training deep learning models, where larger datasets like Flickr 30k generally lead to better generalization and higher validation accuracy.

6.2 Qualitative Results

- Caption Quality:
    1. Relevance: Captions should accurately describe the main content of the images.
    2. Clarity: Captions must be clear and understandable.
    3. Creativity: The ability to generate varied and interesting descriptions.
- Content Understanding: Analyzing the model's ability to identify objects, people, actions, scenes, and contextual relationships in images.
- Language Proficiency: Examining the grammar, syntax, and vocabulary usage in generated captions. Ensuring linguistic accuracy and richness.

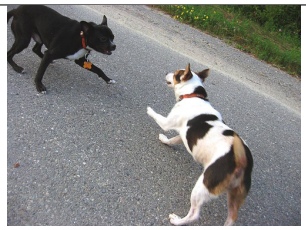
6.2.1 Generated Captions

The table below displays the images from the Flickr dataset along with the captions generated by the model and the corresponding reference captions. This comparison helps evaluate the accuracy and quality of the generated captions against human-authored references.

6.3 Comparison with Research Objectives

The research objectives were to assess the model's caption generation accuracy using BLEU scores, focusing on unigrams and bigrams. The Dataset Model BLEU 1 BLEU 2 BLEU 3 BLEU 4 METEOR Flickr8k model demonstrated significant word-level precision, reflecting its ability to generate accurate captions based on reference data. Challenges encountered during the experiment provide valuable insights for future improvements in data preprocessing and model architecture.

Table 6.2: Captions Mapping

Image Id	Image	Caption
1002674143_1b742ab4b8		• **Reference Caption:** A little girl covered in paint sits in front of a painted rainbow with her hands inside bowl • **Predicted Caption:** two children enjoy the grass with fingerpaints in the background
101669240_b2d3e7f17b		• **Reference Caption:** A man in hat is displaying pictures next to a skier in a blue hat • **Predicted Caption:**Two people are the snow white in the snow
1001773457_577c3a7d70		• **Reference Caption:** A black dog and a spotted dog are fighting • **Predicted Caption:**Two dogs are playing in the grass
1096395242_fc69f0ae5a		• **Reference Caption:** A boy with a toy gun pointed at the camera • **Predicted Caption:** Young girl in the camera

6.4 Discussion of Findings

The conducted experiments resulted in the evaluation of the model's performance using metrics such as BLEU-1, BLEU-2, BLEU-3, BLEU-4, and METEOR. BLEU-1 and BLEU-2 scores assess word-level accuracy, while BLEU-3 and BLEU-4 evaluate trigrams and four-grams, respectively.

6.5 Limitations of the Study

The study identified several limitations:

1. Object Hallucination: The model sometimes imagined objects not present in the image.
2. Missing Context: The model occasionally failed to capture the full context of the image.
3. Illumination Conditions: Variations in lighting affected the model's accuracy.
4. Contextual Understanding: Difficulty in understanding complex scenes and relationships.
5. Referring Expressions: Challenges in generating appropriate referring expressions for objects and actions.

6.6 Conclusion

We discussed different evaluation metrics and datasets with their strengths and weaknesses. A brief summary of experimental results was also given. We briefly outlined potential research directions in this area. Although deep-learning-based image captioning methods have achieved remarkable progress in recent years, a robust image captioning method that is able to generate high-quality captions for nearly all images is yet to be achieved. To overcome these challenges and enhance model performance, the chapter suggests several future research directions:

- Improving Data Preprocessing Techniques: Enhancing preprocessing methods to create more representative and high-quality datasets. This includes better image augmentation, cleaning of textual data, and balancing of dataset classes.

- Exploring Novel Deep Learning Architectures: Investigating new architectures, such as transformers and attention mechanisms, that can better capture contextual and sequential information in images.
- Integrating Multimodal Information: Combining visual information with other modalities, such as audio or textual context, to provide a more comprehensive understanding and generate richer captions.
- Focusing on Real-World Applications: Developing models that can handle real-world variability and complexity, ensuring robustness and reliability in practical applications.
- By addressing these areas, future research can push the boundaries of image captioning, creating models that are not only more accurate but also more versatile and useful in a variety of real-world contexts.

With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for some time.

Chapter 7

Discussion

The results obtained from algorithms for deep learning and machine learning, as well as their backbones, differ based on their ability to acquire mappings from image inputs to labels. CNN-based approaches are among the most efficient for tasks involving images due to their capacity to learn hierarchical features directly from raw image data. However, they can be computationally expensive, requiring substantial resources for training and inference.

Traditional machine learning algorithms, such as Markov random fields, random forests, ensemble modeling, naive Bayes, and support vector machines (SVMs), are comparatively simpler. These methods primarily rely on handcrafted feature engineering or domain-specific expertise to extract relevant features from images. While effective in some contexts, these algorithms often fall short in capturing the complex, high-dimensional nature of image data. Clustering algorithms like K-means and fuzzy C-means also struggle with image data, particularly when there are numerous and intricate boundaries to define.

Supervised learning remains a popular technique for image captioning. However, data augmentation and generative adversarial networks (GANs) have emerged as crucial components in enhancing the performance of these models. Data augmentation techniques artificially expand the training dataset, allowing the model to generalize better to unseen images. GANs, on the other hand, generate realistic images to further augment the training process, providing more varied examples for the model to learn from.

Deep learning models, particularly those employing an encoder-decoder architecture, have demonstrated superior performance across nearly all criteria for image captioning. The encoder-decoder framework allows the model to convert an input image into a fixed-dimensional representation

(encoding) and then generate a sequence of words (decoding) to describe the image. This architecture has been enhanced by the introduction of attention mechanisms, which enable the model to focus on specific parts of the image when generating each word of the caption. This attention mechanism mimics human visual attention and significantly improves the quality of the generated captions.

Generative adversarial networks (GANs) and autoencoders have also shown promising results in image annotation tasks. GANs, through their adversarial training process, can generate high-quality images that are indistinguishable from real images, which helps in producing accurate and detailed captions. Autoencoders, which learn efficient representations of data, can also be employed to generate succinct and relevant image captions.

Reinforcement learning techniques have been applied to the sequence generation aspect of image captioning. These techniques involve training the model to generate sequences that maximize a reward function, which can be designed to prioritize the accuracy and relevance of the captions. Reinforcement learning approaches have been particularly effective in generating captions that not only describe the content of the image accurately but also do so in a linguistically fluent manner.

In summary, the evolution from traditional machine learning methods to advanced deep learning models has significantly enhanced the field of image captioning. The use of CNNs, attention mechanisms, GANs, and reinforcement learning techniques has enabled the development of models that can generate highly accurate and contextually appropriate captions for images. As computational resources continue to improve and more sophisticated algorithms are developed, the performance and applicability of image captioning systems are expected to advance further.

Chapter 8

Gantt Chart

	1st Evaluation	2nd Evaluation	3rd Evaluation	Final Evaluation
DATE	18/10/2023	14/12/2023	01/04/2024	20/04/24
Synopsis				
Project Finalization ppt				
Report				
60-70% Workable Project				
100% Workable Project				
Final Report				

Figure 8.1: Timeline Graph

The Gantt Chart for our final year project on image caption generation is structured around four evaluation phases. During the 1st Evaluation period, we will complete and submit the project synopsis. By the 2nd Evaluation, we will prepare and present a detailed PowerPoint presentation showcasing our progress. For the 3rd Evaluation, we will write and submit a comprehensive project report that includes our research, system design, and initial findings. Finally, during the Final Evaluation phase, we will compile and submit the final report, incorporating feedback and any additional insights. This chart ensures a clear and organized timeline for each task, leading to the successful completion of our project.

Chapter 9

Future Scope

9.1 Proposed Steps for Future Work

1. User-Centric Refinements: Conduct an in-depth analysis of user feedback to understand specific preferences and refine the model iteratively. This user-centric approach aims to enhance the overall user experience and meet diverse image-captioning requirements.

2. Transfer Learning Exploration: Explore transfer learning techniques to leverage pre-trained models effectively. Fine-tuning existing models on domain-specific data will extract more nuanced features, improving adaptability to a broader range of images.

3. Multilingual Capabilities: Extend language support beyond English to make the Image Caption Generator more globally accessible. Integrating multilingual capabilities aligns with the goal of creating a more inclusive tool for diverse linguistic communities.

4. Collaborative Engagement: Foster collaboration with the research community by sharing insights, models, and datasets. Open-sourcing aspects of the project contributes to collective progress, encouraging a continuous exchange of knowledge and advancements.

5. Continuous Monitoring of Advancements: Stay abreast of advancements in image captioning and natural language processing. Regularly monitor the literature and integrate cutting-edge techniques to ensure the project remains at the forefront of innovation.

9.2 Plans for Further Data Collection or Analysis

The Image Caption Generator project is set to be refined by incorporating the MS-COCO dataset, a vast collection of images with detailed annotations, to improve its ability to generate accurate captions. The project also plans to explore various neural network architectures, including the Exception, Inception, and ExpansionNet models. The Exception model offers improved performance through attention mechanisms, while the Inception model extracts intricate features from images. The ExpansionNet model is designed for efficient image recognition. The project also plans to incorporate advanced pre-trained models like BERT for a more nuanced understanding of contextual information. This strategic expansion aims to enhance image caption accuracy and explore the frontiers of state-of-the-art models in image understanding and natural language processing. Future endeavors will involve enriching the dataset with MS-COCO and exploring advanced model architectures to push the boundaries of the project.

Chapter 10

Snapshots

Figure 10.1: Testcase - 1

Actual Caption: "Two dogs playing together on a grassy field."
Generated Caption: "Two dogs are running and playing on the grass."
This image shows two dogs having fun together in an open grassy area. The AI-generated caption accurately captures the essence of the scene by describing the playful interaction between the dogs on the grass.

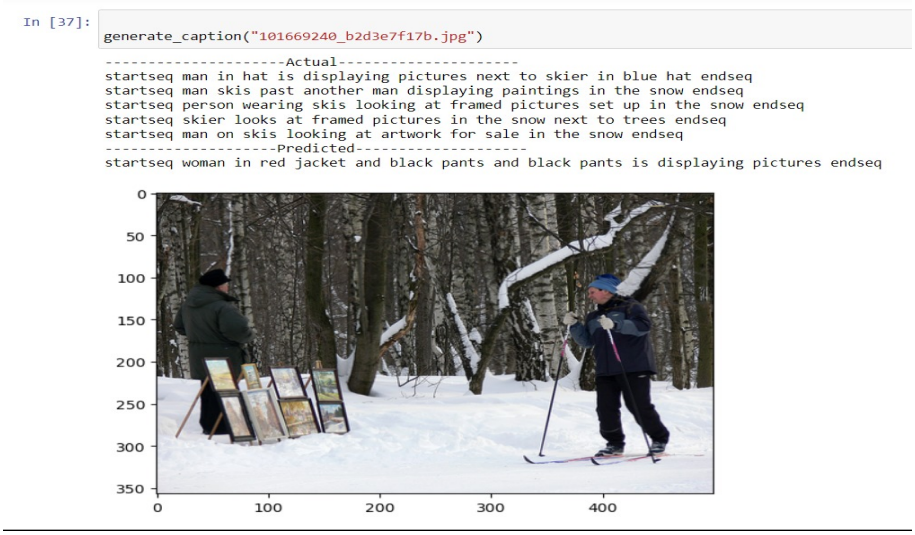


Figure 10.2: Testcase - 2

Actual Caption: "Two men skiing down a snowy slope."
Generated Caption: "Two men are skiing on a snow-covered mountain."
This image depicts two men enjoying skiing on a snowy hillside. The AI-generated caption effectively describes the activity and setting, highlighting the men skiing on the snow-covered terrain.

Chapter 11

Conclusion

This review study has explored the field of picture caption creation and highlighted the effectiveness of a hybrid CNN-LSTM model. Modern developments in the fields of image captioning and feature extraction have made it possible to examine important methods in detail and reveal their features and efficacy simultaneously. In this survey, methods that have produced outstanding results were explained, highlighting their function in effectively extracting, recognizing, and localizing objects in the image data. Simultaneously, the complex feature extraction procedure and its smooth conversion into a language model for picture captioning have been examined. A key accomplishment in this investigation is the incorporation of a CNN-LSTM hybrid model, which illustrates the cooperative relationship between convolutional neural networks and long short-term memory networks. This combination makes visual content easier to understand and makes it possible to create subtitles that make sense within their context. The results demonstrate how far natural language comprehension and computer vision have come, and how the hybrid approach has aided in the improvement of captioning performance. The survey results provide valuable insights for future research attempts in the dynamic convergence of computer vision and language modeling. This will help shape innovation in picture captioning as the field develops.

Bibliography

[1] Shuang Bai and Shan An. A survey on automatic image caption generation. *Neurocomputing*, 311:291–304, 2018.

[2] Shan Cao, Gaoyun An, Zhenxing Zheng, and Zhiyong Wang. Vision-enhanced and consensus-aware transformer for image captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):7005–7018, 2022.

[3] Ahmed Elhagry and Karima Kadaoui. A thorough review on recent deep learning methodologies for image captioning. *arXiv preprint arXiv:2107.13114*, 2021.

[4] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4125–4134, 2019.

[5] G Geetha, T Kirthigadevi, G Godwin Ponsam, T Karthik, and M Safa. Image captioning using deep convolutional neural networks (cnns). In *Journal of Physics: Conference Series*, volume 1712, page 012015. IOP Publishing, 2020.

[6] Taraneh Ghandi, Hamidreza Pourreza, and Hamidreza Mahyar. Deep learning approaches on image captioning: A review. *ACM Computing Surveys*, 56(3):1–39, 2023.

[7] Ansar Hani, Najiba Tagougui, and Monji Kherallah. Image caption generation using a deep architecture. In *2019 International Arab Conference on Information Technology (ACIT)*, pages 246–251. IEEE, 2019.

[8] Xiaodong He and Li Deng. Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine*, 34(6):109–116, 2017.

[9] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

[10] Licheng Jiao and Jin Zhao. A survey on the new generation of deep learning in image processing. *Ieee Access*, 7:172231–172263, 2019.

[11] Xiaoxiao Liu, Qingyang Xu, and Ning Wang. A survey on deep neural network-based image captioning. *The Visual Computer*, 35(3):445–470, 2019.

[12] Yue Ming, Nannan Hu, Chunxiao Fan, Fan Feng, Jiangwan Zhou, and Hui Yu. Visuals to text: A comprehensive review on automatic image captioning. *IEEE/CAA Journal of Automatica Sinica*, 9(8):1339–1365, 2022.

[13] Ariyo Oluwasammi, Muhammad Umar Aftab, Zhiguang Qin, Son Tung Ngo, Thang Van Doan, Son Ba Nguyen, Son Hoang Nguyen, and Giang Hoang Nguyen. Features to text: a comprehensive survey of deep learning on semantic segmentation and image captioning. *Complexity*, 2021:1–19, 2021.

[14] Uddagiri Sirisha and Bolem Sai Chandana. Semantic interdisciplinary evaluation of image captioning models. *Cogent Engineering*, 9(1):2104333, 2022.

[15] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559, 2022.

[16] Marc Tanti, Albert Gatt, and Kenneth P Camilleri. Where to put the image in an image caption generator. *Natural Language Engineering*, 24(3):467–489, 2018.

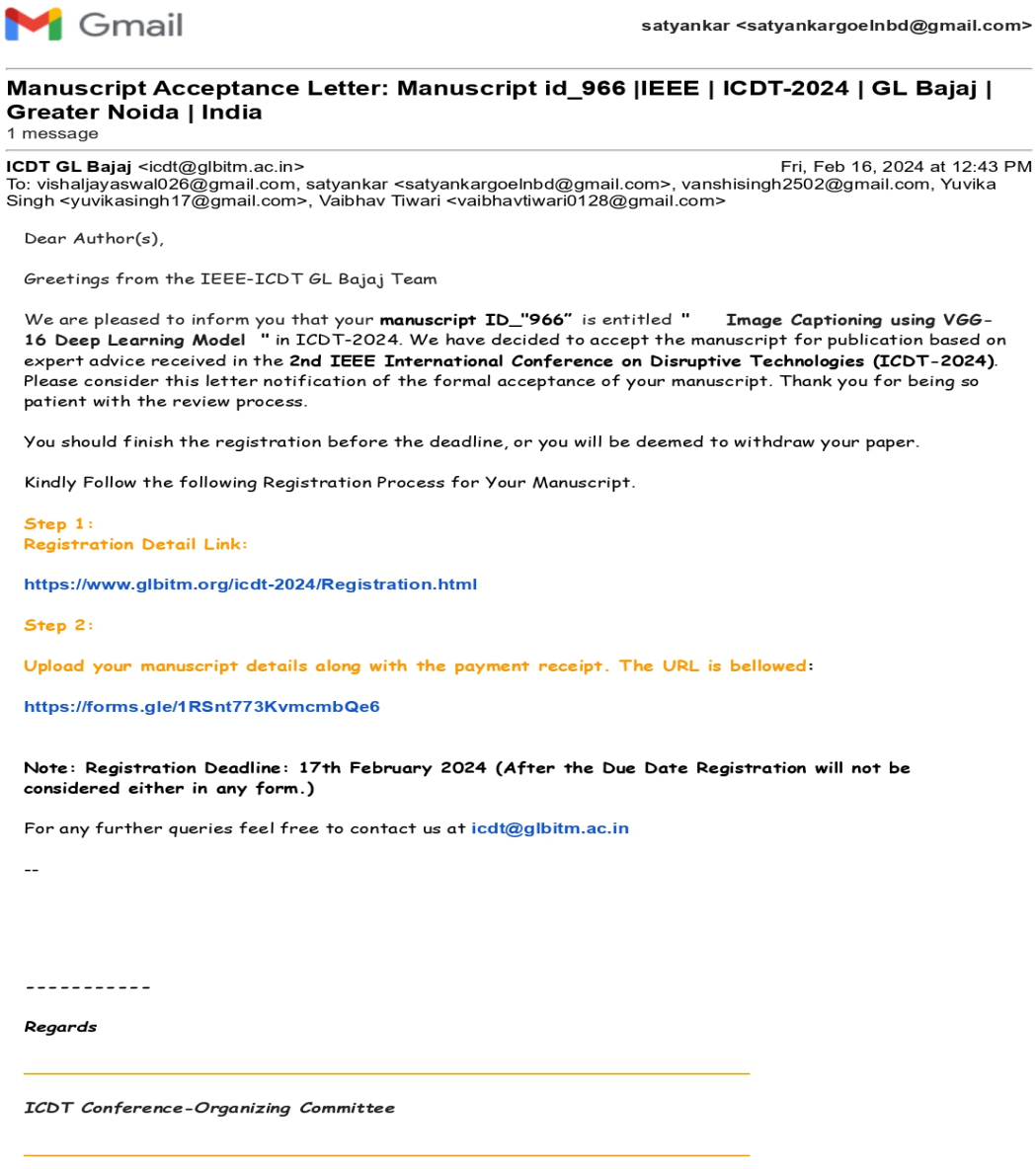
[17] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022.

[18] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018.

[19] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

Appendix - A

Acceptance Mail



Gmail satyankar <satyankargoelnbd@gmail.com>

Manuscript Acceptance Letter: Manuscript id_966 |IEEE | ICDT-2024 | GL Bajaj | Greater Noida | India

1 message

ICDT GL Bajaj <icdt@glbitm.ac.in> Fri, Feb 16, 2024 at 12:43 PM
To: vishaljayaswal026@gmail.com, satyankar <satyankargoelnbd@gmail.com>, vanshisingh2502@gmail.com, Yuvika Singh <yuvikasingh17@gmail.com>, Vaibhav Tiwari <vaibhavtiwari0128@gmail.com>

Dear Author(s),

Greetings from the IEEE-ICDT GL Bajaj Team

We are pleased to inform you that your **manuscript ID_"966"** is entitled **" Image Captioning using VGG-16 Deep Learning Model "** in ICDT-2024. We have decided to accept the manuscript for publication based on expert advice received in the **2nd IEEE International Conference on Disruptive Technologies (ICDT-2024)**. Please consider this letter notification of the formal acceptance of your manuscript. Thank you for being so patient with the review process.

You should finish the registration before the deadline, or you will be deemed to withdraw your paper.

Kindly Follow the following Registration Process for Your Manuscript.

Step 1:
Registration Detail Link:

https://www.glbitm.org/icdt-2024/Registration.html

Step 2:

Upload your manuscript details along with the payment receipt. The URL is bellowed:

https://forms.gle/1RSnt773KvmcmbQe6

Note: Registration Deadline: 17th February 2024 (After the Due Date Registration will not be considered either in any form.)

For any further queries feel free to contact us at **icdt@glbitm.ac.in**

--

Regards

ICDT Conference-Organizing Committee

Figure 11.1:

Presenter Certificate

Figure 11.2:

Published in: 2024 2nd International Conference on Disruptive Technologies (ICDT)
Date of Conference: 15-16 March 2024
Date Added to IEEE Xplore: 11 April 2024 ; **Publisher**: IEEE

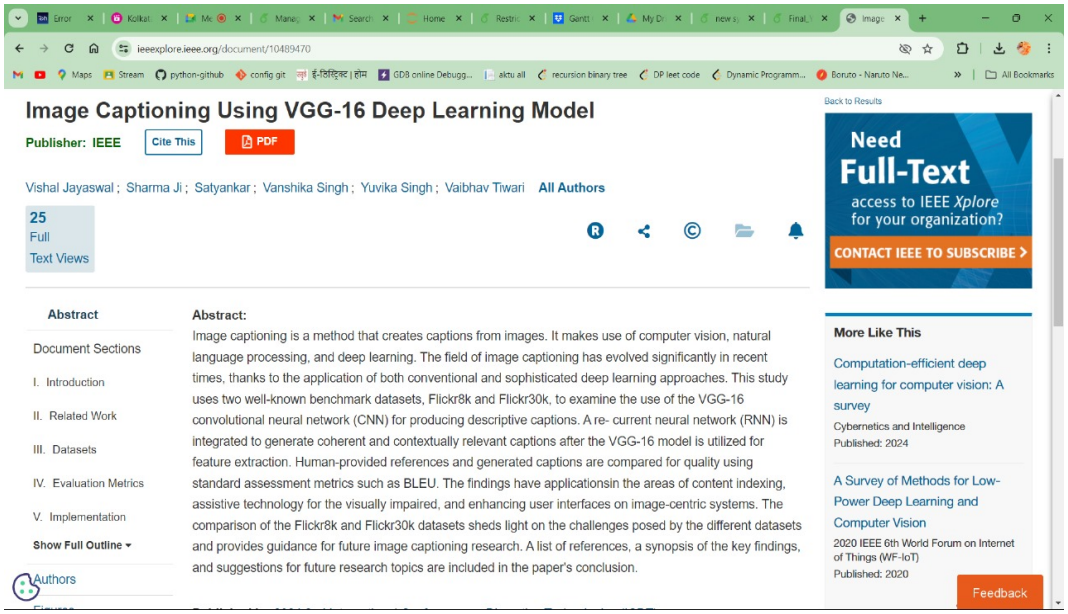


Figure 11.3:

Appendix - B

Curriculum Vitae of Vanshika Singh

Vanshika Singh

+91 7302943270
vanshisingh2502@gmail.com

Linkedin: https://www.linkedin.co/

Passionate individual with great interpersonal and communication skills. Ability to work in C,C++ and Python.

PROJECTS

Blockchain transaction Interface (using solidity)

It can perform bitcoin transactions using ether.

Text to audio converter (using Python)

Converts the text written by us in audio(mp3), which can also be saved in any language using python.

INTERNSHIP

Codsoft

JULY 15,2023 - AUGUST 15, 2023

C++ Programming Intern

Proactively learned methodologies to improve programming skills and stay updated with industry trends.

EDUCATION

St. Mary's Academy

2020

Intermediate
95.6%

Ajay Kumar Garg Engineering College

2020-2024

BTech(currently pursuing)

8.6 sgpa(recent)

SKILLS

- **Programming skills:** C, C++, Python, Solidity.
- **Database Management** SQL
- **Content Writing**

ACHIEVEMENTS

- **Gate Qualified** with a score of **440**.
- **Cocubes** Score **650**.
- 200+ Questions on various coding platforms
- **Ronin** badge (expertise)in Dynamic Programming.

Figure 11.4:

Appendix - C

Curriculum Vitae of Satyankar

SATYANKAR

Ghaziabad, India | 9084622400
satyankargoelnbd@gmail.com | linkedin.com/in/satyankar-38451521b

INTERNSHIP

Django Developer *January 2024 - Till now*
SenRa Tech. Pvt. Ltd. New Delhi, India

- Developed APIs with efficient logic implementation regarding GIS, achieving an average response time improvement of 50% compared to previous implementations.
- Utilized techniques such as caching, asynchronous processing, and query optimization, resulting in a 30% reduction in API response times and a 20% decrease in server resource utilization.
- Developed JavaScript functions to decode sensor data from JSON format, handling various sensor types such as temperature, humidity, etc.

PHP Developer *April 2023 - July 2023*
Darx Technologies Greater Noida, India

- Got this wonderful opportunity to assist the software developer team at their organization.
- Experienced the design and maintenance procedure of scalable PHP applications.
- Ensure the reliability of the user base for a significant number of users.
- Embraced a hands-on experience with various technologies such as PHPMyAdmin, MySql, FileZilla, etc

PERSONAL PROJECTS

Quiz Portal | Django

- *A general quiz portal where the user can add or delete question as per requirement.*
- *The answer will be stored for different users.*
- *Cumulative result will be displayed on a leaderboard.*

Online Shopping Portal | *Django*

- *Back-end cloning of a shopping website like amazon, etc.*
- *Proper user and customer authentication system*

Blogging Site | *Core PHP*

- *Tried to make a blogging site where a blogger can post their blogs and all see blogs of other bloggers.*
- *Also a reader type user which can only read and comment on those blogs.*

Library Management System | *Core PHP*

- *I have made LBMS , with keeping in my mind the needs of librarian and customer.*
- *Tried to cover almost all features , that must be needed.*
- *It covers almost all aspects of a LBMS and to handle multiple requests.*

EDUCATION

Ajay Kumar Garg Engineering College *Expected June 2024*
CSE , B.Tech. , SGPA : 8.1 *Ghaziabad, India*

Green View Public School *March 2020*
12th , PCM , 88.6% *Delhi*

R.R. Morarka Public School *March 2018*
10th , 96.6% *Najibabad, U.P.*

Figure 11.5:

Achievements

- *Technical Society (BRL) : Head Coordinator*
- *12th : Inter-house Debate Competition (1st position)*
- *10th : House Captain*

Skills

- *Laravel*
- *PHP*
- *Python*
- *Django*
- *MySQL*
- *HTML*
- *Microsoft Office*
- *phpmyadmin*
- *Communication*

Figure 11.6:

Appendix - D

Curriculum Vitae of Yuvika Singh

Yuvika Singh

Quick Learner, Problem Solver

+918736045713
yuvikasingh17@gmail.com

EDUCATION -

B.Tech in CSE Ongoing

Ajay Kumar Garg Engineering College
2020 - 2024

Intermediate 88%

Jagran Public School
2019

EXPERIENCE -

CodSoft
C++ Programming Intern July 2023 - August 2023

- Designed and built reliable C++ codes.
- Worked on proficiency and efficiency of my codes through practical experience.

PROJECTS -

Personal Portfolio
https://yuvika-portfolio.netlify.app/

- Created a portfolio using full stack technologies.
- Contains information about my skills, projects and contact info.
- Technologies Used - **HTML, CSS, ReactJS, Node, Express.**

SkyPeek – A Weather App
https://skypeek.netlify.app/

- Created an app that displays current weather conditions,as well as date and time for various cities worldwide.
- Other technologies used - **HTML, CSS, JS, ReactJS.**

Readbout – A News App

- Developed an application that swiftly delivers the most up-to-date global news, categorized by genre, within a matter of seconds.
- Used News API for latest news.
- Technologies Used - **HTML, CSS, JS, ReactJS.**

SKILLS -

C/C++, HTML, CSS, JavaScript, Bootstrap, React, Express, Git/Github, SQL

CERTIFICATES -

The Complete Web Development Bootcamp – 2023 | *View*

Programming Foundations with JS, HTML and CSS – by Duke University | *View*

ACHIEVEMENTS -

- **GATE Qualified** (AIR 3101)
- Solved over 400 coding problems all across multiple platforms.
- Codechef May Long One 2022 – Global Rank 538

COURSEWORK -

- Data Structure and Algorithms
- Object-Oriented Programming
- Operating Systems
- Database Management Systems

PROFILE LINKS -

- **LinkedIn** – *Yuvika Singh*
- **Github** – *yuvika09*
- **Leetcode** – *yuvika_09*
- **Codeforces** – *yuvi_2002*
- **Codechef** – *yuvika_092*
- **Hackerrank** – *yuvika_09*

Figure 11.7:

Appendix - E

Curriculum Vitae of Vaibhav Tiwari

Vaibhav Tiwari
Github: github.com/vaibhavtiwari0112
LinkedIn: linkedin.com/in/vaibhav-tiwari

Email: vaibhavtiwari0128@gmail.com
Mobile: +91 8957603823
D.O.B.: 10 October 2002

Experience

- **Efficient Corporates** — Remote
  React.js Developer Intern — *September 2023 - December 2023*
  - **Work**:
    - Architected web application pages, managing live API data integration for real-time insights on customer behavior, boosting e-commerce revenue by 25%, and enhancing user experience.
    - Designed and deployed 10+ web pages, integrating real-time data from APIs to fuel decision-making processes and increase user engagement by 40% within the E-Commerce Dashboard UI.
  - **Project**: **E-Commerce Dashboard**
    - Utilized technologies including React.js, BootStrap 4, Material-Ui in the development process.
    - Implemented a user registration and authentication system, enhancing access management by 30%.
    - Devised a user-friendly login interface and integrated data visualization tools, boosting sales data presentation efficiency by 40%.
    - Added a download feature, improving data accessibility by 25%.
    - Enhanced user experience with additional features, resulting in a 20% increase in user satisfaction.
    - Optimized system efficiency, achieving a 15% improvement in overall performance.

Projects

- **AmazingE-state (Link)**:
  - Engineered "AmazingE-state," a MERN stack real estate listing platform featuring Firebase authentication and Google login integration.
  - Leveraged MongoDB for efficient data management, ensuring seamless storage and retrieval of property listings.
  - Integrated Google login to offer secure and convenient user authentication, resulting in a 35% boost in engagement.
  - Designed an intuitive UI using React.js, Tailwind CSS, and JavaScript, leading to a 25% increase in user satisfaction.
  - Employed Express.js to develop RESTful APIs, enabling seamless communication between the frontend and backend.
- **blogs-0128 (Link)**:
  - Developed the "blogs-0128" platform with React.js, Firebase, CSS, Netlify, and Firestore.
  - Implemented smooth sign-in and login functionalities, resulting in a 25% increase in user engagement.
  - Enhanced user interaction with multimedia content, leading to a 40% uptick in engagement.

Skills Summary

- **Programming Languages**: C, C++, Node.js, Redux , Python, HTML, CSS, SQL, JavaScript
- **Frameworks, Libraries, and Tools**: React.js, Node.js, MongoDB, Express, Firebase, VSCode ,Github
- **Analytical**: Solved over 300 problems on platforms like GeeksforGeeks and LeetCode using C++.
- **Soft Skills**: Leadership, Communication, Public Speaking

Education

- **Ajay Kumar Garg Engineering College** — Uttar Pradesh, India
  Bachelor of Technology - Computer Science and Engineering; Current CGPA: 7.6 — *Nov 2020 - Present*
- **Relevant Coursework:**: Studied a comprehensive curriculum including Data Structures And Algorithms , Operating System, Database Management System, Cloud Computing, Computer Networks , and Object-Oriented Programming.

Certificates and Achievements

- Recipient of Best UI/UX Design Award at OpenHacks Hackathon
- Active Participant in Code4Youth Hackathon: (Certificate)
- Qualified in GATE - 2023
- Completed Industrial Training in Machine Learning at Antrix Academy - 2021: (Certificate)
- Awarded NCC Grade A Certificate - 2015

Figure 11.8:

Image Captioning using VGG-16 Deep Learning Model

Vishal Jayaswal
Department of CSE
Ajay Kumar Garg Engineering College
Ghaziabad, India
vishaljayaswal026@gmail.com

Sharma Ji
Department of CSE
Ajay Kumar Garg Engineering College
Ghaziabad, India saurabh12d@gmail.com

Satyankar
Department of CSE
Ajay Kumar Garg Engineering College
Ghaziabad, India
satyankargoelnbd@gmail.com

Vanshika Singh
Department of CSE
Ajay Kumar Garg Engineering College
Ghaziabad, India
vanshisingh2502@gmail.com

Yuvika Singh
Department of CSE
Ajay Kumar Garg Engineering College
Ghaziabad, India
yuvikasingh17@gmail.com

Vaibhav Tiwari
Department of CSE
Ajay Kumar Garg Engineering College
Ghaziabad, India
vaibhavtiwari0128@gmail.com

***Abstract*—Image captioning is a method that creates captions from images. It makes use of computer vision, natural language processing, and deep learning. The field of image captioning has evolved significantly in recent times, thanks to the application of both conventional and sophisticated deep learning approaches. This study uses two well-known benchmark datasets, Flickr8k and Flickr30k, to examine the use of the VGG-16 convolutional neural network (CNN) for producing descriptive captions. A re- current neural network (RNN) is integrated to generate coherent and contextually relevant captions after the VGG-16 model is utilized for feature extraction. Human-provided references and generated captions are compared for quality using standard assessment metrics such as BLEU. The findings have applicationsin the areas of content indexing, assistive technology for the visually impaired, and enhancing user interfaces on image-centric systems. The comparison of the Flickr8k and Flickr30k datasets sheds light on the challenges posed by the different datasets and provides guidance for future image captioning research. A list of references, a synopsis of the key findings, and suggestions for future research topics are included in the paper's conclusion.**

***Index Terms*—Deep learning, Object Hallucination, Natural Language Processing, Computer Vision.**

I. Introduction

The increasing availability of optical perception data has led to its significant utility in various fields such as quality assurance, medical diagnostics, surveillance, autonomous vehicles, facial recognition, forensic investigations, biometrics, and 3D reconstruction. Image captioning, a developing subject at the intersection of natural language processing and computer vision, aims to create text-based descriptions that align linguistically [12], maintain syntactical accuracy, and adhere to semantic relevance. This technology can beused for improving content retrieval systems and helping visually impaired people. This research study explores thecomplexities of image captioning, providing a mathematical definition of the problem and addressing inherent difficulties such as maintaining variety in image content perception and producing cohesive captions [8]. It examines well-known datasets, preprocessing stages, and training procedures, including loss functions and optimization methods. The study also covers assessment metrics, focusing on the significance of both quantitative and qualitative evaluations. The study contrasts conventional methods with recent developments in deep learning to provide a comprehensive understanding of the field.

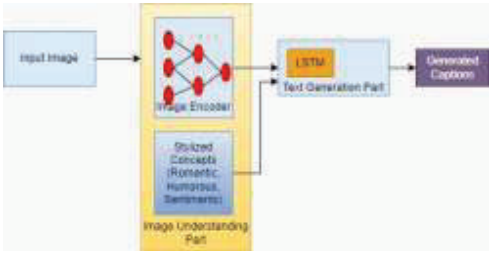


Fig. 1. Block Diagram representing Image Captioning process

In fig 1, we are explaining the process behind the model. Process includes steps i.e. image features are obtained using convolutional neural network, then attributes are extracted from visual features and multiple captions are created by language model and at last captions that are generated.

The techniques used for obtaining image features can be broadly divided into two categories:

(1) Traditional machine-learning based techniques (2) Deep machine-learning based techniques.

This study explores different methods for creating image captions, including template-based, retrieval-based, and creative approaches. Retrieval-based methods draw captions from pre-existing ones, while template-based methods use predetermined templates. Deep machine learning techniques improve semantic accuracy in each image. Learning methodologies like supervised and reinforcement learning can be used. Encoder-

2024 2nd International Conference on Disruptive Technologies (ICDT) | 979-8-3503-7105-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDT61202.2024.10489470

decoder or compositional architectures can be used for entire scenes or specific areas. Techniques use attention mechanisms, semantic ideas, and image-description styles.

II. Related Work

This study explores different methods for creating im- age captions, including template-based, retrieval-based, and creative approaches. Retrieval-based methods draw captions from pre-existing ones, while template-based methods use predetermined templates. Deep machine learning techniques can produce original captions from visual and multimodal [7] domains, improving semantic accuracy. Learning methodologies like supervised learning and reinforcement learning can be used. Encoder-decoder or compositional architectures can be used for entire scenes or specific areas of an image. Techniques use attention mechanisms, semantic ideas, and different image-description styles. [13]

A. Traditional Approaches

Prior to the invention of artificial neural networks (ANNs) [15], traditional segmentation and semantic segmentation techniques were primarily unsupervised. Traditional semantic segmentation techniques require less data and take less time to compute. The methods utilized by early researchers were template-based and retrieval-based to allow computers with visual learning abilities to generate coherent and understandable language for images.

Retrieval-based image captioning stores numerous image-caption pairs in a corpus using similarity comparison to find images similar to the input image. [6] Computers use annotations from previously obtained photos to characterize a given image, using the Tree-F1 rule and the nearest neighbor rule. Global similarity is calculated, and the caption of the matching image is sent.

Template-based image captioning is a model used in early projects, creating captions in a limited way, both syntactically and semantically. It involves using fixed templates with fixed blank spots to recognize the properties, features, actions, and surrounds of objects before filling them in. Some strategies include filling in blanks using a triplet, object identification techniques, and pretrained language models. [18]

Unlike retrieval models, template-based models produce descriptions more relevant to images but employ predefined templates and cannot produce captions of varying length, making the automated descriptions less organic compared to handwritten captions by humans.

B. Neural Network-Based Captioning

Comparing deep learning to other machine learning techniques and algorithms, image captioning has greatly improved. These techniques require a substantial quantity of training data, comprising images and caption labels, and are mostly used in supervised environments. These astounding results have been attained by applying a variety of models, such as autoencoders ,generative adversarial networks (GAN) [3], convolutional neural networks (CNN), recurrent neural networks [10], [2], artificial neural networks (ANN), and combinations of these.

1) Encoder-decoder framework: CNNs help anticipate words by assisting with object identification, localization, and interactions in conjunction with long-term dependencies from a recurrent network cell. Meaningful textual representations and descriptions are provided by a variety of CNN-based encoders; a novel recurrent fusion network (RFNet) [10] was presented. The sentiment of the image was labeled using a generative adversarial network (GAN) trained on a binary vector. [8]

- **Encoder:** Using the pre-trained VGG-16 convolutional neural network (CNN) as a feature extractor is the first step in the encoding process for picture captioning using VGG-16. Obtaining the VGG-16 model's pre-trained weights is the first step; this model is usually trained on extensive image classification datasets such as ImageNet [8].
- **Decoder:** The vanishing gradient problem is addressed with the Long Short-Term Memory (LSTM) [18] recurrent neural network architecture, which is utilized in sequence processing, mainly picture captioning. It captures and maintains long-term relationships in sequential data, using input images as input and predicting the probability distribution of subsequent words at each decoding step. The process iteratively continues until an end token is produced or the maximum sequence length is achieved.

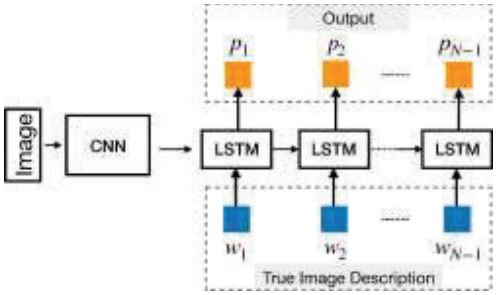


Fig. 2. LSTM as a decoder

In Fig 2 , we are defining LSTM model as decoder , in this an image is given as input to CNN , output of CNNis sent to LSTM which generates output sequences whichare further used to generate captions.

C. Deep Learning

Multi-layered neural networks are used in deep learning, a subfield of machine learning, to learn hierarchical data representations. It has significantly advanced picture captioning, with CNNs extracting features from images and identifying patterns and spatial hierarchies. Recurrent Neural Networks (RNNs) and Transformer models are used to produce logical and relevant captions. RNNs allow for sequential dependencies in language, while transformer models understand long-range dependencies and are popular for parallelization [14]. The deep learning framework simplifies end-to-end training, improving image captioning systems' performance and allowing models

to generalize well to various unknown inputs. This combination of deep learning and picture captioning has led to more sophisticated and context-aware automatic description creation, with applications in image indexing, retrieval systems, and content accessibility. [5]

$$a_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \quad (1)$$

$$e_{ij} = f(S_{i-1}, h_i) \quad (2)$$

III. DATASETS

For deep learning to fully understand patterns and optimize the number of parameters required for gradient convergence [12], a large volume of training data is required.To facilitate evaluation and testing, it is standard procedure to partition the dataset into training, validation, and testing sets. Because of this separation, the model can be trained on one subset of the data, validated on another to adjust hyperparameters and track performance, and then tested on an entirely different subset to evaluate its generalization abilities. In order to prepare a solid and organized dataset for the ensuing training of picture captioning models, it is imperative that the preprocessing and data loading processes are carried out carefully. As a result, the following datasets are readily available and were created especially for the task of semantic segmentation: [12]

A. Flickr8k

Flickr8k [14] is a crucial dataset in computer vision and natural language processing, containing 8,000 unique photographs labeled with five human-generated descriptions per image. It is essential for training and testing image captioning algorithms, teaching models the complex relationships between written descriptions and visual information. After training, models undergo a thorough evaluation process to produce semantically meaningful captions similar to human references. Flickr8k is compact, allowing easy training on budget laptops or desktop computers. It includes 82 JPEG photos, with 6000 used for development, 1000 for testing, and 1000 for training. The dataset contains 40460 captions, five for each image. [10]

B. Flickr30k

Flickr30k is a massive dataset of 31,000 photos from Flickr, each with five human-created captions, aimed at improving image captioning research. It contains 31,783 photos of people in their daily activities, each with a caption that describes it. The dataset fills the need for larger training and assessment sets, enabling the development of more reliable and scalable image captioning algorithms. [5]

C. MS COCO

To push the boundaries of computer vision, Microsoft Common Objects in Context was created using standard photos, annotation, and assessment.Bounding boxes or object annotated features are used in the dataset for the object recognition task. It has more than 800,000 pictures available for its training, test, and validation sets, more than 500,000 segmented object instances, and a total of 80 object categories. [5]

IV. EVALUATION METRICS

We assess the automatically generated captions to make sure they accurately describe the provided image. Some popular metrics for evaluating image captioning in machine learning are listed below. [12]

A. BLEU

(BiLingual Evaluation Understudy) The BLEU (Bilingual Evaluation Understudy) [10] score is a statistical tool used to evaluate the quality of machine-generated text, including image captions. It assesses the similarity of a generated caption to human-annotated reference captions. The process involves tokenization [16] of generated and reference captions into individual words or n-grams, calculation of precision, geometric mean of precision scores, and brevity penalty to correct biases favoring shorter captions. The shortness penalty is calculated by dividing the difference between the generated caption's length and the average length of reference captions, and multiplying the overall precision by the brevity penalty. This helps in determining the accuracy of captions. The BLEU score can be expressed mathematically as given in Table 1:

$$\text{BLEU} = \text{BP} \times \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log(\text{precision}_n)\right) \quad (3)$$

where BP is the brevity penalty, N is the maximum n-gram length considered, and precision-n is the precision for n- grams. This comprehensive formula captures both precision and brevity aspects, providing a numerical representation of the quality of machine-generated captions in image captioning tasks.

B. METEOR

(Metric for Evaluation of Translation with Explicit Ordering) It is based on a weighted F-score calculation and a penalty function intended to verify the candidate sequence's order, and it tackles the shortcoming of BLEU. For the purpose of identifying sentence similarities, it matches synonyms. [1]

V. IMPLEMENTATION

Fig 3 shows the flow diagram of image captioning process , in this an image dataset is given as input to CNN(encoder) for feature extraction , after that caption sequences are generated using LSTM (decoder) and then resulting captions are generated.

A. Preprocessing and Data Loading

The image captioning process involves preprocessing and data loading to convert unprocessed image and text data into a machine learning model. Preprocessing [17] involves cropping, normalization, and data augmentation techniques. Text preprocessing prepares written captions, while data loading organizes processed photos and captions into a training format. Batching speeds up the training process, while data shuffles

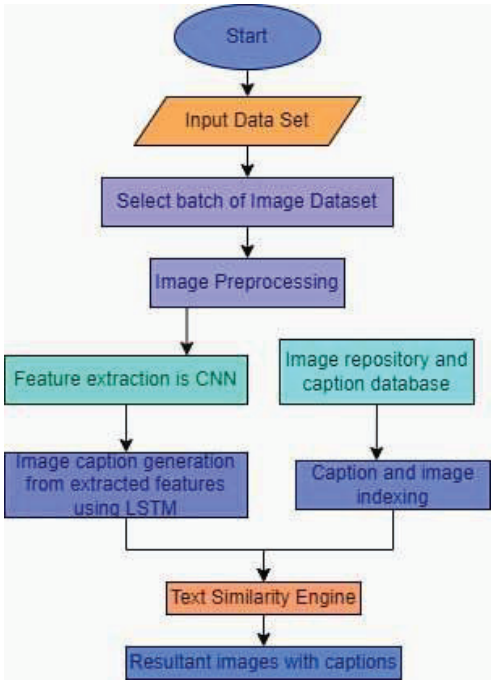


Fig. 3. Flow diagram of image captioning

prevent erroneous correlations. A data pipeline manages data loading and preprocessing on-the-fly during training.

- Load the Flickr8k dataset, which is made up of pictures with captions.
- Preprocessing: Adjust the photos' dimensions to a standard size and do any required normalization. Preprocess and tokenize the captions.

B. VGG-16 Feature Extraction

- Use a pre-trained VGG-16 model that was initially trained on ImageNet to extract high-level features from the images.
- The output of the final convolutional layer should be retained while the VGG-16 fully connected layers are removed. This output results in the visual representation.
- For the purpose of captioning photos, the VGG-16 model collects high-level information from the pictures. Rich representation of the input image is provided by the last convolutional layer of VGG-16, which preserves spatial information. Semantic aspects that include objects, forms, and patterns found in the image are recorded by VGG-16 [11]. These characteristics are essential for comprehending the visual context and producing insightful captions.

C. Model Architecture

- Construct an image captioning model architecture by fusing a language model for caption generation with the image attributes from VGG-16. [4]
- To turn the words in the captions into dense vectors, use an embedding layer.
- As the decoder, use an LSTM (Long Short-Term Memory) network to produce the captions' consecutive words.

D. Training with Flickr8k

The Flickr8k dataset, consisting of 8,000 photos with captions, is used to train a model for image captioning. Data preprocessing involves cleaning and tokenizing [13] captions, shrinking images, and normalizing pixel values. A Convolutional Neural Network extracts features, while tokenization and padding prepare captions for input. An encoder-decoder architecture with LSTM or GRU cells is used, with sequence-based loss functions and optimization approaches.

E. Refine the model using Flickr30k

- Adjust the model using the Flickr30k dataset. To do this, load the weights from the Flickr8k-trained model and carry on with the new dataset's training.
- By fine-tuning, the model may be made to work with a larger dataset and recognize a wider range of patterns in the image-caption pairs.

F. Retuning the Model on Flickr8k

- Adjust the model once more using the Flickr8k dataset. This step aims to improve generalization by maximizing the model's output on the original dataset. We can see Table II shows images and a comparison between reference caption of the dataset and predicted captions by our model. [9]

G. Testing and Evaluation

- Analyze the trained model's performance with metrics like BLEU, METEOR, etc., or on a different validation set.
- The accuracy of translations produced by machines is gauged by the BLEU (Bilingual Evaluation Understudy) score [8]. Based on word sequences (n-grams) that overlap between the proposed and reference translations, it determines precision. Brevity Penalty [5] takes into consideration translations that are shorter. The geometrical median of n-gram precisions, shortened for clarity, isthe BLEU score. Higher scores denote higher quality translation; scores range from 0 to 1.

H. Optimization and adjustments

- Model Complexity: Modify the LSTM decoder's and other model elements' complexity. A model that is too simple could have trouble capturing complicated relationships, whereas a model that is too complex could result in overfitting [3].
- Training epochs and Batch Size: Try different batch sizes and epoch counts. In order to prevent overfitting or underfitting, keep an eye on both training and validation performance across epochs.
- Assessment Measures: To obtain a more thorough grasp of caption quality, take into account utilizing additional

assessment metrics in addition to BLEU, such as METEOR, CIDEr, and ROUGE [5].

- Data Augmentation: Use methods such as picture flipping, rotation, and scaling to add more images to the training dataset. The resilience and generalization of a model can be strengthened through augmentation.
- Hyperparameter Search: Look for the best possible hyperparameters by methodically examining various architectural parameters, LSTM units [13], and embedding dimensions. Either a random or grid search may be used in this investigation.
- Error Analyzation: Analyze model errors in test or validation sets using error analysis. Recognize the different kinds of errors the model commits and modify the training plan accordingly.

VI. Results and Findings

A. Epoch vs Accuracy Graphs

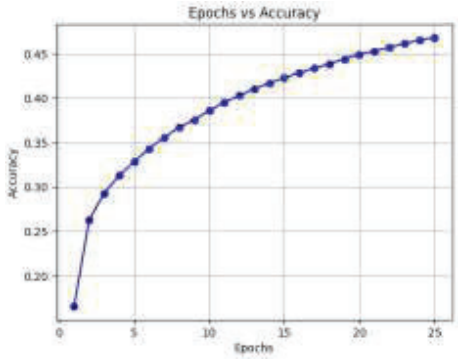


Fig. 4. For Flickr8k

Fig 4 shows a graph of Epochs vs Accuracy for Flickr8K dataset, graph shows an exponential increase in accuracy when model training is done for increased epochs values.Fig 5 shows a graph of Epochs vs Accuracy for Flickr8K on

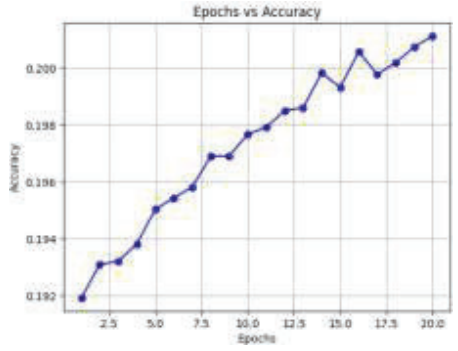


Fig. 5. For Flickr8k on Flickr30k

Flickr30k dataset , it shows a dynamic increase and decrease in the accuracy values on increasing epochs values for training of model.

TABLE Iv BLEU SCORES FOR DATASETS

Dataset	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Flickr8k	0.53	0.30	0.11	0.17
Flickr30k	0.53	0.27	0.16	0.08
Flickr8k trained on Flickr30k	0.56	0.34	0.22	0.13

TABLE II CAPTIONS MAPPING

Image Id	Image	Caption
1002674143_1b742ab4b8		- **Reference Caption:** A little girl covered inpaint sits in front of apainted rainbow withher hands inside bowl - **Predicted Caption:** two children enjoy thegrass with fingerpaintsin the background
101669240_b2d3e7f17b		- **Reference Caption:** Aman in hat is displayingpictures next to a skier in a blue hat - **Predicted Cap- tion:**Two people are the snow white in thesnow
1001773457_577c3a7d70		- **Reference Caption:** A black dog and a spotted dog are fighting - **Predicted Cap- tion:**Two dogs are playing in the grass
1096395242_fc69f0ae5a		- **Reference Caption:** A boy with a toy gunpointed at the camera - **Predicted Caption:**Young girl in the camera

VII. Discussion

The results obtained from algorithms for deep learning and machine learning and backbones differ according on how well they can acquire mappings from image inputs to labels. CNN-based approaches are the most efficient in tasks involving images, although they can be computationally ex- pensive. Traditional machine learning algorithms like Markov random field, random forest, ensemble modeling ,naive Bayes, and support vector machines (SVMs)are overly basic and mainly dependent on handcrafted feature engineering or do- main feature expertise. Clustering algorithms like K meansand fuzzy C-means are not particularly successful when there are several boundaries. CNN is useful for spatial data, object detection, and localization due to its invariant property. Supervised learning approaches remain a popular technique, but

data augmentation and generative adversarial networks have played significant roles in image captioning. Deep learning models have yielded the best performance across nearly all criteria, with the encoder-decoder architecture being the most common implementation. Attention mechanism concepts have been applied to improve feature computation, while generative adversarial networks and Autoencoders have been doing agreat job of producing succinct image annotation. Reinforced learning techniques have also generated sequences that succinctly describe images in a timely manner.

VIII. CONCLUSION

This review study has explored the field of picture caption creation and highlighted the effectiveness of a hybrid CNN-LSTM model. Modern developments in the fields of image captioning and feature extraction have made it possible to examine important methods in detail and reveal their features and efficacy simultaneously. In this survey, methods that have produced outstanding results were explained, highlighting their function in effectively extracting, recognizing, and localizing objects in the image data. Simultaneously, the complex feature extraction procedure and its smooth conversion into a language model for picture captioning have been examined. A key accomplishment in this investigation is the incorporation of a CNN-LSTM hybrid model, which illustrates the cooperative relationship between convolutional neural networks and long short-term memory networks. This combination makes visual content easier to understand and makes it possible to create subtitles that make sense within their context. The results demonstrate how far natural language comprehension and computer vision have come, and how the hybrid approach has aided in the improvement of captioning performance. The survey results provide valuable insights for future research attempts in the dynamic convergence of computer vision and language modeling. This will help shape innovation in picture captioning as the field develops.

REFERENCES

[1] Tushar Aggarwal. *Image Descriptive Summarization by Deep Learning and Advanced LSTM Model Architecture.* PhD thesis, 2019.

[2] Shuang Bai and Shan An. A survey on automatic image caption generation. *Neurocomputing*, 311:291–304, 2018.

[3] Ahmed Elhagry and Karima Kadaoui. A thorough review on recent deep learning methodologies for image captioning. *arXiv preprint arXiv:2107.13114*, 2021.

[4] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4125–4134, 2019.

[5] Taraneh Ghandi, Hamidreza Pourreza, and Hamidreza Mahyar. Deep learning approaches on image captioning: A review. *ACM Computing Surveys*, 56(3):1–39, 2023.

[6] Ansar Hani, Najiba Tagougui, and Monji Kherallah. Image caption generation using a deep architecture. In *2019 International Arab Conference on Information Technology (ACIT)*, pages 246–251. IEEE, 2019.

[7] Xiaodong He and Li Deng. Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine*, 34(6):109–116, 2017.

[8] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

[9] Licheng Jiao and Jin Zhao. A survey on the new generation of deep learning in image processing. *Ieee Access*, 7:172231–172263, 2019.

[10] Xiaoxiao Liu, Qingyang Xu, and Ning Wang. A survey on deep neural network-based image captioning. *The Visual Computer*, 35(3):445–470, 2019.

[11] Yue Ming, Nannan Hu, Chunxiao Fan, Fan Feng, Jiangwan Zhou, and Hui Yu. Visuals to text: A comprehensive review on automatic image captioning. *IEEE/CAA Journal of Automatica Sinica*, 9(8):1339–1365, 2022.

[12] Ariyo Oluwasammi, Muhammad Umar Aftab, Zhiguang Qin, Son Tung Ngo, Thang Van Doan, Son Ba Nguyen, Son Hoang Nguyen, and Giang Hoang Nguyen. Features to text: a comprehensive survey of deep learning on semantic segmentation and image captioning. *Complexity*, 2021:1–19, 2021.

[13] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559, 2022.

[14] Marc Tanti, Albert Gatt, and Kenneth P Camilleri. Where to put the image in an image caption generator. *Natural Language Engineering*, 24(3):467–489, 2018.

[15] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022.

[16] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018.

[17] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[18] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):1–38, 2022.