Image Caption Generation Using CNN & RNN Architectures for Visually Impaired Assistance

M.Praneeth[1], T. Vikas singh[2], V. Madhu[3], M.Narendra[4] , N.vishnu vardhan[5],G. Shruthi [6] ,

Dr. B. Venkataramana[7]

[1]Student, BtechCSE (DS) 4thYear, Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

praneeethkumarmk@gmail.com

[2]student, BtechCSE (DS) 4thYear, Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

Vikassinghthakur221@gmail.com

[3]student, BtechCSE (DS) 4thYear, Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

vurlugondamadhugoud@gmail.com

[4]student, BtechCSE (DS) 4thYear, Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

narendrakumar720721@gmail.com

[5]student, BtechCSE (DS) 4thYear, Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

naikotivishnuvardhan@gmail.com

[6]Asst.Prof CSE (DS), Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

geejulasruthi@gmail.com

[7]Assoc.Prof CSE (DS), Holy Mary Inst.Of Tech.And Science, Hyderabad, TG,India,

venkataramana.b@hmgi.ac.in

ABSTRACT

The project aims to develop an advanced Image Caption Generator using deep learning techniques and computer vision algorithms. In an era of increasing visual content on the internet, the ability to automatically generate descriptive captions for images has become crucial for enhancing accessibility and user experience. This project leverages state-of the-art deep neural networks, specifically Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs) for generating coherent and contextually relevant captions. The system takes an image as input and employs a pre-trained CNN to extract high level features, creating a rich representation of the visual content. Subsequently, an RNN-based sequence-to-sequence model processes these features to generate natural language captions. To improve the quality and fluency of captions, the model incorporates attention mechanisms, allowing it to focus on different parts of the image while generating each word. The outcome of this project has broad applications in fields such as image indexing, content retrieval, and accessibility, making digital visual content more understandable and engaging for a wide range of users. Additionally, the project contributes to the advancement of deep learning techniques in computer vision and natural language processing, pushing the boundaries of Al capabilities in understanding and describing visual information.

KeyWords ***Image Captioning,Convolutional Neural Network (CNN),Recurrent Neural Network (RNN),Long Short-Term Memory (LSTM),Encoder-Decoder Architecture,Feature Extraction,Image Features,Sequence Generation,Natural Language Processing (NLP)***

I.INTRODUCTION

In today's digital age, the ubiquity of visual content is undeniable. From social media platforms to educational resources, images have become a fundamental means of communication and information dissemination. Yet, one critical challenge remains: how can we make these images accessible and comprehensible to everyone, including individuals with visual impairments or those seeking enhanced search and retrieval capabilities? The answer lies in the development of an advanced Image Caption Generator—a cutting edge application of artificial intelligence (Al) and computer vision. This project embarks on a journey to bridge the gap between the visual and linguistic realms by harnessing the power of deep learning techniques and neural networks. The objective is to create a system that can automatically generate descriptive and contextually relevant captions for a wide array of images, making them not only visually appealing but also informative and accessible. The rationale behind this endeavor is twofold. First and foremost, it addresses the imperative of accessibility. For individuals with visual impairments, providing meaningful descriptions of images through generated captions enables them to engage with and understand digital content that was previously beyond their reach. This inclusivity aligns with the principles of universal design and ensures equitable access to information. Integrate attention mechanisms into the model to allow it to focus on different regions of the image while generating each word in the caption, improving the relevance of descriptions. Implement and use evaluation metrics such as BLEU, METEOR, and CIDEr to quantitatively assess the quality and relevance of generated captions in comparison to ground truth annotations. Test the image caption generator across various domains, including social media images, product images, educational materials, and more, to assess its adaptability and versatility.

II. LITERATURE REVIEW

Image caption generation using CNN and RNN architectures has been one of the most influential research directions in the intersection of computer vision and natural language processing. The fundamental idea is to combine the visual feature extraction capability of Convolutional Neural Networks (CNNs) with the sequence modeling ability of Recurrent Neural Networks (RNNs). In this framework, a CNN encodes the input image into a compact feature representation, and an RNN, typically a Long Short-Term Memory (LSTM) network, decodes these features to generate a descriptive sentence word by word. This encoder–decoder structure became the foundation for most early image captioning systems.

Problem Tended to: The past settled vector representation was inadequately to capture spatial detail, particularly when different objects or locales required to be portrayed distinctly. Proposed Arrangement: Presented the consideration instrument into encoderdecoder framework. This permitted the show to "go to" to distinctive parts of the picture whereas producing each word. Used delicate consideration (differentiable) to compute consideration weights over highlight maps. Advantages: Advantage: Empowered the demonstrate to powerfully localize pertinent picture locales, moving forward exactness and interpretability. Limitation: The utilize of delicate consideration expanded computational complexity, particularly for high-resolution pictures and longer captions. Impact: Spearheaded the integration of consideration in vision language assignments and affected future models like Transformer-based picture captioning systems. This not only minimizes idle times but also prevents unnecessary delays. One of the pioneering works in this area is Show and Tell: A Neural Image Caption Generator by Oriol Vinyals et al. (2015). This model used a pretrained CNN such as Inception to extract image features, which were then fed into an LSTM network to generate captions. The model was trained end-to-end using a maximum likelihood objective function. The study demonstrated that neural networks could directly translate visual content into coherent natural language descriptions, significantly outperforming earlier template-based and retrieval-based approaches. However, the use of a single fixed-length image vector limited the model’s ability to focus on specific regions of the image during caption generation.

To overcome this limitation, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention by Kelvin Xu et al. introduced the attention mechanism into the CNN-RNN framework. Instead of compressing the entire image into one vector, the model extracted spatial feature maps from the CNN and allowed the LSTM decoder to attend selectively to different regions of the image at each time step. This attention-based approach improved both performance and interpretability, as the model could visually indicate which parts of the image contributed to each generated word. The introduction of attention marked a significant improvement in caption relevance and descriptive accuracy.Another important contribution is Long-term Recurrent Convolutional Networks for Visual Recognition and Description by Jeff Donahue et al., which explored deeper integration of CNNs and RNNs for visual recognition and sequence generation tasks. Their work demonstrated that combining convolutional representations with recurrent layers could effectively model temporal and contextual information, reinforcing the suitability of CNN-RNN combinations for captioning tasks.

III. SYSTEM ARCHITECTURE AND METHODOLOGY :

Project Architecture

A. Model Architecture
1) Convolutional Neural Network (CNN)
Convolutional Neural Networks (CNNs) are founda- tional to the proposed image captioning model, acting as feature extractors that distill the essential visual information from input images.[27] The architecture of CNNs is well-suited to capturing intricate spatial hierarchies and local patterns within images, makingthem ideal for the initial preprocessing phase:

a) Feature Extraction:

- The CNN extracts high-level visual features directly from the raw image data. Using pre- trained models like VGG16 or ResNet50, the model processes the input image to generate a 256-dimensional feature vector that encapsulates key visual aspects such as textures, edges, and shapes

A. Model Architecture
1) Convolutional Neural Network (CNN) Convolutional Neural Networks (CNNs) are founda- tional to the proposed image captioning model, acting as feature extractors that distill the essential visual information from input images.[27] The architecture of CNNs is well-suited to capturing intricate spatial hierarchies and local patterns within images, making them ideal for the initial preprocessing phase:

a) Feature Extraction:

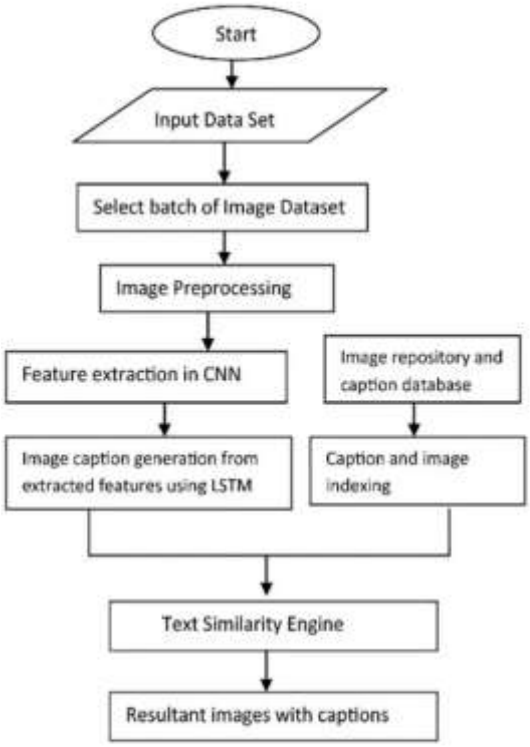
- The CNN extracts high-level visual features directly from the raw image data. Using pre- trained models like VGG16 or ResNet50, the model processes the input image to generate a 256-dimensional feature vector that encapsulates key visual aspects such as textures, edges, and shapes

A. Model Architecture
1) Convolutional Neural Network (CNN) Convolutional Neural Networks (CNNs) are founda- tional to the proposed image captioning model, acting as feature extractors that distill the essential visual information from input images.[27] The architecture of CNNs is well-suited to capturing intricate spatial hierarchies and local patterns within images, making them ideal for the initial preprocessing phase:

a) Feature Extraction:

- The CNN extracts high-level visual features directly from the raw image data. Using pre- trained models like VGG16 or ResNet50, the model processes the input image to generate a 256-dimensional feature vector that encapsulates key visual aspects such as textures, edges, and

shapesSystem Architecture of Image Caption Generation Using CNN & RNN Architectures:
The system architecture for image caption generation using CNN and RNN architectures typically involves the followi ng components.An image caption generation system using CNN + RNN follows an encoder–decoder architecture: the CNN encodes the image into a feature vector, and the RNN (often LSTM or GRU) decodes that vector into a natural language caption. This pipeline bridges computer vision and natural language processing, enabling automatic description of visual content.


Start
Input Data Set
Select batch of Image Dataset
Image Preprocessing
Feature extraction in CNN
Image repository and caption database
Image caption generation from extracted features using LSTM
Caption and image indexing
Text Similarity Engine
Resultant images with captions

System Architecture Overview

Encoder (CNN)

In an encoder-decoder model both the encoder and decoder are separate networks each one has its own specific task. These networks can be different types such as Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), Gated Recurrent Units (GRUs), Convolutional Neural Networks (CNNs) or even more advanced models like Transformers.

In deep learning the encoder-decoder model is a type of neural network that is mainly used for tasks where both the input and output are sequences. This architecture is used when the input and output sequences are not the same length for example translating a sentence from one language to another, summarizing a paragraph, describing an image with a caption or convert speech into text. It works in two stages:

Feature Transformation

Feature transformation in image caption generation refers to the process of converting raw image pixels into meaningful visual representations and then transforming those representations into natural language descriptions using deep learning models.

Most machine learning algorithms are statistics dependent, meaning that all of the algorithms are indirectly using a statistical approach to solve the complex problems in the data. In statistics, the normal distribution of the data is one that a statistician desires to be. A normal distribution of the data helps statisticians to solve the complex patterns of the data and gain valuable insights from the same. But for the algorithm scenario, a normal distribution of the data can not be desired every time with every type of dataset, which means the data which is not normally distributed needs preprocessing and cleaning before applying the machine learning algorithm to it.

In this article, we will be discussing the feature transformation techniques in machine learning which are used to transform the data from one form to another form, keeping the essence of the data. In simple words, the transformers are the type of functions that are applied to data that is not normally distributed, and once applied there is a high of getting normally distributed data.

Decoder (RNN)

In image caption generation, the decoder is responsible for transforming visual features extracted by the CNN encoder into a natural language sentence. The decoder is typically implemented using a Recurrent Neural Network (RNN) or its variants such as LSTM or GRU. In image caption generation, the decoder is responsible for converting visual features extracted by the CNN encoder into a natural language sentence. The decoder is typically implemented using a Recurrent Neural Network (RNN) or its variants such as LSTM or GRU.

Deep learning architectures have revolutionized the field of artificial intelligence, offering innovative solutions for complex problems across various domains, including computer vision, natural language processing, speech recognition, and generative models. This article explores some of the most influential deep learning architectures: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Transformers, and Encoder-Decoder architectures, highlighting their unique features, applications, and how they compare against each other.

Training

Training an image caption generation system involves teaching the model to map images to correct textual descriptions by jointly learning visual features (CNN) and language patterns (RNN).. Training Dataset consists of Images Multiple human-annotated captions per imageCommon datasets Flickr8k Flickr30k MS COCO

Before training, the captions associated with each image are preprocessed by tokenizing the text, converting words into numerical indices, and padding sequences to a uniform length. Special tokens such as start-of-sequence and end-of-sequence markers are added to help the model understand when a caption begins and ends. During training, the RNN decoder uses a technique called teacher forcing, where the actual previous word from the ground-truth caption is given as input at each time step to predict the next word in the sequence. This allows the model to learn correct word ordering and grammatical structure more effectively. Before training, the captions associated with each image are preprocessed by tokenizing the text, converting words into numerical indices, and padding sequences to a uniform length. Special tokens such as start-of-sequence and end-of-sequence markers are added to help the model understand when a caption begins and ends. During training, the RNN decoder uses a technique called teacher forcing, where the actual previous word from the ground-truth caption is given as input at each time step to predict the next word in the sequence. This allows the model to learn correct word ordering and grammatical structure more effectively

Convolutional Layers

Convolutional layers are the fundamental building blocks of Convolutional Neural Networks (CNNs) and play a crucial role in image-based tasks such as image caption generation. These layers are responsible for automatically extracting meaningful visual features from input images by applying a set of learnable filters, also known as kernels. Each filter slides over the image spatially and performs convolution operations to detect specific patterns such as edges, corners, textures, and shapes. The output of a convolutional layer is a feature map that highlights the presence of these visual patterns at different locations in the image. Convolutional layers are the fundamental building blocks of Convolutional Neural Networks (CNNs) and play a crucial role in image-based tasks such as image caption generation. These layers are responsible for automatically extracting meaningful visual features from input images by applying a set of learnable filters, also known as kernels. Each filter slides over the image spatially and performs convolution operations to detect specific patterns such as edges, corners, textures, and shapes. The output of a convolutional layer is a feature map that highlights the presence of these visual patterns at different locations in the image.

METHODOLOGY

The proposed ResNet (Residual Network) is a type of deep neural network architecture designed to address the vanishing gradient problem during training of deep convolutional neural networks (CNNs). ResNet introduces skip connections, also known as residual connections, which allow the network to learn residual functions. These skip connections pass the input directly to the output of deeper layers, enabling the model to skip over certain layers. This helps in mitigating the vanishing gradient problem, making it easier to train very deep networks. captions are cleaned, tokenized, converted into numerical sequences, and padded to maintain uniform length, with special tokens like <start> and <end> added to mark sentence boundaries. For feature extraction, a pre-trained CNN model such as VGG16, ResNet50, or InceptionV3 is used by removing its final classification layer and extracting high-level feature vectors that represent the image content. These features are then passed to an RNN-based decoder such as LSTM or GRU, which generates the caption sequentially, predicting one word at a time based on the image features and previously generated words. The model is trained using categorical cross-entropy loss and optimized through backpropagation through time, often applying techniques such as teacher forcing and batch training to improve learning efficiency. During inference, the trained model generates captions starting from the <start> token and continues until the <end> token is predicted, using strategies like greedy search or beam search to enhance output quality. The performance of the system is evaluated using standard metrics such as BLEU, ROUGE, METEOR, and CIDEr, which compare generated captions with ground-truth captions. Overall, this methodology integrates computer vision and natural language processing techniques to automatically generate meaningful and contextually accurate descriptions for images. For feature extraction, a pre-trained CNN such as VGG16, ResNet50, or InceptionV3 is used as the encoder. These models are generally pre-trained on large-scale datasets like ImageNet to learn rich visual representations. The final classification layer is removed, and the output from the last convolutional or pooling layer is extracted as a high-dimensional feature vector. This vector represents important visual elements such as objects, textures, and spatial information within the image. In some implementations, global average pooling is applied to reduce dimensionality and prevent overfitting.The decoder component is typically an RNN variant such as LSTM or GRU, which is well-suited for handling sequential data. The extracted image feature vector is either used to initialize the hidden state of the RNN or concatenated with word embeddings at each time step. During training, the model predicts the next word in the sequence based on the image features and previously generated words. A softmax activation function produces a probability distribution over the vocabulary, and categorical cross-entropy loss measures the prediction error. The network weights are updated using backpropagation through time (BPTT) and optimization algorithms such as Adam.To improve performance, an attention mechanism can be incorporated between the encoder and decoder. Instead of relying on a single global feature vector, attention allows the model to focus on different spatial regions of the image while generating each word. This dynamic alignment significantly enhances caption relevance and descriptive detail. Beam search decoding can also be applied during inference to generate more accurate and fluent captions compared to greedy search.Model evaluation is conducted using standard metrics such as BLEU, ROUGE, METEOR, and CIDEr, which compare generated captions with reference captions based on n-gram overlap and semantic similarity. Human evaluation may also be performed to assess grammatical correctness, relevance, and fluency. Finally, the trained model can be deployed in real-world applications such as assistive

DATASETS

In image caption generation using CNN and RNN architectures, datasets play a crucial role because they provide paired image–text data required to train the encoder–decoder model. These datasets contain images along with multiple human-written captions describing the content of each image. The CNN learns visual features from images, while the RNN learns language patterns from the captions. Some of the most widely used datasets are described below.One of the most popular datasets is MS COCO (Microsoft Common Objects in Context). It contains over 120,000 images with five captions per image. The dataset includes complex everyday scenes with multiple objects, making it highly suitable for training deep learning models. It also provides object detection and segmentation annotations, which are useful for advanced captioning models with attention mechanisms.Another commonly used dataset is Flickr8k. It contains 8,000 images, each paired with five captions written by humans. Due to its smaller size, it is widely used for academic projects and initial experimentation in CNN–RNN-based caption generation. It is suitable for understanding model architecture and implementation before moving to larger datasets.Similarly, Flickr30k is an extension of Flickr8k and contains 30,000 images with five captions per image. It provides more diversity and complexity compared to Flickr8k and is commonly used for benchmarking caption generation models.The Visual Genome dataset is another large-scale dataset that contains region-based descriptions, object annotations, attributes, and relationships between objects. It is particularly useful for models that incorporate attention mechanisms and detailed scene understanding.The Conceptual Captions dataset, created by Google, consists of millions of image-caption pairs collected from the web. Unlike MS COCO, the captions are automatically generated from web descriptions, making the dataset large but noisier. It is useful for training large-scale deep learning models.Another dataset is SBU Captioned Photo Dataset, which contains around one million image-caption pairs collected from Flickr. The captions are user-generated and may be less descriptive, but the dataset is valuable for large-scale training.For specialized tasks, datasets like AI Challenger Image Captioning Dataset provide captions in languages other than English, enabling multilingual caption generation research.In summary, datasets such as MS COCO, Flickr8k, Flickr30k, and Visual Genome are commonly used for training and evaluating CNN–RNN image captioning models. Smaller datasets are useful for experimentation and learning, while larger datasets improve model generalization and performance. The choice of dataset depends on project requirements, computational resources, and the desired level of caption complexity. In image caption generation using CNN and RNN architectures, datasets play a crucial role because they provide paired image–text data required to train the encoder–decoder model. These datasets contain images along with multiple human-written captions describing the content of each image. The CNN learns visual features from images, while the RNN learns language patterns from the captions. Some of the most widely used datasets are described below.One of the most popular datasets is MS COCO (Microsoft Common Objects in Context). It contains over 120,000 images with five captions per image. The dataset includes complex everyday scenes with multiple objects, making it highly suitable for training deep learning models. It also provides object detection and segmentation annotations, which are useful for advanced captioning models with attention mechanisms

IV. IMPLEMENTATION

The implementation of image caption generation involves building an **Encoder–Decoder model**, where a **CNN extracts image features** and an **RNN generates captions** based on those features. 1. Dataset SelectionUse standard datasets containing images with captions: Flickr8k,Flickr30k,MS COCO,Each image has multiple human-written captions.2. Data Preprocessing (a) Image Preprocessing ,Resize images to a fixed size (e.g., 224×224),Normalize pixel values,Convert images into numerical tensors

Implementation of image caption generation using CNN and RNN architectures involves building an end-to-end system that automatically generates textual descriptions for input images by combining visual feature extraction and sequence modeling. In this approach, a Convolutional Neural Network (CNN) is implemented as an encoder to process the input image and extract high-level visual features. A pre-trained CNN model such as VGG16, ResNet, or Inception is commonly used, with its final classification layer removed so that the output represents a dense feature vector capturing the semantic information of the image. This feature vector serves as the visual context for caption generation.The implementation begins with dataset preparation, where each image is paired with one or more human-annotated captions. The images are resized and normalized to match the input requirements of the CNN, while the captions are preprocessed through tokenization, vocabulary creation, integer encoding, and padding to a fixed length. Special tokens indicating the start and end of a sentence are added to guide the sequence generation process. These processed image–caption pairs form the training data used by the model.Once the image features are extracted, they are fed into the RNN decoder, which is typically implemented using Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) layers. An embedding layer converts each word index into a dense vector representation, and the image feature vector is either used to initialize the hidden state of the RNN or concatenated with the word embeddings. At each time step, the RNN predicts the next word in the caption based on the image features and the previously generated words. A dense layer with a softmax activation is used to output a probability distribution over the entire vocabulary

Computer Vision (CV)

Enables machines to understand visual content.Used to detect objects, scenes, and visual patterns in images.Implemented using Convolutional Neural Networks (CNNs). Convolution layers are key building blocks of convolutional neural networks (CNNs) which are used in computer vision and image processing. They apply convolution operation to the input data which involves a filter (or kernel) that slides over the input data, performing element-wise multiplications and summing the results to produce a feature map. This process allows the network to detect patterns such as edges, textures and shapes in the input images.Filters(Kernels):Small matrices that extract specific features from the input.For example, one filter might detect horizontal edges while another detects vertical edges.The values of filters are learned and updated during training.Stride:Refers to the step size with which the filter moves across the input data.Larger strides result in smaller output feature maps and faster computation.Padding:Zeros or other values may be added around the input to control the spatial dimensions of the output.Common types: "valid" (no padding) and "same" (pads output so feature map dimensions match input).Activation Function:After convolution, a non-linear function like ReLU (Rectified Linear Unit) is often applied allowing the network to learn complex relationships in data.Common activations: ReLU, Tanh, Leaky ReLU.Types of Convolution Layers2D Convolution (Conv2D): Most common for image data where filters slide in two dimensions (height and width) across the image.Depthwise Separable Convolution: Used for computational efficiency, applying depthwise and pointwise convolutions separately to reduce parameters and speed up computation.Dilated (Atrous) Convolution: Inserts spaces (zeros) between kernel elements to increase the receptive field without increasing computation, useful for tasks requiring context aggregation over larger areas. Steps in a Convolution Layer Initialize Filters: Randomly initialize a set of filters with learnable parameters.Convolve Filters with Input: Slide the filters across the width and height of the input data, computing the dot product between the filter and the input sub-region.Apply Activation Function: Apply a non-linear activation function to the convolved output to introduce non-linearity.Pooling (Optional): Often followed by a pooling layer (like max pooling) to reduce the spatial dimensions of the feature map and retain the most important information.Example Of Convolution LayerConsider an input image of size 32x32x3 (32x32 pixels with 3 color channels). A convolution layer with ten 5x5 filters, a stride of 1 and 'same' padding will produce an output feature map of size 32x32x10. Each of the 10 filters detects different features in the input image

Natural Language Processing (NLP)

Enables machines to generate meaningful sentences.Handles grammar, syntax, and semantics.Implemented using Recurrent Neural Networks (RNNs). Natural Language Processing (NLP) is a field that combines computer science, artificial intelligence and language studies. It helps computers understand, process and create human language in a way that makes sense and is useful. With the growing amount of text data from social media, websites and other sources, NLP is becoming a key tool to gain insights and automate tasks like analyzing text or translating languages. NLP is used by many applications that use language, such as text translation, voice recognition, text summarization and chatbots. You may have used some of these applications yourself, such as voice-operated GPS systems, digital assistants, speech-to-text software and customer service bots. NLP also helps businesses improve their efficiency, productivity and performance by simplifying complex tasks that involve language.

NLP Techniques

NLP encompasses a wide array of techniques that aimed at enabling computers to process and understand human language. These tasks can be categorized into several broad areas, each addressing different aspects of language processing. Here are some of the key NLP techniques:Text Processing and Preprocessing Tokenization: Dividing text into smaller units, such as words or sentences.Stemming and Lemmatization: Reducing words to their base or root forms.Stopword Removal: Removing common words (like "and", "the", "is") that may not carrysignificant meaning.Text Normalization: Standardizing text, including case normalization, removing punctuation and correcting spelling errors.Syntax and ParsingPart-of-Speech (POS)Tagging: Assigning parts of speech to each word in a sentence (e.g., noun, verb, adjective).Dependency Parsing: Analyzing the grammatical structure of a sentence to identify relationships between words.Constituency Parsing: Breaking down a sentence into its constituent parts or phrases (e.g., noun phrases, verb phrases).. Semantic AnalysisNamed Entity Recognition (NER): Identifying and classifying entities in text, such as names of people organizations, locations, dates, etc.Word Sense Disambiguation (WSD): Determining which meaning of a word is used in a given context.Coreference Resolution: Identifying when different words refer to the same entity in a text (e.g., "he" refers to "John").Information ExtractionEntity Extraction: Identifying specific entities and their relationships within the text.Relation Extraction: Identifying and categorizing the relationships between entities in a text. Text Classification in NLPSentiment Analysis: Determining the sentiment or emotional tone expressed in a text (e.g., positive, negative, neutral).Topic Modeling: Identifying topics or themes within a large collection of documents.Spam Detection: Classifying text as spam or not spamWorking in natural language processing (NLP) typically involves using computational techniques to analyze and understand human language. This can include tasks such as language understanding, language generation and language interaction.Text Input and Data CollectionData Collection: Gathering text data from various sources such as websites, books, social media or proprietary databases.Data Storage: Storing the collected text data in a structured format, such as a database or a collection of documents.Text Preprocessing

Preprocessing is crucial to clean and prepare the raw text data for analysis. Common preprocessing steps include:Tokenization: Splitting text into smaller units like words or sentences.Lowercasing: Converting all text to lowercase to ensure uniformity.Stopword Removal: Removing common words that do not contribute significant meaning, suchas "and," "the," "is."Punctuation Removal: Removing punctuation marks.Stemming and Lemmatization: Reducing words to their base or root forms. Stemming cuts off suffixes, while lemmatization considers the context and converts words to their meaningful base form.Text Normalization: Standardizing text format, including correcting spelling errors, expanding contractions and handling special characters.

Deep Learning Models

2.1 Convolutional Neural Networks (CNN)Extract visual features from images.Learns spatial and hierarchical representations.Common CNN models:VGG16,ResNet,Inception. Deep Learning is transforming the way machines understand, learn and interact with complex data. Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data. Neural network consists of layers of interconnected nodes or neurons that collaborate to process input data. In a fully connected deep neural network data flows through multiple layers where each neuron performs nonlinear transformations, allowing the model to learn intricate representations of the data.

data using nonlinear functions. The final output layer generates the model's prediction. EvoluArchitectures

Perceptron (1950s)First simple neural network with a single layerCould only solve linearly separable problemsFailed on complex tasks like the XOR problemMulti-Layer Perceptrons (MLPs)Introduced hidden layers and non-linear activation functionsEnabled modeling of non-linear relationshipsTrained effectively using backpropagationMarked a major leap in neural network capabilitiesTypes of neural networks

Feedforward neural networks (FNNs): They are the simplest type of ANN, where data flows in one direction from input to output. It is used for basic tasks like classification.Convolutional Neural Networks (CNNs): They are specialized for processing grid-like data, such as images. CNNs use convolutional layers to detect spatial hierarchies, making them ideal for computer vision tasks.Recurrent Neural Networks (RNNs): Theyare able to process sequential data, such as time series and natural language. RNNs have loops to retain information over time, enabling applications like language modeling and speech recognition. Variants like LSTMs and GRUs address vanishing gradient issues.Generative Adversarial Networks (GANs): This consist of two networks—a generator and a discriminator—that compete to create realistic data.

GANs are widely used for image generation, style transfer and data augmentation.Autoencoders: They are unsupervised networks that learn efficient data encodings. They compress input data into a latent representation and reconstruct it, useful for dimensionality reduction and anomaly detection.Transformer Networks: It has revolutionized NLP with self-attention mechanisms. Transformers excel at tasks like translation, text generation and sentiment analysis, powering models like GPT and BERT.Applications

In computer vision, deep learning models enable machines to identify and understand visual data. Some of the main applications of deep learning in computer vision include:Object detection and recognition: Deep learning models are used to identify and locate objects within images and videos, making it possible for machines to perform tasks such as self-driving cars, surveillance and robotics.

Image classification: Deep learning models can be used to classify images into categories such as animals, plants and buildings. This is used in applications such as medical imaging, quality control and image retrieval. Image segmentation: Deep learning models can be used for image segmentation into different regions, making it possible to identify specific features within images.Natural language processing (NLP)In NLP, deep learning model enable machines to understand and generate human language. Some of the main applications of deep learning in NLP include: Automatic Text Generation: Deep learning model can learn the corpus of text and new text like summaries, essays can be automatically generated using these trained models.Language translation: Deep learning models can translate text from one language to another, making it possible to communicate with people from different linguistic backgrounds. Sentiment analysis: Deep learning models can analyze the sentiment of a piece of text, making it possible to determine whether the text is positive, negative or neutral.Speech recognition: Deep learning models can recognize and transcribe spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search and voice-controlled devices

Reinforcement learningIn reinforcement learning, deep learning works as training agents to take action in an environment to maximize a reward. Some of the main applications of deep learning in reinforcement learning include: Game playing: Deep reinforcement learning models have been able to beat human experts at games such as Go, Chess and Atari. Robotics: Deep reinforcement learning models can be used to train robots to perform complex tasks such as grasping objects, navigation and manipulation. Control systems: Deep reinforcement learning models can be used to control complex systems such as power grids, traffic management and supply chain optimization.

Recurrent Neural Networks (RNN)

Processes sequential data.Generates captions word by word.Variants used:LSTM (Long Short-Term Memory)GRU (Gated Recurrent Unit) Recurrent Neural Networks (RNNs) differ from regular neural networks in how they process information. While standard neural networks pass information in one direction i.e. from input to output, RNNs feed information back into the network at each step. Imagine reading a sentence and you try to predict the next word, you don't rely only on the current word but also remember the words that came before. RNNs work similarly by "remembering" past information and passing the output from one step as input to the next i.e it considers all the earlier words to choose the most likely next word. This memory of previous steps helps the network understand context and make better predictions.

RNN unfolding or unrolling is the process of expanding the recurrent structure over time steps. During unfolding each step of the sequence is represented as a separate layer in a series illustrating how information flows across each time step.This unrolling enables backpropagation through time (BPTT) a learning process where errors are propagated across time steps to adjust the network's weights enhancing the RNN's ability to learn dependencies within sequential data. RNNs share similarities in input and output structures with other deep learning architectures but differ significantly in how information flows from input to output. Unlike traditional deep neural networks where each dense layer has distinct weight matrices.

RNNs use shared weights across time steps, allowing them to remember information over sequences.In RNNs the hidden state H_i is calcula Since RNNs process sequential data Backpropagation Through Time (BPTT) is used to update the network's parameters. The loss function L(θ) depends on the final hidden state h_3 and each hidden state relies on preceding ones forming a sequential dependency chainh_3 depends on depends on h_2, h_2 depends on $h_1, \ldots, h_1$ depends on h_0.ted for every input X_i to retain sequential dependencies. The computations follow these core formulas: In a One-to-Many RNN the network processes a single input to produce multiple outputs over time. This is useful in tasks where one input triggers a sequence of predictions (outputs). For example in image captioning a single image can be used as input to generate a sequence of words as a caption.

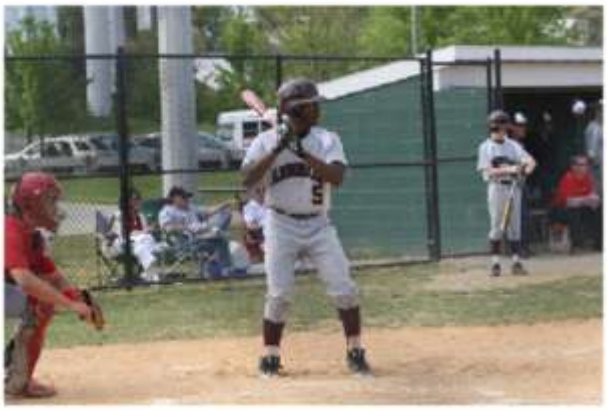
The Many-to-One RNN receives a sequence of inputs and generates a single output. This type is useful when the overall context of the input sequence is needed to make one prediction. In sentiment analysis the model receives a sequence of words (like a sentence) and produces a single output like positive, negative or neutral The Many-to-Many RNN type processes a sequence of inputs and generates a sequence of outputs. In language translation task a sequence

of words in one language is given as input and a corresponding sequence in another language is generated as output. This is the simplest type of neural network architecture where there is a single input and a single output. It is used for straightforward classification tasks such as binary classification where no sequential data is involved. Since RNNs process sequential data Backpropagation Through Time (BPTT) is used to update the network's parameters. The loss function L(θ) depends on the final hidden state h_3 and each hidden state relies on preceding ones forming a sequential dependency chain:
h_3 depends on depends on h_2, h_2 depends on $h_1, \ldots, h_1$ depends on h_0.

Each word in the phrase "feeling under the weather" is part of a sequence, where the order matters. The RNN tracks the context by maintaining a hidden state at each time step. A feedback loop is created by passing the hidden state from one-time step to the next. The hidden state acts as a memory that stores information about previous inputs. At each time step, the RNN processes the current input (for example, a word in a sentence) along with the hidden state from the previous time step. This allows the RNN to "remember" previous data points and use that information to influence the current output.

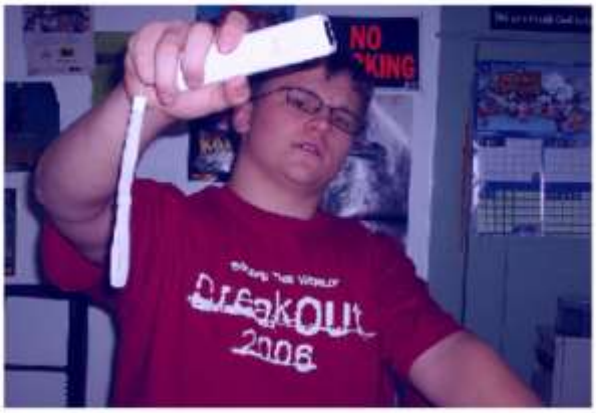
V.Results

a baseball player holding a bat on a field

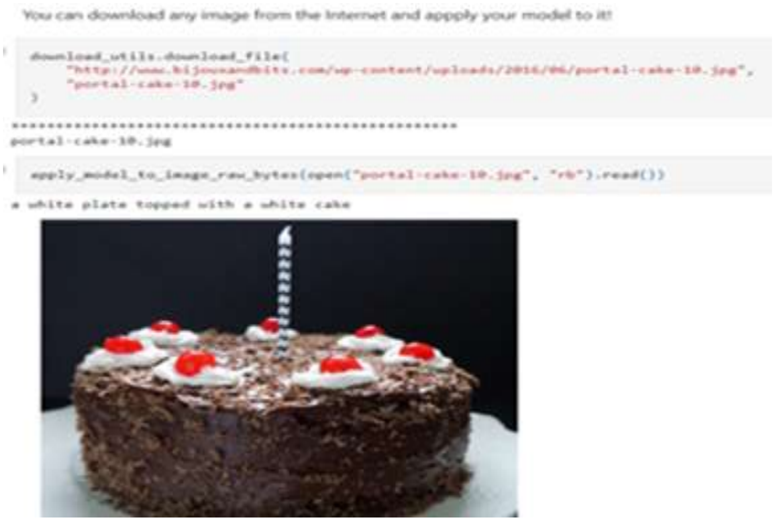


t9]:

a young boy is holding a donut in his hand



a group of sheep standing next to a fence



You can download any image from the Internet and appply your model to it!

```
download_utils.download_file(
    "http://www.bijouandbits.com/wp-content/uploads/2016/06/portal-cake-10.jpg",
    "portal-cake-10.jpg"
)
```

```
##########################################################
portal-cake-10.jpg
```

```
apply_model_to_image_raw_bytes(open("portal-cake-10.jpg", "rb").read())
```

```
a white plate topped with a white cake
```

VI. Conclusion and Future work

Conclusion

this paper, we have reviewed deep learning-based image captioning methods. We have givena taxonomy of image captioning techniques, shown generic block diagram of the major groups and highlighted their pros and cons. We discussed different evaluation metrics and datasets with weaknesses. their strengths and A brief summary of experimental results is also given. We briefly outlined potential research directions in this area. Although deep learningbased image captioning methods have achieved a 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. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for some timeOutput Caption of Given Image. We have used Flickr_8k dataset which includes nearly 8000 images, and the corresponding captions are also stored in the text file. Although deep learning -based image captioning methods have achieved a IJNRD2403518 International Journal of Novel Research and Development (www.ijnrd.org) f151 c151 © 2024 IJNRD | Volume 9, Issue 3 March 2024| ISSN: 2456-4184 | IJNRD.ORG remarkable progress in recent years, a robust image captioning method that is able to generate high quality captions for nearly Figure.6.1. The below picture image-caption pairs, the model learns to capture relevant semantic information from visual features. However, with a static image, embedding our caption generator will focus on features of our images useful for all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area for sometime. The scope of image-captioning is very vast in the future as the users are increasing day by day on social media and most of them would post photos.

The neural image caption generator gives a useful framework for learning to map from images to human-level image captions. By training on large numbers of image classification and not necessarily features useful for caption generation. To improve the amount of task-relevant information contained in each feature, we can train the image embedding model (the VGG-16 network used to encode features) as a piece of the caption generation model, allowing us to fine tune the image encoder to better fit the role of generating captions. Also, if we actually look closely at the captions generated, we notice that they Image caption generation using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) effectively integrates computer vision and natural language processing to automatically generate meaningful textual descriptions for images. In this approach, CNNs act as powerful visual feature extractors by learning spatial and semantic representations from images, while RNNs—particularly LSTM or GRU models—serve as decoders that transform these visual features into coherent and grammatically correct sentences.

The encoder–decoder architecture demonstrates strong performance in understanding image content and generating context-aware captions when trained on large, well-annotated datasets. The use of transfer learning with pretrained CNN models significantly reduces training time and improves accuracy, while sequential modeling in RNNs enables the system to capture temporal dependencies and linguistic structure.Despite its effectiveness, the CNN–RNN based approach has limitations, including difficulty in capturing complex object relationships and generating highly detailed captions. These challenges highlight the need for enhancements such as attention mechanisms and Transformer-based models. Nevertheless, CNN and RNN architectures remain a foundational and widely accepted solution for image caption generation and serve as a strong baseline for advanced captioning systems.Overall, this project demonstrates that CNN–RNN architectures provide a reliable and interpretable framework for image caption generation, with applications in assistive technologies, content retrieval, and human–computer interaction.If you want, I can also provide:Short conclusion (5–6 lines) Conclusion with future scope Conclusion for IEEE paper Conclusion + advantages & limitationsThe encoder–decoder architecture demonstrates strong performance in understanding image content and generating context-aware captions when trained on large, well-annotated datasets. The use of transfer learning with pretrained CNN models significantly reduces training time and improves accuracy, while sequential modeling in RNNs enables the system to capture temporal dependencies and linguistic structure.

Future Work

Future Work of Image Caption Generation using CNN & RNN ArchitecturesAlthough image caption generation using CNN and RNN architectures produces meaningful and coherent descriptions, there are several directions in which the system can be further improved and extended.Integration of Attention MechanismFuture systems can incorporate visual attention mechanisms to allow the model to focus on specific regions of an image while generating each word. This helps in generating more accurate and detailed captions, especially for images containing multiple objects and complex scenes.Adoption of Transformer-Based ModelsReplacing RNN-based decoders with Transformer architectures can improve caption quality by capturing long-range dependencies more effectively and enabling parallel processing during training, resulting in faster and more accurate caption generation.Fine-Grained Feature ExtractionAdvanced CNN backbones such as EfficientNet or Vision Transformers (ViT) can be used to extract richer visual features, leading to improved understanding of object relationships and scene context. Multimodal Learning EnhancemenFuture work can explore deeper multimodal fusion techniques to better align visual and textual representations, improving semantic consistency between images and generated captionsMultilingual Caption GeneratioThe system can be extended to generate captions in multiple languages, making it useful for global applications and accessibility tools. Real-Time and Mobile DeploymentOptimizing the model for real-time inference and deploying it on mobile or edge devices can enable applications such as smart cameras, assistive devices for visually impaired users, and embedded systems.

Improved Evaluation MethodIn addition to standard automatic metrics (BLEU, METEOR, CIDEr), future work can include human-centered evaluation to better assess caption relevance, fluency, and usefulnessDomain-Specific CaptioningThe model can be fine-tuned for domain-specific applications such as medical imaging, satellite imagery, surveillance, and autonomous driving, where specialized vocabulary and accuracy are required.Summary Statement (For Report)Future improvements in image caption generation using CNN and RNN architectures focus on enhancing visual attention, adopting advanced deep learning models, improving multimodal understanding, and expanding real-world applicability.Future work Image captioning has become an important problem in recent days due to the exponential growth of images in social media and the internet. This report discusses the various research in image retrieval used in the past and it also highlights the various techniques and methodology used in the research. As feature extraction and similarity calculation in images are challenging in this domain, there is a tremendous scope of possible research in the future. Current image retrieval systems use similarity calculation by making use of features such as color, tags, IMAGE RETRIEVAL USING IMAGE CAPTIONING 54 histogram, etc. There cannot be completely accurate results as these methodologies do not depend on the context of the image. Hence, complete research in image retrieval making use of context of the images such as image captioning will facilitate to solve this problem in the future. This project can be further enhanced in future to improve the identification of classes which has a lower precision by training it with more image

Future work Image captioning has become an important problem in recent days due to the exponential growth of images in social media and the internet. This report discusses the various research in image retrieval used in the past and it also highlights the various techniques and methodology used in the research. As feature extraction and similarity calculation in images are challenging in this domain, there is a tremendous scope of possible research in the future. Current image retrieval systems use similarity calculation by making use of features such as color, tags, 54 histogram, etc. There cannot be completely accurate results as these methodologies do not depend on the context of the image. Hence, complete research in image retrieval making use of context of the images such as image captioning will facilitate to solve this problem in the future. This project can be further enhanced in future to improve the identification of classes which has a lower precision by training it with more imageimage classification and not necessarily features useful for caption generation. To improve the amount of task-relevant information contained in each feature, we can train the image embedding model (the VGG-16 network used to encode features) as a piece of the caption generation model, allowing us to fine tune the image encoder to better fit the role of generating captions. Also, if we actually look closely at the captions generated, we notice that they

VIII.REFERENCES

[1] Abhaya Agarwal and Alon Lavie. 2008. Meteor, m-bleu and m-ter: Evaluation metrics for high correlation with human rankings of machine translation output. In Proceedings of the ThirdWorkshop on Statistical Machine Translation. Association for Computational Linguistics, 115—118.

[2] Ahmet Aker and Robert Gaizauskas. 2010. Generating image descriptions using dependency relational patterns. In Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 1250-1258.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In European Conference on Computer Vision. Springer, 382—398.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 (2017).

[5] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2018. Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5561—5570.

[6] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, Trevor Darrell, Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, et al. 2016. Deep compositional © 2024 IJNRD | Volume 9, Issue 3 March 2024| ISSN: 2456-4184 | IJNRD.ORG captioning: Describing novel object categories without paired training data. In Proceedings of the IEEE Conference on Computer Vision and Pattern recognition.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference Learning Representations (ICLR). on Shuang Bai and Shan An. 2018. A Survey on Automatic Image Caption Generation. Neurocomputing. ACM Computing Surveys, Vol. O, No. O, Article O. Acceptance Date: October 2018. 0:30 Hossain et al. Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human Neural Information Processing Systems. 1171—1179 .

[8] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. Journal of machine learning research 3, Feb, 1137—1155.

[9] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. Computational linguistics 22, 1 (1996), 39—71. [13] Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli IkizlerCinbis, Frank Keller, Adrian Muscat, Barbara Plank, et al. 2016. Automatic Description Generation from Images: A Survey of Models, judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Vol. 29. 65— 72.