ImageCaptioningUsingDeepLearning

Amajor project reportsubmittedinpartialfulfilmentofthe
requirement for the award of a degree of

BachelorofTechnology

in

ComputerScience&Engineering/InformationTechnology

Submittedby

KushagraShukla(201303)

SupritiSharma(201487)

Undertheguidance&supervisionof

Dr.KushalKanwar

DepartmentofComputerScience&Engineering and Information Technology

JaypeeUniversityofInformationTechnology,

Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **"Image Captioning Using Deep Learning"**in partial fulfilment of the requirementsfor theaward ofthe degree of **Bachelor of Technology** in**Computer Science&Engineering/InformationTechnology** submittedtotheDepartmentof Computer Science AndEngineering,JaypeeUniversityofInformationTechnology, Waknaghat is an authentic record of work carried out by **Kushagra Shukla(201303)**and**SupritiSharma(201487)**duringtheperiodfromAugust2023 toMay2024underthesupervisionof**Dr.KushalKanwar,**(AssistantProfessor(SG), Department of Computer Science and Engineering, Jaypee University of Information Technology).

(Student Signature with Date) (Student Signature with Date) Student Name: Kushagra Shukla Student Name: Supriti Sharma Roll No.: 201303 Roll No.:201487

Theabovestatementmadeiscorrecttothebestofourknowledge.

(Supervisor Signature with Date)
SupervisorName:Dr.KushalKanwar
Designation:AssistantProfessor(SG)
Department: CSE/IT
Dated:13May2024

DECLARATION

We at this moment declare that the work presented in this report entitled**'Image CaptioningUsingDeepLearning'** in partial fulfilment oftherequirementsfor the award of the degree of**Bachelor of Technology**in**Computer Science& Engineering / Information Technology** submittedintheDepartment ofComputer Science& Engineering and Information Technology**,**Jaypee University of Information Technology, Waknaghat is an authentic record of our work carried out over a period from August 2023 to May 2024 under the supervision of **Dr Kushal Kanwar**(AssistantProfessor(SG),Department ofComputer Science&Engineering and Information Technology).

Thematterembodiedinthereporthasnotbeensubmittedforanyotherdegreeor diploma award.

(StudentSignaturewithDate)(StudentSignaturewithDate)StudentName: KushagraShuklaStudentName:SupritiSharmaRollNo.:201303RollNo.: 201487

Thisistocertifythattheabovestatementmadebythecandidateistruetothebest of my knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr.KushalKanwar

Designation:AssistantProfessor(SG)

Department: CSE/IT

Dated:13May2024

ACKNOWLEDGEMENT

Firstly,IexpressmyheartiestthanksandgratefulnesstoalmightyGodforHis divine blessingto makeit possible to completethe project work successfully.

We are grateful and wish out profound indebtedness to Supervisor**Dr Kushal Kanwar, Assistant Professor(SG)**, Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge& keen interest of our supervisor in the field of "**Research Area**" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correctingthemat all stages have made it possible to complete this project.

We wouldalsogenerously welcomeeachoneofthoseindividuals whohavehelped me straightforwardly or in a roundabout way in making this project a win. In this uniquesituation, We want tothankthevariousstaffindividuals, botheducatingand non-instructing, who have developed their convenient help and facilitated our undertaking.

Finally,Wemustacknowledgewithduerespecttheconstantsupportandpatience of our parents.

KushagraShukla
(201303)

SupritiSharma
(201487)

TABLEOFCONTENT

Content

ListOfFigures

ListOfTables

Abstract

Thanks to the advancement of the deep learning, the field of the concept artificial intelligence has seen a significant increase in interest in research concerning the combination of natural language processing and computer vision. An English description of a photograph's context is automatically generated. When an image has a caption, the computer is trained to decipher the image's visual data using one or more sentences.

The meaningful description generating process of high-level picture semantics requiresthecapacitytoexaminethestate, characteristics,andrelationshipsbetween these objects.Inthis research, weaim to discover objectsand notify individuals via text messagesbyapplying CNN -LSTMarchitectural modelsonthecaptioningofa graphical image. In order to accurately identify the objects, the input image is first converted to grayscale.Computer vision and natural language processing were combined then. We made use of the COCO Dataset 2017.

In order to help blind people realizetheir full potential and track their intelligence, the suggestedapproach for blind people is meantto be broadened toinclude people with visual loss to speech messages. In this study, we create An iterative CNN-LSTM framework that outperforms human baselines by adhering to several key concepts in picture captioning and its conventional procedures.

CHAPTER-1INTRODUCTION

GeneralIntroduction

Wearesurroundedbyimagesinthenews,onsocialmedia,andinoursurroundings. Photoscan only berecognised by humans. Withouttheaccompanying descriptions, people are able to identify photographs; however, machines need to first be trained with images. Input vectors are used by the encoder-decoder Image Caption's architecture Generator models produce appropriate and legitimate captions. The domains of Natural language processing and computer vision are related by this paradigm. It is the responsibility to identify and assess the image's context before providing a full description in an organic language such as English.

Long short-term memory (LSTMs) and convolutional neural networks (CNNs) are the foundational models of our methodology. In the derived application, CNN is employed as an encoder With the goal to extract features from an image or snapshot, while LSTM is employed as a converter arrangeThe expressions and produce captions. Image captioningcanbeuseful for many purposes. For example, itcanhelptheblindvia text-to-speechbyprovidingtrue-timeinformationabout the scene via a camera feed. It can also increase social media leisure by organizing captions for both spokenremarks and photos in social media feeds.

Onesteptowardslearningthelanguageistohelpkidsidentify chemicals. Authentic photoexplorationandindexingcan beacceleratedandimproved byaddingcaptions to every image available online.Image captioning is used in many fields, including biology, business, the internet, and self-driving motor vehicles, in which the surroundings are explained and CCTV cameras, which can trigger alarms if they detect malicious activity. This research article's primary goal is to provide readers with a fundamental understanding of deep learning techniques.

ProblemStatement

In the modern world, information is valued, and some people have severe difficulty seeing images. We investigate this further, taking blindness into account as a significant factor, and produce a sentence by letting usersupload or scan a picture.

Advantage:-

- RecommendationsinEditingApplications
- Assistanceforvisuallyimpaired
- SocialMediaposts
- Self-Drivingcars
- Robotics
- Easytoimplementandconnecttonewdata sources

Disadvantages:-

- Neverofferanend-to-enddevelopedgeneralmodeltosolvethisproblem,
- Neithertheyofferintuitivefeatureobservationsontheobjectsoractionsinthe image.

1.2.2ProblemAnalysis

As image captioning is a very vast and underdeveloped field of deep learning there are many problems such as incorrect captions that are being generated for a specific picture that is uploaded in the model for the captions. This can be reduced by using correct techniques and proper guidance if a person has knowledge in the field of deep learning,and neural networks this problem can be easily solved andthe captions thatis to be generated willbe correct.

Scope:Ourprojectextendsandisbeingusedinanylarge-scalebusinessindustryandalso small-scale business industry.

Objectives

1.Theproject'sgoalistodevelopaDeepLearning(DL) methodforcontextualizing photographs in basic English sentences.

2.TherequirementtoworkwithLSTMandCNNratherthanRNN

3. Toworkonthewebpageforthecaptiongeneration.

SignificanceandMotivationoftheProjectWork

Making captions for imagesisan essential task in the domains of computer vision and natural language processing images. It is an impressive advancement in artificial intelligence when a machine replicate human ability to describe images. Capturing the connections between the objects in the picture and how to describe them in a language that is natural (like English) is the main challenge ofthistask.Computer systems have historically generated text descriptions for images by utilizing pre-defined templates. Neural networks are frequently employed in state-of-the-art modelstogenerate captionsbyusinganimageasinputand forecastingthe next lexical unit in the output sentence. Nevertheless, this method falls short of offering the necessarydiversitytoproducelexicallyrichtext descriptions. Withneural networks'increased efficiency, this weakness has been suppressed.

We can develop a product for the blind that will allow them to navigate the roadways without theassistanceofothers.We couldaccomplishthisbyturningthe scenetotextandthenthetext to voiceautomatic driving is oneamong the most important challengesand if we can properly captionthesceneroundthecar,it canenhancetotheself-drivingsystem.Automaticcaptioning can help Google Image Search, which will become nearly as excellent as Google Search, as each image is first converted into a caption, and then searches are conducted based on the caption.CCTV cameras are now ubiquitous, but if we combine viewing the globe with the generation ofappropriate captions, we will beable to raisealarmsassoon ascriminal conduct is detected. This will undoubtedly aid in thereduction of crime and accidents.

The model based on LSTM and CNN uses deep convolutional neural network (CNN) is to create a dense feature vector from the images. The dense vectors are also called embedding. Fortheimagecaption model,thisembeddingactsasa denserepresentationoftheimage which can be used as the initial state ofthe LSTM. The CNN network can be trained directly on the images in our dataset. An LSTM is recurrent neural network architecture that is mainly used for problems havingtemporal dependences. It useful for capturinginformationabout previous states to better informthe current predictionthroughits memory cell state.

Chapter-2

LiteratureSurvey

OverviewOfRelevantLiterature

The most crucial stage in the the process for creating software is the literature review. DeterminingThe economy, time factor, and strength of the company is essential before developing the tool.Selecting the operating system and language that can be utilized for the toolisthenext step. development oncetheserequirementsare met.Theprogrammersrequirea great deal of outside assistance once they begin developing the tool. Books, websites, and senior programmers can all provide this support. The aforementioned factors are taken into account before developing the system that is suggested.

Most of the project's development industry takes into account and thoroughly investigates every needneededtodevelopthe project. Aliteraturereviewisthe mostimportant crucial step inthetheprocess ofdeveloping software foranyproject.Itisvitaltoascertainandevaluatethe company's strength, economy, workforce, resource requirements, and time factor prior to developing the tools alongside their designs.

To improve and personalize the user experience on its products, photos use image classification.Numeroustypicalcomputervisionissues,suchasintraclassvariation,occlusion, deformation, size variation, perspective variation, and lighting, are represented by the picture classification problem. Techniques that are effective for classifying pictures are also likely to be effective for other crucial computer vision tasks, such as segmentation, detection, and localization.

Imagecaptioningisagreat illustrationofthis.Givenanimage,theimagecaptioning challenge is to generate a sentence description of the image. The picture captioning problem is comparable to the image classification problem in that it expects more detail and has a bigger universe of possibilities. Image classification isused asa black box system in modern picture captioning systems, therefore greater image classification leads to better captioned.

The image captioning problem is interesting in and of itself because it combines natural language processing and computer vision, two important areas of artificial intelligence. An image captioning systemshowsthat it can comprehend naturallanguageandimagesemantics. Once all of these demands are met, the next step is to determine the software specifications in thecorresponding system,includingwhat kindofoperatingsystemtheproject wouldneedand what all the software is needed to move onto the next step, which is developing the tools and the related operations and thoroughly surveyed.

Image classification isa crucial step in the object recognition and picture analysisprocessthat leads to the construction ofan image sentence. A statement could be

the end result of the image categorization process. Many methods for captioning images have beenpresentedthusfar.Theidealmethodforcaptioningimageshasbeenthesubjectof numerous studies. Since the outcomesandaccuracy varydepending ona numberof factors, it is challenging to select one method as the best of all of them.

Both new image captioning techniques developed in the last few decades and traditional approaches have been continuously modified to achieve the most accurate results. Numerous complex tasks like image classification, semantic segmentation, and object detection, have demonstratedexceptionalperformancewithmultilayerconvolutionalneuralnetworksinrecent times. In particular, two-stage approaches are often employed for semantic segmentation. In this manner, Convolutional neuronal networks are learned to deliver high-quality local pixel-wise data for the subsequent step, which istypically a more comprehensive graphical analysis model.

The Vanishing Gradient problem will be resolved by employing Long Short-Term Memory (LSTM)., a subset of RNNs. The Vanishing Gradients problem is the primary objective of LSTM. The ability of LSTM to retain data values over extended periods of time makes it uniquein solving the vanishing gradient problem. Numerousadvanced tasks, including object detection, image classification,and, morerecently, semantic segmentation, haverecently been proven to obtain outstanding results using convolutional neural networks with many layers. A two-stage technique is frequently used, especially for semantic segmentation. In this way, convolutional networksaretrained to giveexcellentlocal pixel-wise data for the second stage, which is often a more global graphical analysis model. The results showed that using a combination ofLSTM produced better resultsthan applying RNN. Unlike conventional image recognition algorithms, CNNs use multilayer convolutional to carry out feature engineering and integratethese featuresinternally. It also makesuse of SoftMax, the fully connected (FC) and poolingL

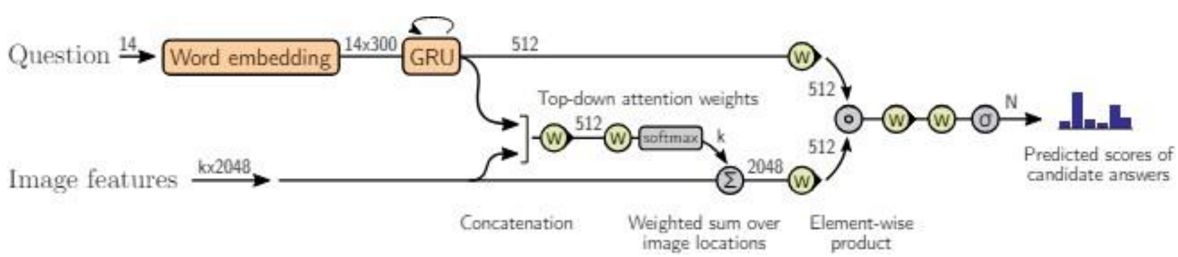

Fig1OverviewoftheproposedVAQmodel

V.ProposedSystem

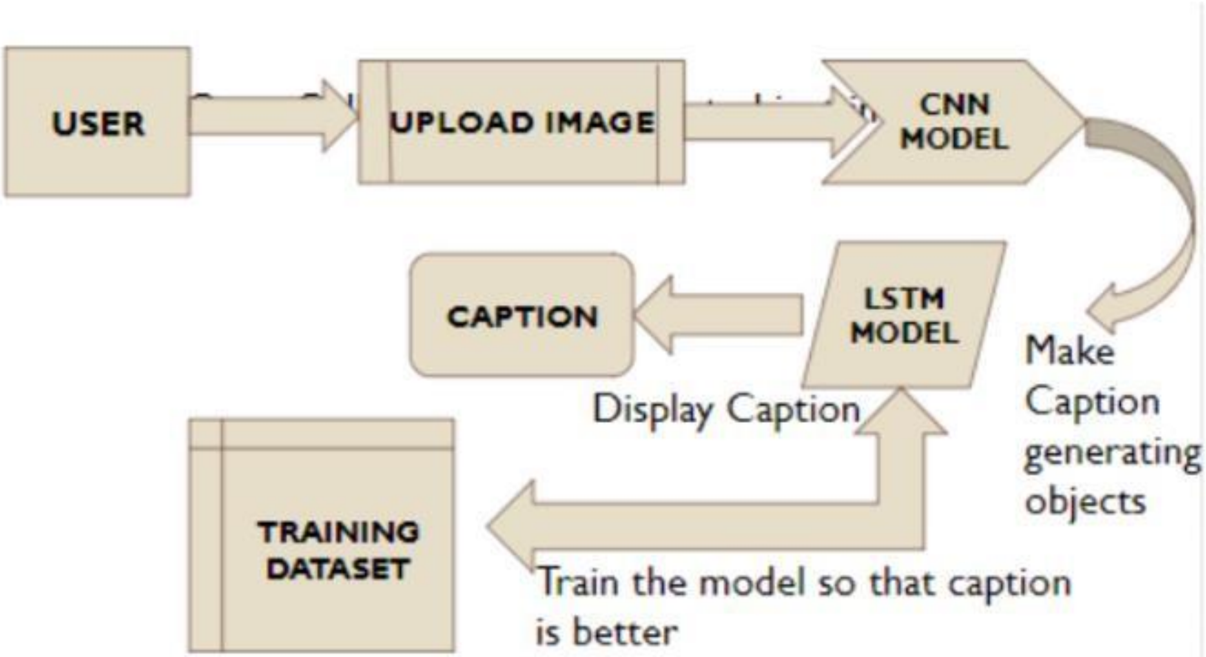

Fig2.Proposed ModelSystem

SNo.	AuthorName	Dataset and Methodology	Results
1.	RajendranSubash	Dataset: MS COCOMethod: NLP and CNNLSTM basedmodel	Using CNN LSTM andNLPtechniques themodelforimage captioning is generated
2.	Seung-HoHan,Ho-JinChoi	Semantic Ontology	Using semantic ontology the model forimagecaptioning is generated
3.	PranayMathur,AmanGill, Nand Kumar Bansode, Anurag Mishra	Dataset: MS COCO Method: Advanced deep reinforcement learning based on NLPand computerVision	The model proposed generates the real time environment highqualitycaptions with the help of tenserflow
4.	SimaoHerdade, Armin Kappeler,KofiBoakye, Joao Soares	Dataset: MS COCOMethod: architecture model using CNN as well as NLPtechniques	Using CNN and LSTMmodelsthe image'scaptionis generated.
5.	Manish Raypurkar, Abhishek Supe, Pratik Bhumkar,PravinBorse, Dr.Shabnam Sayyad	Dataset: Flickr_8k Method: CNN and LSTM modeltoextract features and sequence the words and finally generating captions.	Proposed model is basedonmultilabel Neural networks
6.	Oriol	Dataset:MS	Proposedmodelis

	vinyals,AlexanderToshev,SamyBengio,DumitruErhan.	COCO Method:Deep recurrent network	basedonneural networksystem
7.	JianhuiChen,WenqiangDong,Michen Li.	Dataset:flickr_8k,MS COCO Method:LRCN, CNN,RNN	Using CNN,LSTM,RNN models the image's captionisgenerated.
8.	Peter Anderson,XiaodongHe,ChrisBuchler,DamienTeney,MarkJohnson,StephenGould, And Lei Zhang	Dataset:MS COCO Method:Bottom up and Top down mechanism	UsingBottomup And Top down mechanism the image caption is generated
9.	JyotiAneja,Aditya DeshpandeAndAlexander G Schwing	Dataset:Flickr_8k Method:CNN And RNNfor spatialimages	Using CNN and RNNthecaptionfor image is generated
10.	ShuangBaiAndShanAn	Dataset:MS COCO Method:Neural Network	Using the dataset and neural network conceptsthecaption for image is generated.
Table1			

KeyGapsinTheLiterature

- Couldnotdescribethecaptionsbasedondifferenttargets.
- Doesnotrecogonizetheminutechangesintheimageduringthecaptioning.
- LossishigherforCNNasofRNN.
- AccuracyandPrecesionarenottheidealparameterforthemodelevaluation.
- Insufficientintransformingobjectsintowords.
- Doesnotworkonthelargedataset.
- BottomUpmechanismperformssuperiorlyascomparedtoTopdownmechanismandlanguage LSTM.
- Eachimagehasatleast5Captions.

Chapter-3
SystemDevelopment

REQUIREMENTSANDANALYSIS

SYSTEMREQUIREMENTS

All computer software requires specific hardware parts or additional software resources to function properly on a computer. These prerequisites are referred to as computer system requirements, and they are frequently used as recommendations rather than strict regulations. Two sets of system requirements are typically defined by software. minimal as well as suggested. System requirements typically rise over time due to the growing demand for more processing power and resources in newer software versions. Industry observers contend that rather than technological breakthroughs, this trend is more responsible for the upgrades of current computer systems.

HARDWAREREQUIREMENTS

• System:i3Processor

• HardDisk:500GB.

• Monitor:15’’LED

• InputDevices:Keyboard,Mouse

• Ram: 4GB.

SOFTWAREREQUIREMENTS

• Platform:GoogleColab

• CodingLanguage:Python

ProjectDesignAndArchitecture

Usingdeeplearningandcomputervision,animagecaptiongeneratorcandetermineitscontext and comment on it with pertinent captions. It involves using datasets provided during model training to label an image with English keywords. The CNN model called Xception has been trained using the imagenet dataset. The extraction of characteristics from images is done by Xception. The LSTM model will receive these extracted features and use them to create the image caption..

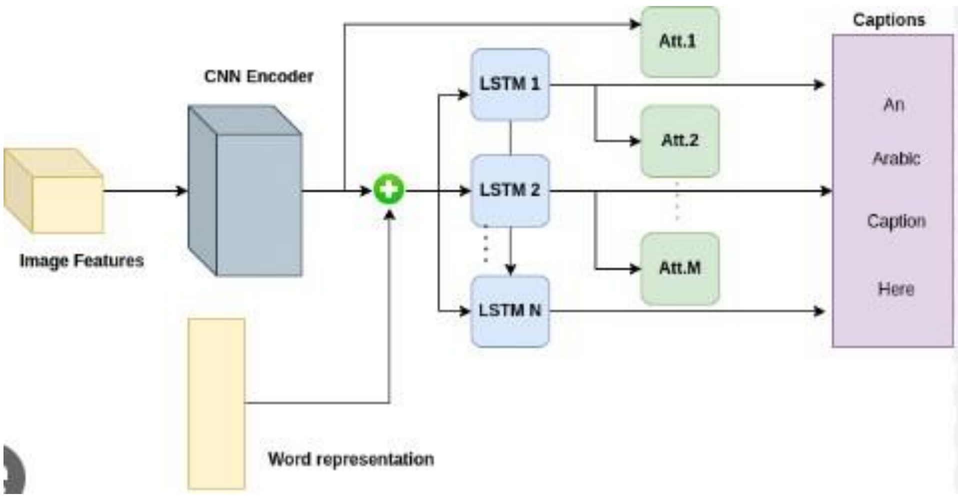

Fig3.ArchitectureAndDesignOfProject

DataPreparation

For tasks including finding objects, segmentation, and captioning, an in-depth image recognition dataset is called COCO (Common Objects in Context). It has more than 330,000 photos, each with five captions that describe the scene and 80 object categories annotated on them. Many cutting-edge object detection and segmentation models have been trained and evaluatedusing the COCO dataset, which is extensivelyused in computer visionresearch.

Theimagesandtheirannotationsmakeupthetwo maincomponentsof the dataset.Theimages arearranged into a hierarchy ofdirectories, withthetrain, validation,and test setscontainedin subdirectories of the top-level directory. JSON files containing annotations are supplied; each file corresponds to a single image.

Thefollowingdetailsareincludedwitheveryannotationinthedataset:-

● Nameofimagefile

● Imagedimensions(heightandwidth)

● Alistofitemscontainingthedatalistedbelow:classofobjects(suchas"person,","car"); coordinates ofthe bounding box; segmentation mask (x, y, width, height);. Importantideasandwheretheyfitin(ifavailable).

● Fivecaptionsthatexplaintheimage.

- AdditionalinformationisalsoavailablefromCOCOdatasetincludinglicense,supercategories for images and coco-stuff.
- Fortheapplicationoftheimagecaptiongeneratorweusedthedatasetnamesflickr_8kdataset
- Thisdatasetcontainsawiderangeofimagesthathasmanydifferenttypesofsituationand scenes.
- Flickr_8kdatasethas8000imagesand everyimagehas5captions.
- Wedividedtheentiredatasetof8000imagesas6000,1000and1000astraining,validationand testing respectively.
- Everyimagehasdifferentdimensions.

LibrariesUsed

- Tensorflow:itsanopensource librarythat supportsdeep learning usingpythonetc frameworks

- Keras:itsanopensourcepythonlibrarythatallowstoevaluatethedeeplearning models.

- Pillow:Pillow isapythonimagine library(PIL),thataddssupportfor opening,manipulating,and saving images.

- Numpy:Toworkwitharrays,numpylibraryisused.

- Matplotlib:Librarytocreatestaticandanimatedvisualisationsinpythonframework.

- Transformers: This is the flagship library that provides state of the art pre trained models for the natural language understanding,generation,andtranslation.It alsosupports both Pytorch and tensorflow frameworks and includes a wide range of models from BERT to GPT.

- Tokenizers: It offers efficient tokenization algorithms essential for preprocessing the text data before feeding it into the deep learning models.It also supports various tokenization strategies including byte pair encoding and word piece tokenization.

- Datasets: They simplifies the process of loading and pre processing datasets for the training and evaluation.It provides a unified interface for accessing various datasets commonly used in natural language processing tasks.

These Librariesformthe core infrastructurefor building,Fine-Tuning and deployingthe advanced NLP models and applications within the hugging face ecosystem.They do enable researchers and developers to easilyaccess and leverage cutting edge techniques in the field.

Implementation

WorkingExplanation

1. Tocreateacaptionforanimage,theuseruploadstheimage.
2. CNNisusedtoprocessagrayscaleimageandidentifytheobjects.
3. CNNisusedtoprocessagrayscaleimageandidentifytheobjects.
4. CNNextractssignificantimagefeaturesbyscanningimagesfromlefttorightand fromtop to bottom.

5. Usinganactivationfunctionandavarietyoflayers,includingpooling, convolutional,and fully connected, We succeededin clearing the features from eachimage.

6. Afterthat,LSTMisused.

Algorithms

1.ConvulationalNeuralNetwork
2.LongShort-TermMemory

OverviewOnCNN

One kind of the model for deep learning used for processing data with a grid pattern, like images, is the convolutional neural network (CNN). To classify a probabilistic object with values ranging from 0 to 1, each input image will pass through a series of fully connected layers(FC), pooling, convolution layerswith filters(Kernels), andtheSoftmax function. CNN models with deep learning are utilized for training and testing.CNNs are distinct from RNNs and other neural networks due to their convolutional layers. The input is transformed in a convolutional layer before being forwarded to the following layer. A CNN applies filters to change the data.

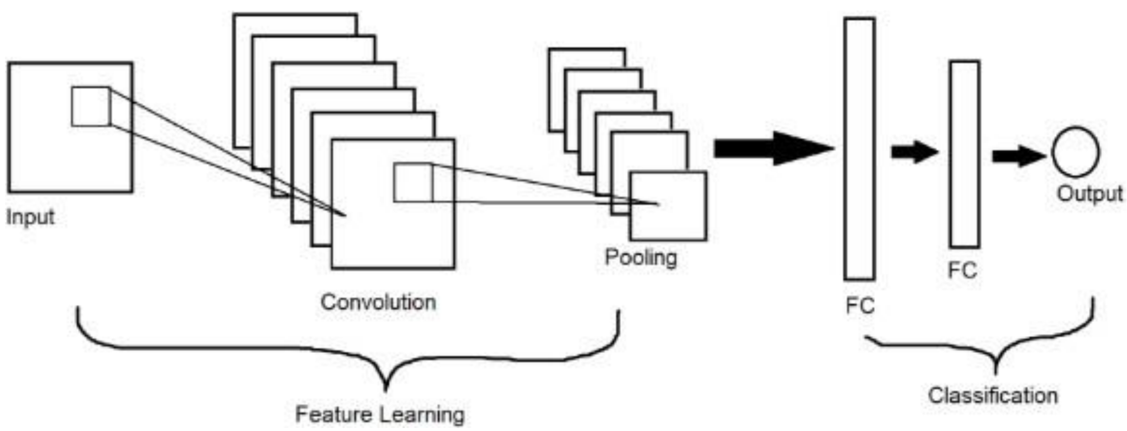

Fig4.CNN

SomeadvantagesofCNN:

- Convolutionalneuralnetworks(CNNs)areOnekindofdeeplearningmodelwhichprocesses grid-patterned data such as images.

- To classify an object with probabilistic values between 0 and 1, each input image will pass through a series of convolution layers with filters (Kernels), pooling, fully connected layers (FC), and the Softmax function. CNN models with deep learning are employed for both training and testing.

- CNNs have convolutional layers, which set them apart from RNNs and other neural networks.Before being passed to the next layer, the input is transformed in a convolutional layer. CNN modifies the data by applying filters.

OverviewOfLSTM

Recurrent neural networks of the LSTM type can recognise order dependence in sequence prediction problems.In difficult Thisbehaviorisrequired in problemdomainssuchasmachine translation and speech recognition, among others.One intricate subfield of deep learning is LSTMs. In difficult problem domains like speech recognition and machine translation, among others, this behavior is necessary. Oneintricate subfield of deep learningin LSTM’s.

SomeAdvantagesofLSTM are:

- Providesuswithawiderangeofparametersincludinginputandoutputbiasesandlearning rates.
- UsingLSTMsthecomplexitytoupdateeachweightisloweredtoO(1).

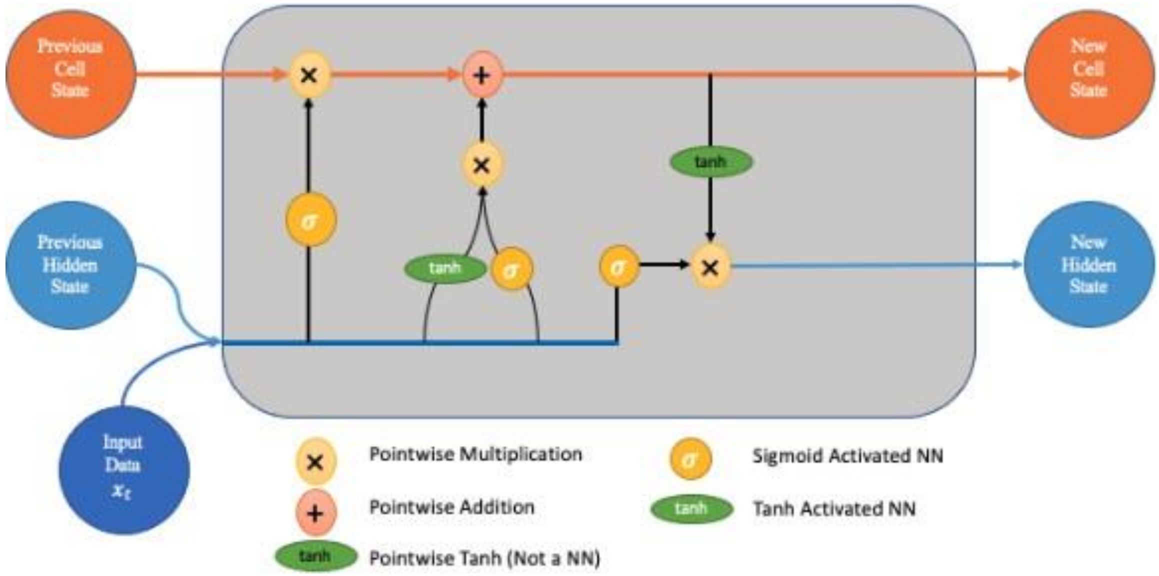

Fig5.LSTM

CNN-LSTMArchitectureModel

In order to support sequence prediction, CNN's LSTM architecture combines LSTMs with Layers in a convolutional neural network (CNN) operate to extract characteristics from input data.

CNN-LSTMs were created for applications such as textual description generation from image sequences (e.g., videos) and visualtime series prediction problems.In particular,the issue of

- ActivityRecognition:Makingatextualdescriptionofaprocedurewhichappearsinaseriesof methods.
- ImageDescription:Generatingatextualdescriptionofasingleimage.
- VideoDescription:Makinganessaysummaryofasetofimages.

Although we will refer to this architecture as"CNN LSTM," the original name for it was "LRCN model stands for Long-Term Recurrent Convolutional Network.In order to extract Featuresoftheillustration,CNNisutilised.The Xceptionpre-trained modelwillbeemployed.

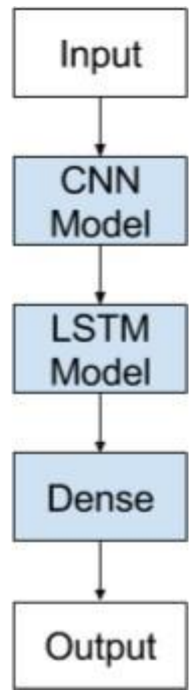

Fig6.CNN-LSTMModel

Techniques

- AddLibraries.
  - Providethe2017COCO(CommonObjectsandContexts)Dataset.(PreprocessingData)
  - UseCNNtodeterminewhichobjectsareinthe picture. Tokenize and preprocess the captions.
  - Predictthesentence'snextwordusingLSTM.CreateaData GeneratorandViewCaptioned Images.

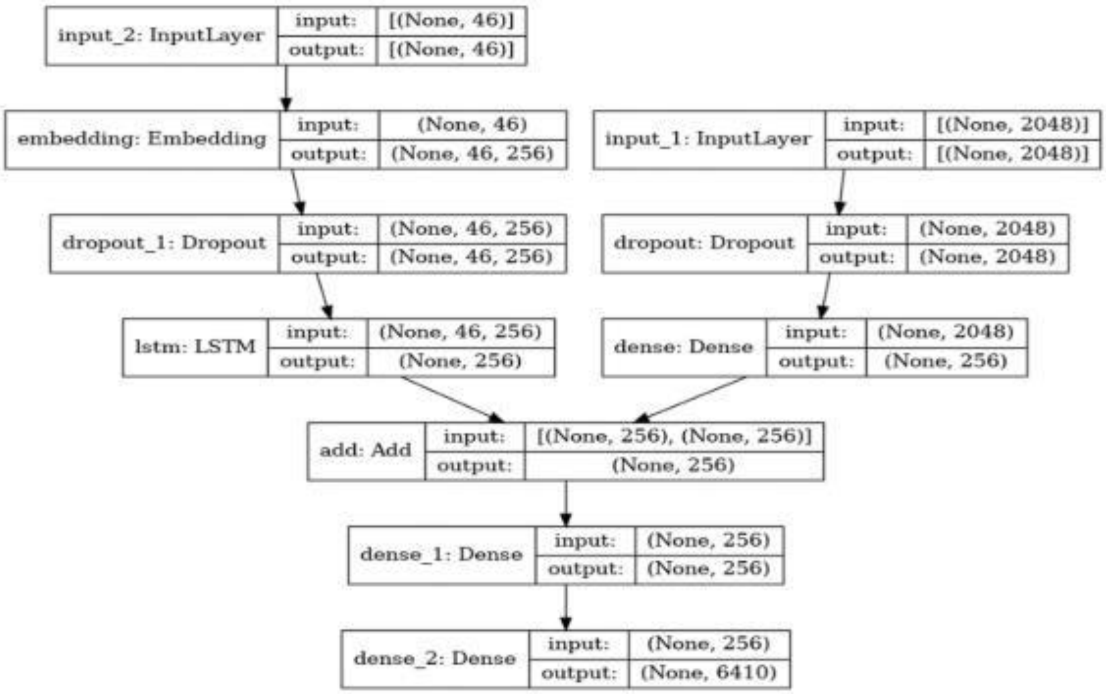

Fig7.WorkflowDiagram

CodeSnippets

```
!pip -q install kaggle
```

```
[2] import tensorflow as tf
    import os
    import json
    import pandas as pd
    import re
    import numpy as np
    import time
    import matplotlib.pyplot as plt
    import collections
    import random
    import requests
    from math import sqrt
    from PIL import Image
    from tqdm.auto import tqdm
```

Fig8.ImportingLibraries

```
[3] if not os.path.exists('/content/data/'):

        api_token = {"username": "<-- your username -->",
                     "key": "<-- your api key -->"}

        with open('/content/kaggle.json', 'w') as file:
            json.dump(api_token, file)

        os.environ["KAGGLE_CONFIG_DIR"] = "/content/"

        os.system('kaggle datasets download -d adityajn105/flickr8k')
        os.makedirs('/content/data/', exist_ok=True)
        os.system('mv /content/flickr8k.zip /content/data/flickr8k.zip')
        os.system('unzip -q /content/data/flickr8k.zip -d /content/data/')
        os.remove('/content/data/flickr8k.zip')
```

```
captions = pd.read_csv('/content/data/captions.txt')
captions['image'] = captions['image'].apply(
    lambda x: f'/content/data/Images/{x}')
captions.head()
```

[4]

	image	caption
0	/content/data/Images/1000268201_693b08cb0e.jpg	A child in a pink dress is climbing up a set o...
1	/content/data/Images/1000268201_693b08cb0e.jpg	A girl going into a wooden building .
2	/content/data/Images/1000268201_693b08cb0e.jpg	A little girl climbing into a wooden playhouse .
3	/content/data/Images/1000268201_693b08cb0e.jpg	A little girl climbing the stairs to her playh...
4	/content/data/Images/1000268201_693b08cb0e.jpg	A little girl in a pink dress going into a woo...

Next steps: Generate code with captions | View recommended plots

```
def preprocess(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub('\s+', ' ', text)
    text = text.strip()
    text = '[start] ' + text + ' [end]'
    return text
```

```
[6] captions['caption'] = captions['caption'].apply(preprocess)
    captions.head()
```

Fig9.Importing Data

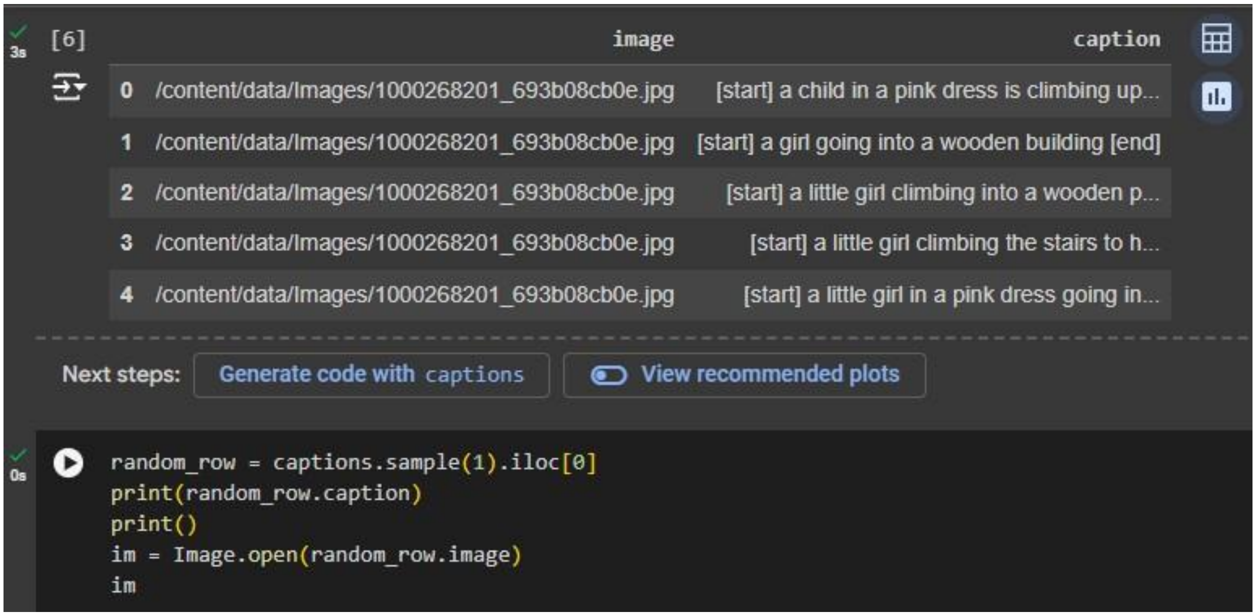

Fig10.InitialCodeGeneration

Fig11. First Outcome

```
[8]  MAX_LENGTH = 40
     VOCABULARY_SIZE = 10000
     BATCH_SIZE = 32
     BUFFER_SIZE = 1000
     EMBEDDING_DIM = 512
     UNITS = 512
```

```
[9]  tokenizer = tf.keras.layers.TextVectorization(
         max_tokens=VOCABULARY_SIZE,
         standardize=None,
         output_sequence_length=MAX_LENGTH)

     tokenizer.adapt(captions['caption'])
```

```
[10] word2idx = tf.keras.layers.StringLookup(
         mask_token="",
         vocabulary=tokenizer.get_vocabulary())

     idx2word = tf.keras.layers.StringLookup(
         mask_token="",
         vocabulary=tokenizer.get_vocabulary(),
         invert=True)
```

Fig12.AssigningValuestoelement

```python
img_to_cap_vector = collections.defaultdict(list)
for img, cap in zip(captions['image'], captions['caption']):
    img_to_cap_vector[img].append(cap)

img_keys = list(img_to_cap_vector.keys())
random.shuffle(img_keys)

slice_index = int(len(img_keys)*0.8)
img_name_train_keys, img_name_val_keys = (img_keys[:slice_index],
                                          img_keys[slice_index:])

train_imgs = []
train_captions = []
for imgt in img_name_train_keys:
    capt_len = len(img_to_cap_vector[imgt])
    train_imgs.extend([imgt] * capt_len)
    train_captions.extend(img_to_cap_vector[imgt])

val_imgs = []
val_captions = []
for imgv in img_name_val_keys:
    capv_len = len(img_to_cap_vector[imgv])
    val_imgs.extend([imgv] * capv_len)
    val_captions.extend(img_to_cap_vector[imgv])
```

Fig13.ImagetocaptionGeneration

```
[12] len(train_imgs), len(train_captions), len(val_imgs), len(val_captions)

(32360, 32360, 8095, 8095)
```

```
def load_data(img_path, caption):
    img = tf.io.read_file(img_path)
    img = tf.io.decode_jpeg(img, channels=3)
    img = tf.keras.layers.Resizing(299, 299)(img)
    img = img / 255.
    caption = tokenizer(caption)
    return img, caption
```

Fig14. Assigningthelength

```
[14] train_dataset = tf.data.Dataset.from_tensor_slices(
         (train_imgs, train_captions))

     train_dataset = train_dataset.map(
         load_data, num_parallel_calls=tf.data.AUTOTUNE
         ).shuffle(BUFFER_SIZE).batch(BATCH_SIZE)

     val_dataset = tf.data.Dataset.from_tensor_slices(
         (val_imgs, val_captions))

     val_dataset = val_dataset.map(
         load_data, num_parallel_calls=tf.data.AUTOTUNE
         ).shuffle(BUFFER_SIZE).batch(BATCH_SIZE)
```

```
[15] image_augmentation = tf.keras.Sequential(
         [
             tf.keras.layers.RandomFlip("horizontal"),
             tf.keras.layers.RandomRotation(0.2),
             tf.keras.layers.RandomContrast(0.3),
         ]
     )
```

Fig15. TrainingDataset

```python
def CNN_Encoder():
    inception_v3 = tf.keras.applications.InceptionV3(
        include_top=False,
        weights='imagenet'
    )
    inception_v3.trainable = False

    output = inception_v3.output
    output = tf.keras.layers.Reshape(
        (-1, output.shape[-1]))(output)

    cnn_model = tf.keras.models.Model(inception_v3.input, output)
    return cnn_model
```

Fig16.CNNEncoder

```python
class TransformerEncoderLayer(tf.keras.layers.Layer):

    def __init__(self, embed_dim, num_heads):
        super().__init__()
        self.layer_norm_1 = tf.keras.layers.LayerNormalization()
        self.layer_norm_2 = tf.keras.layers.LayerNormalization()
        self.attention = tf.keras.layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim)
        self.dense = tf.keras.layers.Dense(embed_dim, activation="relu")


    def call(self, x, training):
        x = self.layer_norm_1(x)
        x = self.dense(x)

        attn_output = self.attention(
            query=x,
            value=x,
            key=x,
            attention_mask=None,
            training=training
        )

        x = self.layer_norm_2(x + attn_output)
        return x
```

Fig17.TransformerEncodingLayer

```python
class Embeddings(tf.keras.layers.Layer):

    def __init__(self, vocab_size, embed_dim, max_len):
        super().__init__()
        self.token_embeddings = tf.keras.layers.Embedding(
            vocab_size, embed_dim)
        self.position_embeddings = tf.keras.layers.Embedding(
            max_len, embed_dim, input_shape=(None, max_len))


    def call(self, input_ids):
        length = tf.shape(input_ids)[-1]
        position_ids = tf.range(start=0, limit=length, delta=1)
        position_ids = tf.expand_dims(position_ids, axis=0)

        token_embeddings = self.token_embeddings(input_ids)
        position_embeddings = self.position_embeddings(position_ids)

        return token_embeddings + position_embeddings
```

```python
[19] Embeddings(tokenizer.vocabulary_size(), EMBEDDING_DIM, MAX_LENGTH)(next(iter(train_dataset))[1]).shape
```

```
TensorShape([32, 40, 512])
```

Fig18.Defininglength,range,anddimension

```
class TransformerDecoderLayer(tf.keras.layers.Layer):

    def __init__(self, embed_dim, units, num_heads):
        super().__init__()
        self.embedding = Embeddings(
            tokenizer.vocabulary_size(), embed_dim, MAX_LENGTH)

        self.attention_1 = tf.keras.layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim, dropout=0.1
        )
        self.attention_2 = tf.keras.layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim, dropout=0.1
        )

        self.layernorm_1 = tf.keras.layers.LayerNormalization()
        self.layernorm_2 = tf.keras.layers.LayerNormalization()
        self.layernorm_3 = tf.keras.layers.LayerNormalization()

        self.ffn_layer_1 = tf.keras.layers.Dense(units, activation="relu")
        self.ffn_layer_2 = tf.keras.layers.Dense(embed_dim)

        self.out = tf.keras.layers.Dense(tokenizer.vocabulary_size(), activation="softmax")

        self.dropout_1 = tf.keras.layers.Dropout(0.3)
        self.dropout_2 = tf.keras.layers.Dropout(0.5)
```

Fig19.TransformerDecodingLayer

```python
        out_2 = self.layernorm_2(out_1 + attn_output_2)

        ffn_out = self.ffn_layer_1(out_2)
        ffn_out = self.dropout_1(ffn_out, training=training)
        ffn_out = self.ffn_layer_2(ffn_out)

        ffn_out = self.layernorm_3(ffn_out + out_2)
        ffn_out = self.dropout_2(ffn_out, training=training)
        preds = self.out(ffn_out)
        return preds


    def get_causal_attention_mask(self, inputs):
        input_shape = tf.shape(inputs)
        batch_size, sequence_length = input_shape[0], input_shape[1]
        i = tf.range(sequence_length)[:, tf.newaxis]
        j = tf.range(sequence_length)
        mask = tf.cast(i >= j, dtype="int32")
        mask = tf.reshape(mask, (1, input_shape[1], input_shape[1]))
        mult = tf.concat(
            [tf.expand_dims(batch_size, -1), tf.constant([1, 1], dtype=tf.int32)],
            axis=0
        )
        return tf.tile(mask, mult)
```

Fig20. Decoding Layer

```python
class ImageCaptioningModel(tf.keras.Model):

    def __init__(self, cnn_model, encoder, decoder, image_aug=None):
        super().__init__()
        self.cnn_model = cnn_model
        self.encoder = encoder
        self.decoder = decoder
        self.image_aug = image_aug
        self.loss_tracker = tf.keras.metrics.Mean(name="loss")
        self.acc_tracker = tf.keras.metrics.Mean(name="accuracy")


    def calculate_loss(self, y_true, y_pred, mask):
        loss = self.loss(y_true, y_pred)
        mask = tf.cast(mask, dtype=loss.dtype)
        loss *= mask
        return tf.reduce_sum(loss) / tf.reduce_sum(mask)


    def calculate_accuracy(self, y_true, y_pred, mask):
        accuracy = tf.equal(y_true, tf.argmax(y_pred, axis=2))
        accuracy = tf.math.logical_and(mask, accuracy)
        accuracy = tf.cast(accuracy, dtype=tf.float32)
        mask = tf.cast(mask, dtype=tf.float32)
        return tf.reduce_sum(accuracy) / tf.reduce_sum(mask)
```

Fig21.ICModelWithLossAndAccuracy

```python
    def compute_loss_and_acc(self, img_embed, captions, training=True):
        encoder_output = self.encoder(img_embed, training=True)
        y_input = captions[:, :-1]
        y_true = captions[:, 1:]
        mask = (y_true != 0)
        y_pred = self.decoder(
            y_input, encoder_output, training=True, mask=mask
        )
        loss = self.calculate_loss(y_true, y_pred, mask)
        acc = self.calculate_accuracy(y_true, y_pred, mask)
        return loss, acc


    def train_step(self, batch):
        imgs, captions = batch

        if self.image_aug:
            imgs = self.image_aug(imgs)

        img_embed = self.cnn_model(imgs)

        with tf.GradientTape() as tape:
            loss, acc = self.compute_loss_and_acc(
                img_embed, captions
            )
```

Fig22.ComputingLossAnd Accuracy

```python
[22] encoder = TransformerEncoderLayer(EMBEDDING_DIM, 1)
     decoder = TransformerDecoderLayer(EMBEDDING_DIM, UNITS, 8)

     cnn_model = CNN_Encoder()
     caption_model = ImageCaptioningModel(
         cnn_model=cnn_model, encoder=encoder, decoder=decoder, image_aug=image_augmentation,
     )

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [==============================] - 0s 0us/step

cross_entropy = tf.keras.losses.SparseCategoricalCrossentropy(
    from_logits=False, reduction="none"
)

early_stopping = tf.keras.callbacks.EarlyStopping(patience=3, restore_best_weights=True)

caption_model.compile(
    optimizer=tf.keras.optimizers.Adam(),
    loss=cross_entropy
)
```

Fig23. Encoder Decoder

```
history = caption_model.fit(
    train_dataset,
    epochs=5,
    validation_data=val_dataset,
    callbacks=[early_stopping]
)
```

```
Epoch 1/5
1012/1012 [==============================] - 311s 278ms/step - loss: 4.3215 - acc: 0.2231 - val_loss: 3.7957 - val_acc: 0.3202
Epoch 2/5
1012/1012 [==============================] - 276s 269ms/step - loss: 3.5071 - acc: 0.3369 - val_loss: 3.5019 - val_acc: 0.3484
Epoch 3/5
1012/1012 [==============================] - 309s 301ms/step - loss: 3.2519 - acc: 0.3630 - val_loss: 3.4091 - val_acc: 0.3617
Epoch 4/5
1012/1012 [==============================] - 307s 300ms/step - loss: 3.0939 - acc: 0.3786 - val_loss: 3.3570 - val_acc: 0.3676
Epoch 5/5
1012/1012 [==============================] - 274s 268ms/step - loss: 2.9738 - acc: 0.3900 - val_loss: 3.3221 - val_acc: 0.3730
```

Fig24. TrainingDataset

```
[27] idx = random.randrange(0, len(val_imgs))
     img_path = val_imgs[idx]

     pred_caption = generate_caption(img_path)
     print('Predicted Caption:', pred_caption)
     print()
     Image.open(img_path)
```

```
Predicted Caption: a white crane is flying in the air
```

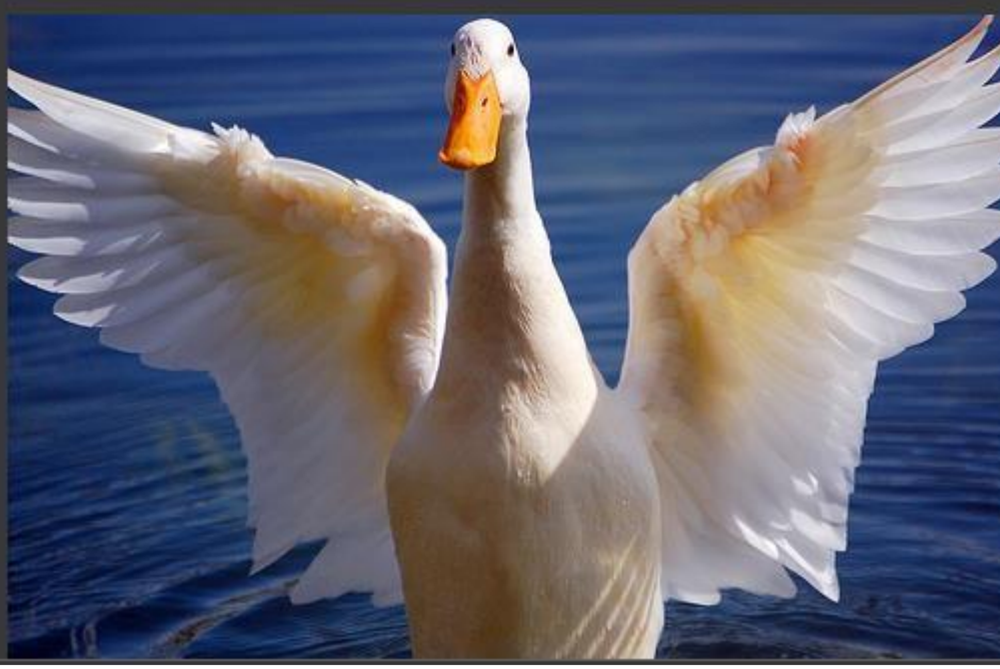

Fig25.PredictedOutput

```python
url = "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT2j6yclbKYDav4BGUKLAdTvSFXp1gtuzy5DQ&usqp=CAU"
im = Image.open(requests.get(url, stream=True).raw)
im.save('tmp.jpg')

pred_caption = generate_caption('tmp.jpg')
print('Predicted Caption:', pred_caption)
print()
im
```

```
Predicted Caption: a black and white dog is jumping over a hurdle
```

Fig26.PredictedOutput

```python
CHUNK_SIZE = 40960
DATA_SOURCE_MAPPING = 'coco-2017-dataset:https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-data-sets%2F857191%2F1462296%2Fbundle%2Farchive.zip%3FX-Goog-Algorithm%3D

KAGGLE_INPUT_PATH='/kaggle/input'
KAGGLE_WORKING_PATH='/kaggle/working'
KAGGLE_SYMLINK='kaggle'

!umount /kaggle/input/ 2> /dev/null
shutil.rmtree('/kaggle/input', ignore_errors=True)
os.makedirs(KAGGLE_INPUT_PATH, 0o777, exist_ok=True)
os.makedirs(KAGGLE_WORKING_PATH, 0o777, exist_ok=True)

try:
  os.symlink(KAGGLE_INPUT_PATH, os.path.join("..", 'input'), target_is_directory=True)
except FileExistsError:
  pass
try:
  os.symlink(KAGGLE_WORKING_PATH, os.path.join("..", 'working'), target_is_directory=True)
except FileExistsError:
  pass
```

Fig27.DataSourceMapping

```python
class TransformerEncoderLayer(tf.keras.layers.Layer):

    def __init__(self, embed_dim, num_heads):
        super().__init__()
        self.layer_norm_1 = tf.keras.layers.LayerNormalization()
        self.layer_norm_2 = tf.keras.layers.LayerNormalization()
        self.attention = tf.keras.layers.MultiHeadAttention(
            num_heads=num_heads, key_dim=embed_dim)
        self.dense = tf.keras.layers.Dense(embed_dim, activation="relu")


    def call(self, x, training):
        x = self.layer_norm_1(x)
        x = self.dense(x)

        attn_output = self.attention(
            query=x,
            value=x,
            key=x,
            attention_mask=None,
            training=training
        )

        x = self.layer_norm_2(x + attn_output)
        return x
```

Fig28.OutputLayer

```python
def calculate_accuracy(self, y_true, y_pred, mask):
    accuracy = tf.equal(y_true, tf.argmax(y_pred, axis=2))
    accuracy = tf.math.logical_and(mask, accuracy)
    accuracy = tf.cast(accuracy, dtype=tf.float32)
    mask = tf.cast(mask, dtype=tf.float32)
    return tf.reduce_sum(accuracy) / tf.reduce_sum(mask)


def compute_loss_and_acc(self, img_embed, captions, training=True):
    encoder_output = self.encoder(img_embed, training=True)
    y_input = captions[:, :-1]
    y_true = captions[:, 1:]
    mask = (y_true != 0)
    y_pred = self.decoder(
        y_input, encoder_output, training=True, mask=mask
    )
    loss = self.calculate_loss(y_true, y_pred, mask)
    acc = self.calculate_accuracy(y_true, y_pred, mask)
    return loss, acc
```

Fig29.AccuracyAndLossFunction

KeyChallenges

SemanticUnderstanding

Solution:EmploydeeplearningmodelslikeCNNsforthefeature extraction and combine with RNNs.

VisualRecogonition

Solution:UtilizePre-Trained objectdetection modelorthefine-tunethemfor specific datasets to improve accuracy in recognition the visuals.

LanguageUnderstanding

Solution: Use techniques like attention mechanisms and beam search during captiongenerationtofocusonrelevantpartsoftheimageandimprovecoherence in the language generation.

HandlingAmbiguity

Solution:solutionistoaugmentthedatasetwithdiverseimagesandthecaptions to expose the model to various contexts,reducing ambiguity.

Handlingrareorunseenobjects

Solutions:Leveragingtransferlearningbyfinetuningmodelsonthespecific captioning tasks and utilizing techniques like domain adaption togeneralize better to rare or the unseen objects.

Chapter-4
Testing

TestingStrategy

The trained model will now be loaded and predictions will be produced by a separate file called testing_caption_generator.py. We will extract the words from their index values using the same tokenizer.p pickle file since the predictions include the maximum length of index values.Thanks to the advancement of deep neural networks,Image captioning (IC) systems generatea textdescription ofthe important objectsinanimageautomatically.images, whether theyaresyntheticor real—haveadvanced significantly inrecentyears. In human society,IC is essential for tasks like labeling large photos for research studies and helping those with visual impairmentssee the world. But even the best IC systems—like IBM Image Caption Generator and Microsoft Azure Cognitive Services—can produce inaccurate results, which leaves out important information and deep misunderstanding.

We suggest MetaIC, the first metamorphic testing strategy for IC system validation, as a solution to thisissue. Our main hypothesisisthat, following object insertion, theobject names ought to show directional changes. In particular, MetaIC:

(1) createsanobjectcorpusbyextractingobjectsfrompre-existingimages;

(2) uses innovative Location tuning and object resizing algorithms to insert an object into an image; and

(3) indicates see pairs with captions that don't show expected differences. We evaluate one commonly-used image captioning API and five state-of-the-art (SOTA) MetaIC-based image captioning models. Using 1,000 seeds, MetaIC successfully reports 16,825 erroneous issues with high precision (84.9%-98.4%).

(4) AtestingstrategyincludemetricessuchasBLEUScore,METEOR,ROUGE,whichmeasure the similarity between generated captions and thereference captions provided by humans.

UnitTesting

- Preprocessing Units: It tests the preprocessing steps suchas image resizing,normalization and the feature extraction to ensurethat they produce the expected andreal outputs.

- Model Components: Test each component of the captioning model including theencoder and decoder.

End-To-EndTesting

- DataPipelineTesting:Verifiesthedatapipelineiscorrectincludingdataloading,feature extraction and preprocessing.

- Performance Testing: Measure the performance and ensures the system that it meets the performance requirements or not.

SecurityTesting

- Data Poisoning: Assessing the model to poisoned training data where maliciously images or captions are being inserted into the training data to manipulate the outcome and model's behaviour.

- PrivacyConcern:Ensuringthatthecaptioningsystemdonotrevealsensitiveinformationabout individual depicted in the images.

TestCasesAndOutcomes

Test cases, which are used to assess an image captioning model's performance, are usually pairs of images and the captions that go with them. The captions that the model produces for every test case are the results. To evaluate the caliber of the generated captions, these results arethen contrasted with the ground truth captions, or the real captions. This isan explanation:

1.Testscenarios:

- Animagefromadatasetoranactualsituationistheinputimage.
- GroundTruthCaption:Theaccuratecaptionthataccuratelysummarizesthepicture's content.

2.Results:

- GeneratedCaption:Thecaptionthataninputimage'simagecaptioningmodelgeneratesfor it.
- Evaluation Metrics: A number of metrics are commonly used to measure the degree to which the generated caption corresponds to the ground truth caption, including BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit Ordering),CIDEr(Consensus-basedImageDescriptionEvaluation),ROUGE(Recall-Oriented Understudy for Gisting Evaluation), etc.

3. Processofevaluation:

- QuantitativeEvaluation: Using evaluation metrics,thegenerated captionsare compared to the ground truth captions. These metrics provide numerical scores indicating the quality of the generated captions.

- Qualitative Evaluation: Human reviewers may also subjectively rate the captions based on criteria like fluency, coherence, and relevance.

4. Asanillustration:

- Correct Outcome: The evaluation metrics accurately reflect the accuracy with which the generated caption describes the image's content.

Chapter-5
ResultAndEvaluation

Results

Theresultof thisprogram isgoing to bea user beingallowedto generatea caption for a visual image using Deep Learning, NLP, and Computer Vision. Using the concept of CNN and LSTM we have build a model and trained the model using flickr_8k data set and MS COCO for gettingtheappropriate captions forthe given image.We have tried different methods to get the perfect outcome.

After installing the the data set and loading it into the model the model will train the dataset and give the desired result as per the image.And we have used hugging face as as the hosting element in which we can entertheurl oftheimage or the image fromthe dataset folder forthe outcome.

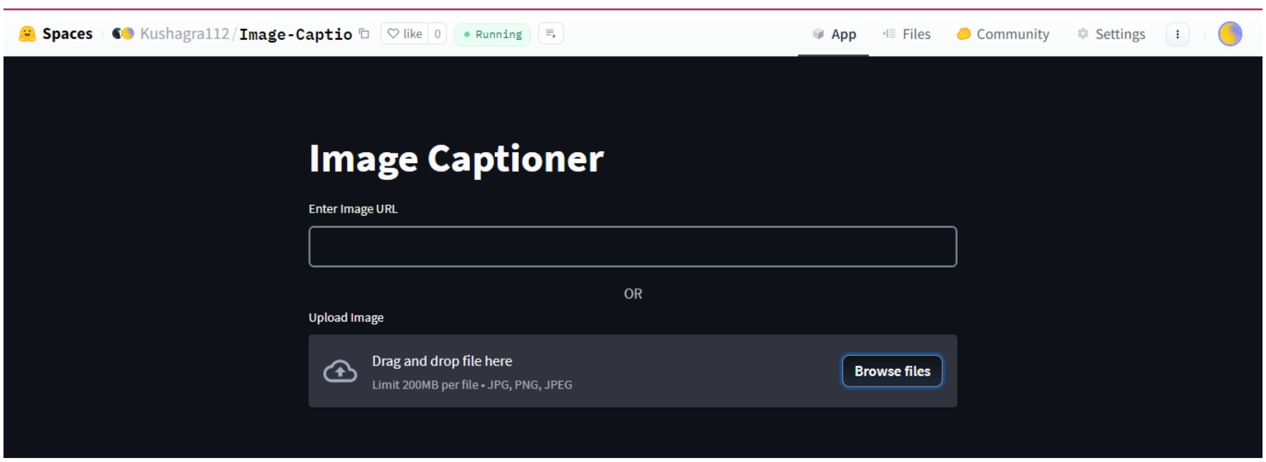

Fig30. HuggingFaceInterface

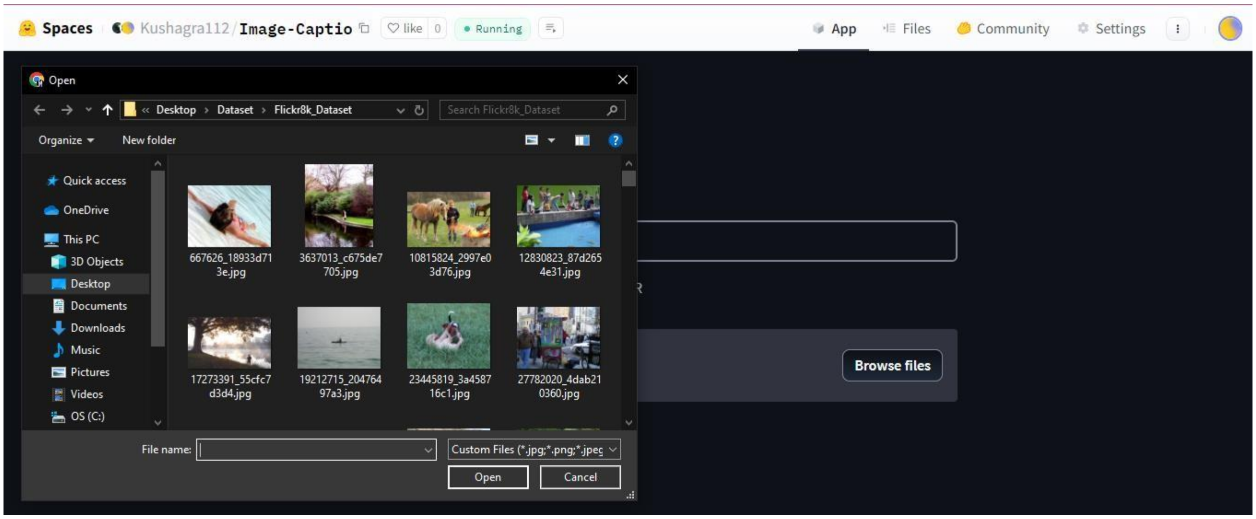

Fig31.Uploadingtheimage

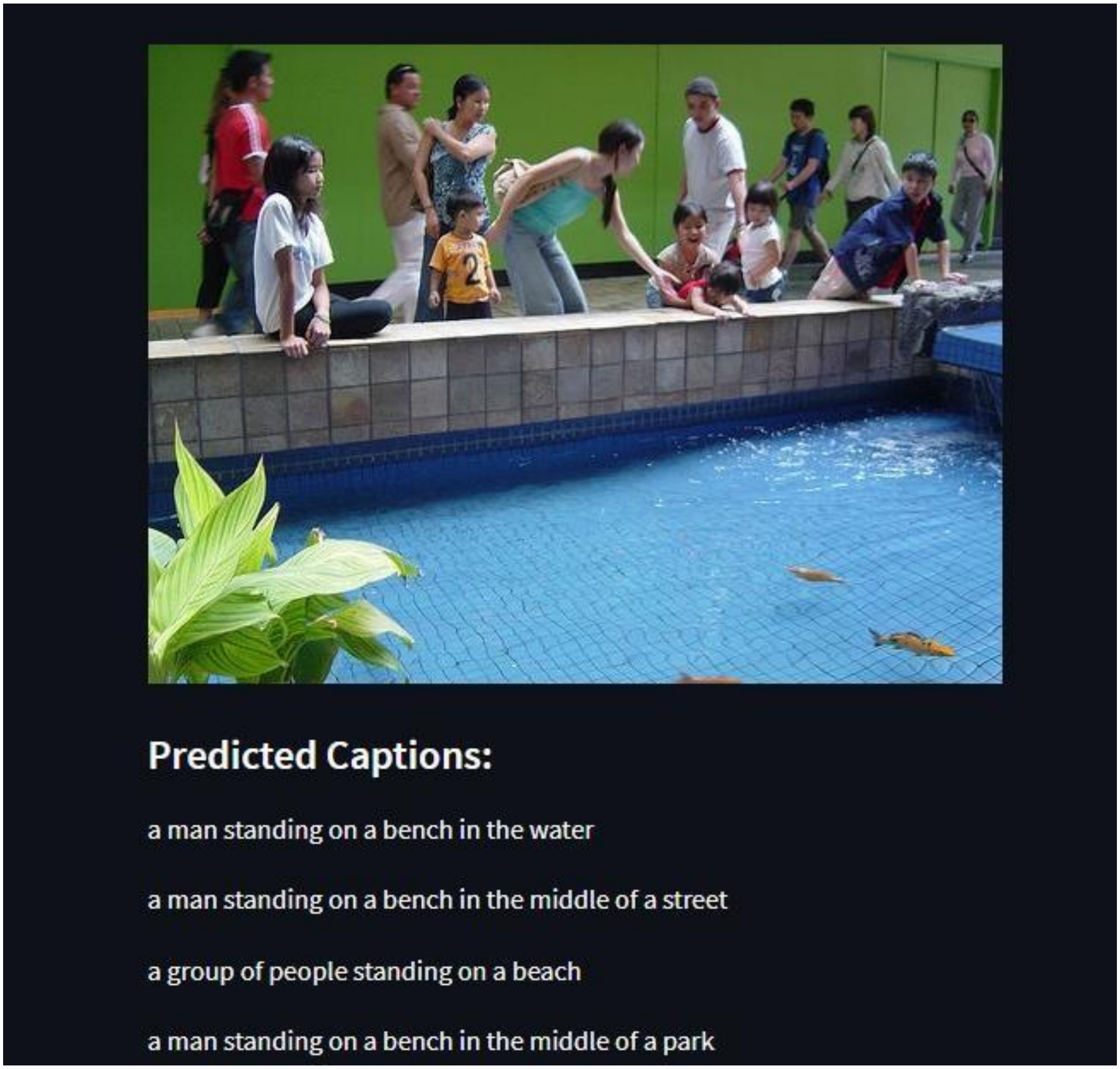

Fig32. HuggingFaceOutput1(AGroupOfPeople)

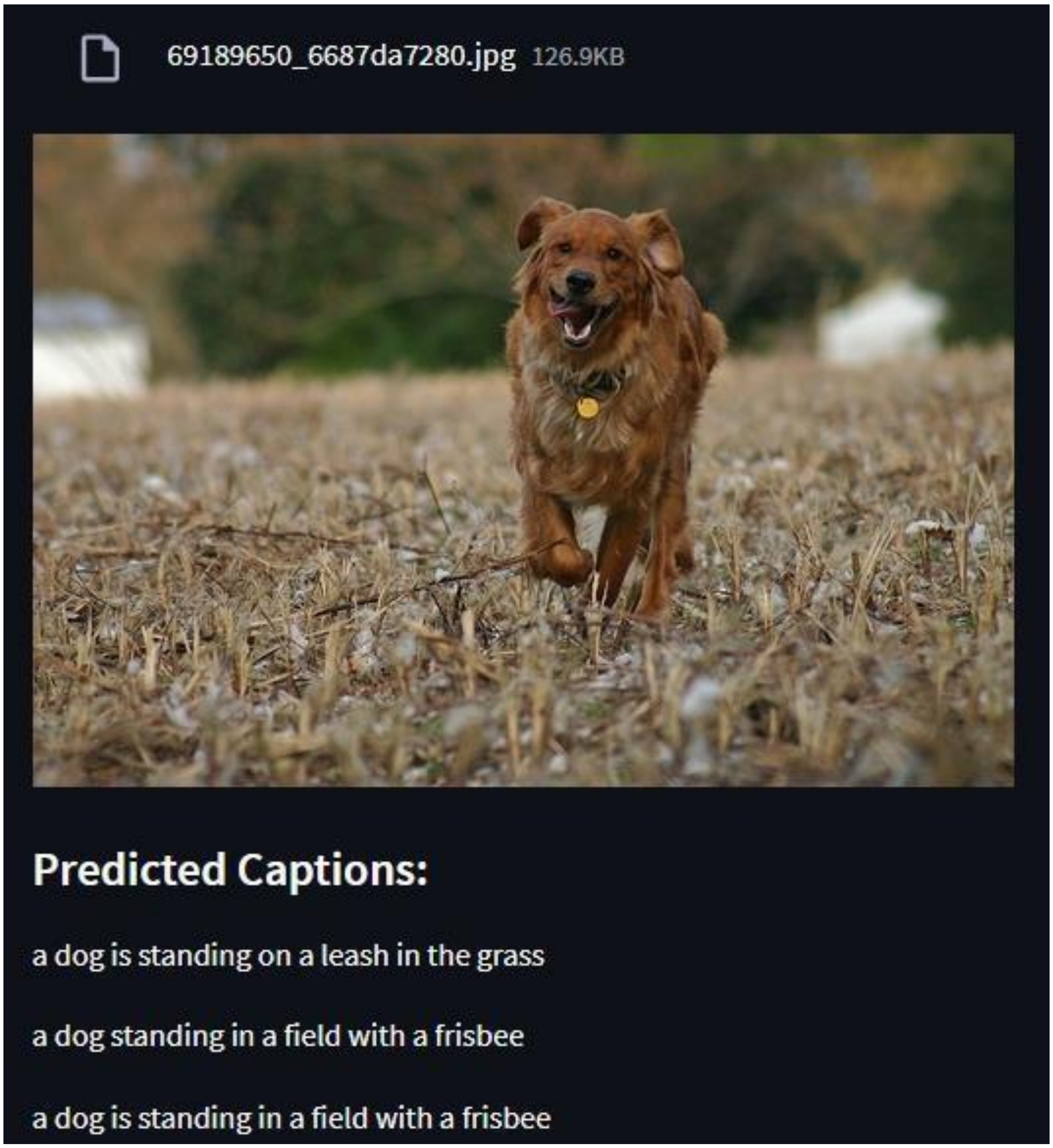

Fig33.HuggingFaceOutput2(Doginthe field)

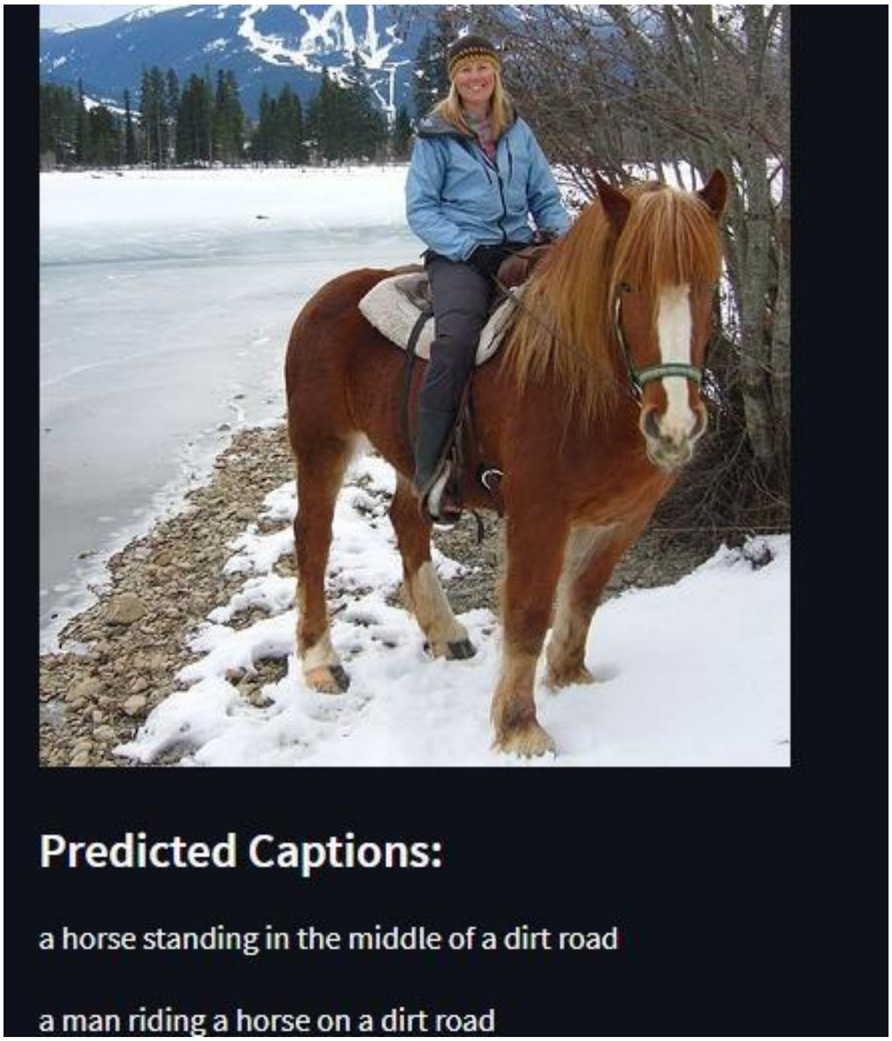

Fig34.HuggingFaceOutput3(WomenRidingHorse)

Chapter-6
ConclusionAndFutureScope

Conclusion

For the purpose of automatically producing captions for the input images, the CNN-LSTM model wasdeveloped.Therearemanycontextsinwhichthisideacanbeapplied.Bycreatinga CNN-LSTM model that can scan any input image, extract information from it, and convert it into a single, natural language sentence English, we were able to overcome earlier limitations onto the area of graphical image captioning.

Theprimarysubjectsofdiscussion werealgorithmattentionandtheapplicationoftheattention mechanism. We have managedto producea model that significantly outperformsthe previous image caption generator. By the use of hugging faces we are able to get output on an hosting element rather than on colab.

Deep learning-based image captioning has become a potent instrument for producing textual descriptions of images automatically, and it has found wide-ranging uses in fields like multimedia comprehension, assistive technology, image indexing, and content accessibility. The quality, diversity, and adaptability of generated captions should continue to improve as this field of study develops, opening the door to improved visual content comprehension and human-computer interaction.

KeyFindings

- Effective Representation Learning: Deep learning models have proven to be able to simultaneouslylearn effectiverepresentationsofbothvisualandtextualdata.Thisisespecially true for Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) networks for sequence generation.
- Contextual Understanding: Image captioning models based on deep learning can produce captions that demonstrate a contextual understanding of the images' content. These models capture complex relationships between objects, scenes, and actions shown in images by utilizing multimodal architectures and attention.

- Deep learning models that have been trained to caption images have some semantic understanding, which enables them to produce captions that are more complex than simple object recognition. With the help of these models, captions can be made more detailed and educational by inferring characteristics, connections, and evenabstract ideasfrom visual input.

FutureScope

Deep learning-based picture captioning has a wide future potential with many directions for research and development. The following are some important areas that show promise for additional study and advancement:

- Multimodal Understanding: Richer and morethorough understanding of visual content can be achieved by extending image captioning models to include multiple modalities, such as text, audio, and video. This involves creating models that can interpret intricate multimedia interactions, describe dynamic scenes, and create captions for videos.

- AttentionMechanisms: Byimprovingtheattentionmechanismsindeeplearningarchitectures, models will be better able to concentrate on pertinent image regions or temporal segments during the caption-generating process. It ispossible to expand attention mechanismsto handle occlusions, capture fine-grained details, and adapt to variousaspectsof theimage.

- Investigating the combination of generative adversarial networks (GANs) and image captioning can help produce captions that are more aesthetically pleasing and realistic. GANs can help reduce potential discrepancies between the generated text and the visual content, improve the perceptual quality of generated captions, and improvetheir visual fidelity.

- Investigatingmethodstofostercreativityandstorytellinginthecaptioningofimagescanresult in the creation of captions that are more captivatingand narratively focused.

- This entails creating models that can comprehend narrative structures and coherence in addition to incorporating personality, emotion,and stylistic variation into captions.

- Fine-GrainedUnderstandingandGeneration:Bystrengtheningmodels'abilitytorecognizeand characterizefine-graineddetails,attributes,andrelationshipswithinimages,thespecificityand accuracy of generated captions can be improved. This means addressing challenges related to object counting, spatial reasoning, andunderstandingthe background of complex scenes.

- Ethical Considerations and Bias Mitigation: In order to ensure an equitable and responsible deployment in real-world applications, it is imperative that image captioning models address ethical considerations such as bias, fairness, and inclusivity. This entails creating methods for reducing biases in training data, encouraging diversity and representation in the creation of captions, and encouraging accountability and transparency in the model-building process.

REFERENCES

- R.Subash(November2019):AutomaticImageCaptioningUsingConvolutionNeural Networks and LSTM.
- Seung-HoHan,Ho-JinChoi(2020):Domain-SpecificImageCaptionGenerator withSemantic Ontology.
- PranayMathur, Aman Gill, Aayush Yadav, Anurag Mishra and Nand Kumar Bansode (2017): Camera2Caption: A Real-Time Image Caption Generator
- SimaoHerdade,ArminKappeler,KofiBoakye,JoaoSoares(June2019):ImageCaptioning: Transforming Objects into words.
- ManishRaypurkar,AbhishekSupe,PratikBhumkar,PravinBorse,Dr.ShabnamSayyad(March 2021): Deep learning-based Image Caption Generator.
- OriolVinyals,AlexanderToshev,SamyBengio,DumitruErhan(2015):ShowandTell:A Neural Image Caption Generator.
- Jianhui Chen, Wenqiang Dong, Minchen Li (2015): Image Caption Generator based on Deep Neural Networks
- 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.
- JyotiAneja,AdityaDeshpande,andAlexanderGSchwing(2018):Convolutionalimage captioning.
- ShuangBaiandShanAn(2018):Asurveyonautomaticimagecaptiongeneration.
- D.Bahdanau,K.Cho, and Y.Bengio(2015): Neuralmachine translationbyjointly learning to align and translate.
- K.Xu, J.Ba, K.Cho, and R.Salakhutdinov (2018): Show attend and tell: Neural image caption generator with visual attention.
- M.Pedersoli,T.Lucas,C.Schmid,andJ.Verbeek(2017):Areasofattentionforimage captioning.
- H.R.Tavakoli,R.Shetty,B.Ali,andJ.Laaksonen(2017):Payingattentiontodescriptions generated by image captioning models.
- A.Matthews, L.Xie, and X.He(2018): Sem Style-learning to generate stylized image captions using unaligned text.
- C.Park,B.Kim,andG.kim(2018):Towardspersonalizedimagecaptioningviamultimodal memory networks.
- X.Chen,MaLin,W.Jiang,J.Yao,andW.Liu(June2018):RegularizingRNNsforcaption generation by reconstructing the past with the present

- T.Yao,Y.Pan,Y.Li,Z.Qiu,andT.Mei(June2016):Boostingimagecaptioningwithattributes.
- Deep Learning in big dataAnalytics: A comparative study-Scientific Figure on ResarchGate (2021).
- Kaustavetal.(June2016):AFacialExpressionRecognitionSystemtopredictEmotions.
- M. Hodosh, P. Young and J. Hockenmaier (2013) "Framing Image Description as a Ranking Task
- OriolVinyals,AlexanderToshev,SamyBengio,DumitruErhanShowandTell:ANeural Image Caption Generator
- CS231nWinter2016Lesson10RecurrentNeuralNetworks,ImageCaptioningandLSTM

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _________________________ __**Department:** ________________ **Enrolment No** ________

Contact No. ____________________________**E-mail.** ____________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _______________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com