A

Major Project

On

AUTOMATIC GESTURE RECOGNITION AND IMAGE CAPTION GENERATOR

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

B. KAWSHIK (197R1A05K0)

G. SAI MAHESH (197R1A05K9)

M.SAI KUMAR (197R1A05M8)

Under the Guidance of

J. NARASIMHARAO

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road,

Hyderabad-501401

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATION

This is to certify that the Project entitled "**AUTOMATIC GESTURE RECOGNITION AND IMAGE CAPTION GENERATOR**" being submitted by **B.KAWSHIK (197R1A05K0),G.SAI MAHESH (197R1A05K9) &M.SAI KUMAR (197R1A05M8**) in partial fulfillment of the requirements for the award of degree B. TECH Computer science and engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bondafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

J. NARASIMHA RAO
(Associate Professor)
INTERNAL GUIDE

DR.A. RAJI REDDY
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **J. Narasimharao,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Ms. Shilpa, Dr. T. Subha Mastan Rao & J. Narasimharao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering, **Dr. Ashuthosh Saxena,** Dean R&D, and **Dr. D T V Dharmajee Rao,** for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion ofthis project.

B. KAWSHIK (197R1A05K0)

G.SAI MAHESH (197R1A05K9)

M.SAI KUMAR (197R1A05M8)

ABSTRACT

Caption generation is a difficult synthetic intelligence problem wherein a textual description maybe generated for a given photograph. On every occasion when a photograph seems in front of people, their mind is capable of annotating or labeling it, however how can computer structures procedure and label it with a noticeably relevant and correct caption? So we will train the computers using deep learning algorithms and make it easy. With advanced deep learning techniques, accessibility of big datasets and computer power we can build an efficient model to generate captions. The strategies used right here are **LSTM (Long ShortTerm Memory) and CNN (Convolutional Neural Networks).** Feature extraction is done first and then captions are generated. The flickr_8k dataset is used for training the model. The dataset which we are using contains 8000 images and each image is mapped with five different captions

DOMAIN: DEEP LEARNING

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

Using Natural Language Processing (NLP) to make a computer system discover and describe items is an age-old artificial intelligence problem. Image caption generation is an image processing activity that recognises the context of an image and describes it in plain language. This image caption generator will generate the caption from a trained model that has been trained using algorithms and on a big dataset based on the picture file we provide or upload. It may be a difficult challenge to automatically describe the substance of a photograph using properly constructed English sentences, but it might have a significant impact, such as assisting visually impaired persons in better understanding the image online. We come across a wide variety of things every day.

1.2 PROJECT PURPOSE

Captions for every picture at the net can result in quicker and descriptively correct photos searches and indexing. Since researchers began working on image recognition, it has been clear that simply providing the names of the items recognised does not make the same kind of impression as a detailed human-like description. Natural language descriptions will remain a problem to be solved as long as machines do not think, speak, or behave like humans. Image captioning is used in a variety of sectors, including biomedicine, trade, internet searching, and the military. Instagram, Facebook, and other social media platforms can produce captions for photos automatically.

1.3 PROJECT FEATURES

Image caption generator is a project that uses computer vision and natural language processing ideas to comprehend the context of an image and explain it in natural language like English. We might have utilised CNN (Convolutional Neural Networks) and LSTM to generate captions for this Python-based application (Long Short Term Memory). The photograph features will be taken from VGG16, a CNN version trained on the image net dataset, and then fed into the LSTM model, which will be responsible for creating the photograph captions.Convolutional neural networks are

deep neural networks that have been customised to process data in the form of a second matrix Photographs are easy. It is able to handle the photos that have been translated, circled, scaled and modifications in angle. Long ShortTerm Memory (LSTM) is a form of RNN (Recurrent Neural Network) that excels at sequence prediction. Based on the prior textual content, we can estimate what the next phrase will be. LSTM may perform relevant statistics throughout the input processing and, using an overlook gate, it can eliminate irrelevant data. So that it will be easy for the user to recognise the image in less time. The flickr_8k dataset is used for training the model. The dataset which we are using contains 8000 images and each image is mapped with five different captions.These applications in image captioning have important theoretical and practical research value. Image captioning is a more complicated but meaningful task in the age of artificial intelligence. Given a new image, an image captioning algorithm should output a description about this image at a semantic level. In this an Image caption generator, basis on our provided or uploaded image file It will generate the caption from a trained model which is trained using algorithms and on a large dataset. The main idea behind this is that users will get automated captions when we use or implement it on social media or on any applications.

1.4 INTRODUCTION TO PYTHON

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking C in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s.

Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners, Guido van Rossum said: " In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people

know ABC's influence on Python. I try to mention ABC's influence because I am indebted to everything I learned during that project and to the people who worked on it
that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

1.4.1 PYTHON DEVELOPMENT STEPS

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one obvious way to do it. "Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e., int. long is int as well.

• The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behavior.
• Text Vs. Data Instead of Unicode Vs. 8-bit.

1.4.2 PURPOSE

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –
• Machine Learning
• GUI Applications (like Kivy, Tkinter, PyQt etc.)
• Web frameworks like Django (used by YouTube, Instagram, Dropbox)
• Image processing (like Opencv, Pillow)
• Web scraping (like Scrapy, BeautifulSoup, Selenium)
• Test frameworks
• Multimedia

1.4.3 ADVANTAGES OF PYTHON

1. Extensive Libraries

Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we do not have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn**,** understand**,** and code**.** This is why when people pick up Python, they have a hard time adjusting to other more

verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory**.** This further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available**.** But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it is not the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA)**.** However, you need to be careful enough not to include any system dependent features.

ADVANTAGES OF PYTHON OVER OTHER LANGUAGES

1. Less Coding

Almost all the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you do not

have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

1.4.4 DISADVANTAGES OF PYTHON

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client side**.** Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle**.** The reason it is not so famous despite the existence of Python is that it isn't that secure.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analysed . The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

A general statement of the proposed model is to generate the relevant natural language caption to the given input image, instead of just describing a single target object the model detects multiple target objects for generating grammatically correct caption. We have used flickr_8k data set, the working of the model.

2.2 EXISTING SYSTEM

Image captioning has gotten a lot of attention recently, especially in the natural language area. The necessity for natural language image descriptions with context is critical. While this may appear far-fetched, recent advancements in domains like neural networks, computer vision, and natural language processing have paved the path for effectively characterising images, or capturing their visually grounded meaning.We're using cutting-edge approaches like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), as well as associated image and humanperceived description datasets, to do this. We show that our alignment approach works in retrieval tests on datasets like Flickr.

2.2.1 DRABACKS OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Can describe only one single Target object. So that, we can't get accurate correct caption.
- Not used advance approaches at present.
- It only gives caption to single frame.
- It is only for images.

2.3 PROPOSED SYSTEM

The proposed model is to generate the relevant natural language caption to the given input image, instead of just describing a single target object the model detects multiple target objects for generating grammatically correct caption .We have used flickr_8k data set .

The working of the model involves the following steps:

1.The given input image is pre-processed.

2. **Convolution neural network** (CNN) is used to extract features from the provided input image.

3.The information from the CNN is used by the LSTM for generating the relevant caption for the given image.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- In this project, instead of just describing a single target object, this model detects multiple target objects and generating grammatically correct caption.
- Higher accuracy &stability.
- Quicker time to unlock a device.
- Can convey a range of Accurate emotions.

2.4 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system.

The following are some hardware requirements:

- Processor – i5 and above (64-bit OS).
- Memory – 4GB RAM (Higher specs are recommended for high performance).
- Input devices – Keyboard, Mouse

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system.

The following are some software requirements:

- Windows/Mac.
- Kaggle Notebook
- Python3
- CNN and LSTM algorithm
- NumPy, Keras, Tensor Flow libraries

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final output.

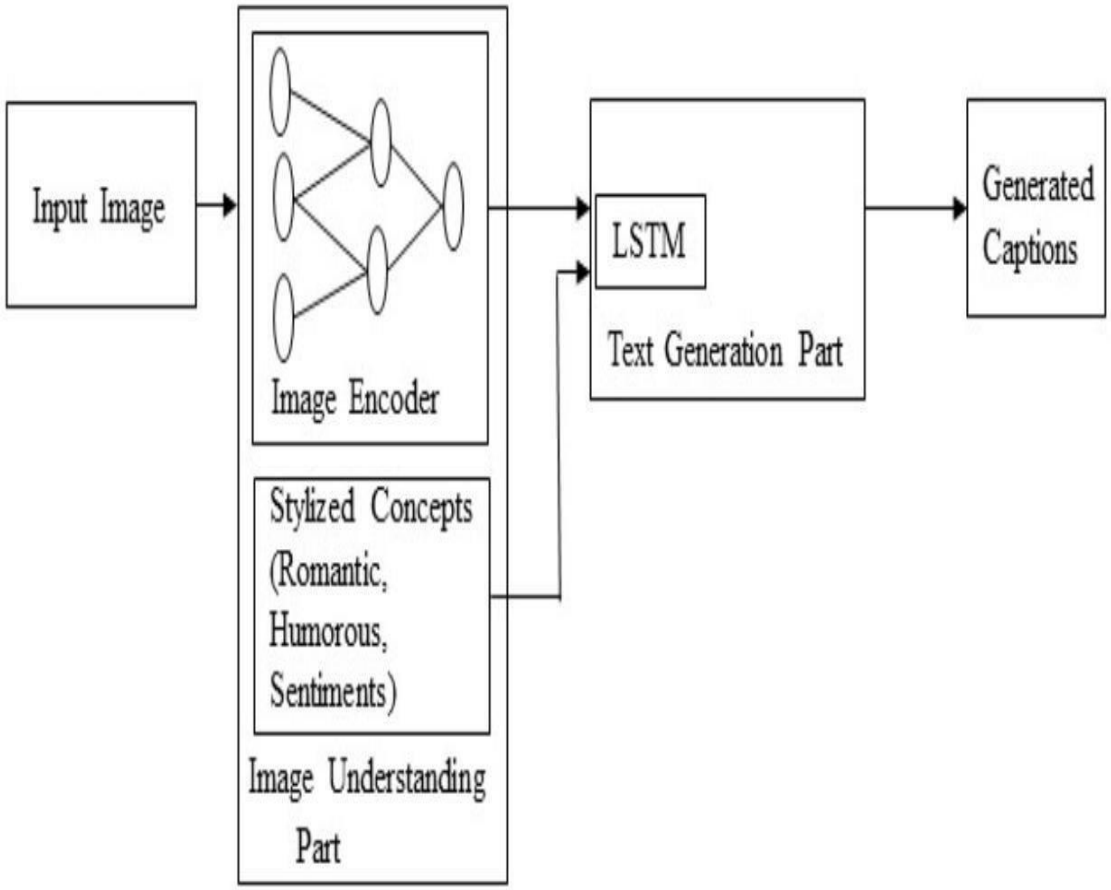


Figure 3.1: Project Architecture For Automatic Gesture Recognition And Image Caption Generator

3.2 DESCRIPTION

This project is totally based upon to generate the relevant natural language caption to the given input image, instead of just describing a single target object the model detects multiple target objects for generating grammatically correct caption. We have used flickr_8k data set .The given input image is pre-processed. **Convolution neural network** (CNN) is used to extract features from the provided input image .The information from the **CNN** is used by the **LSTM** for generating the relevant caption for the given image. The proposed model is to generate the relevant natural language caption to the given input image, instead of just describing a single target object the model detects multiple target objects for generating grammatically correct caption. We have used flickr_8k data set, the working of the model. These applications in image captioning have important theoretical and practical research value. Image captioning is a more complicated but meaningful task in the age of artificial intelligence. Given a new image, an image captioning algorithm should output a description about this image at a semantic level. In this an Image caption generator, basis on our provided or uploaded image file It will generate the caption from a trained model which is trained using algorithms and on a large dataset. The main idea behind this is that users will get automated captions when we use or implement it on social media or on any applications.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model. A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

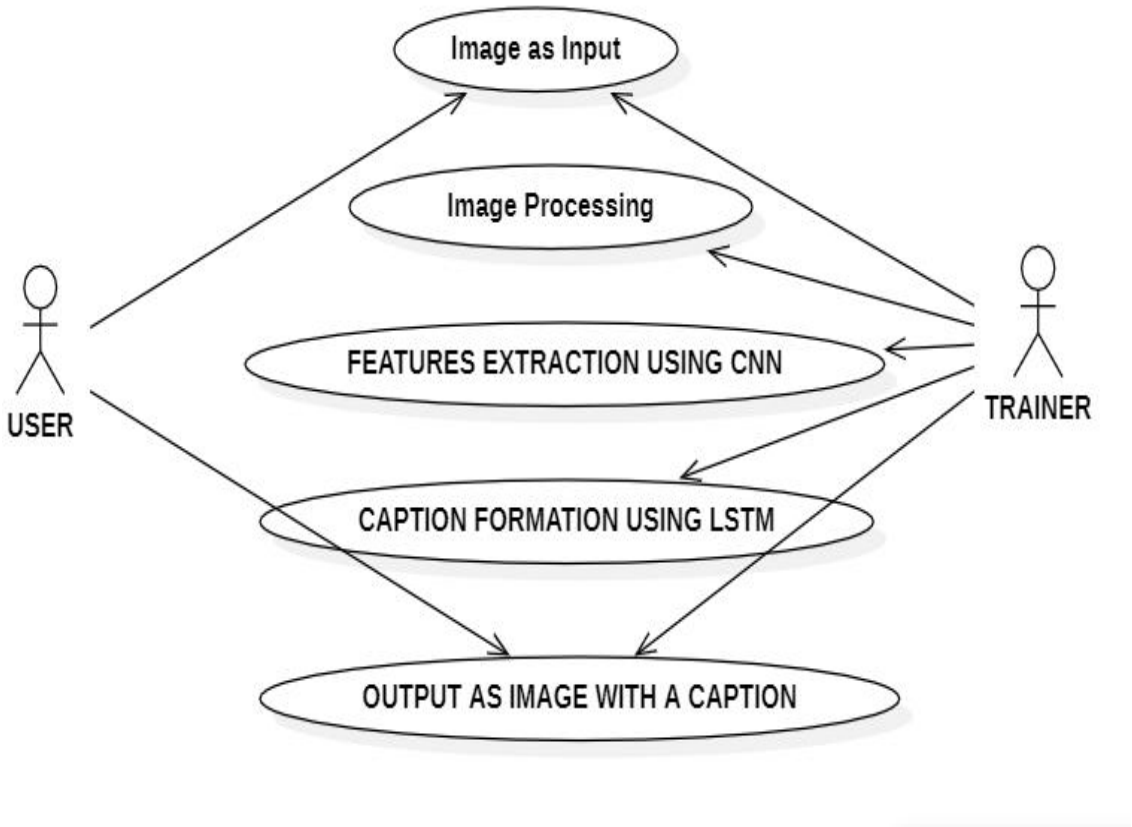


Figure 3.2: Use Case Diagram For Automatic Gesture Recognition And Image Caption Generator

3.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(methods),and the relationships among objects.

Figure 3.3: Class Diagram For Automatic Gesture Recognition And Image Caption Generator

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

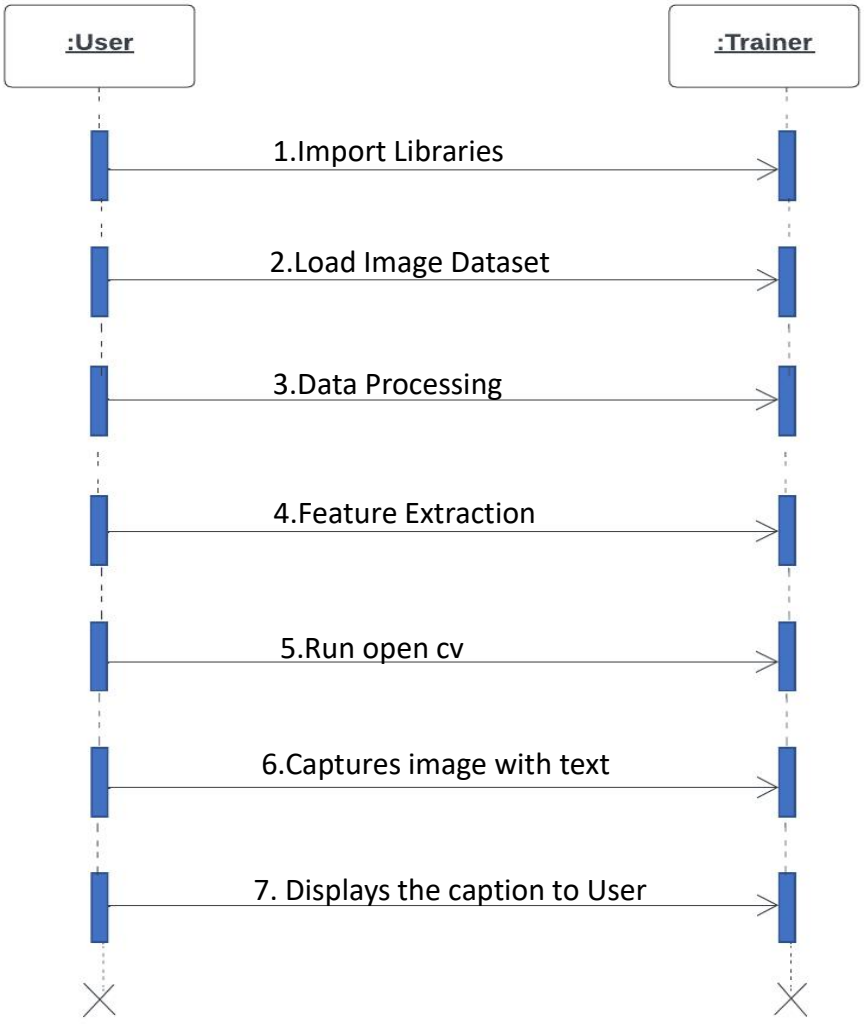


Figure 3.4: Sequence Diagram For Automatic Gesture Recognition And Image Caption Generator

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of work flows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

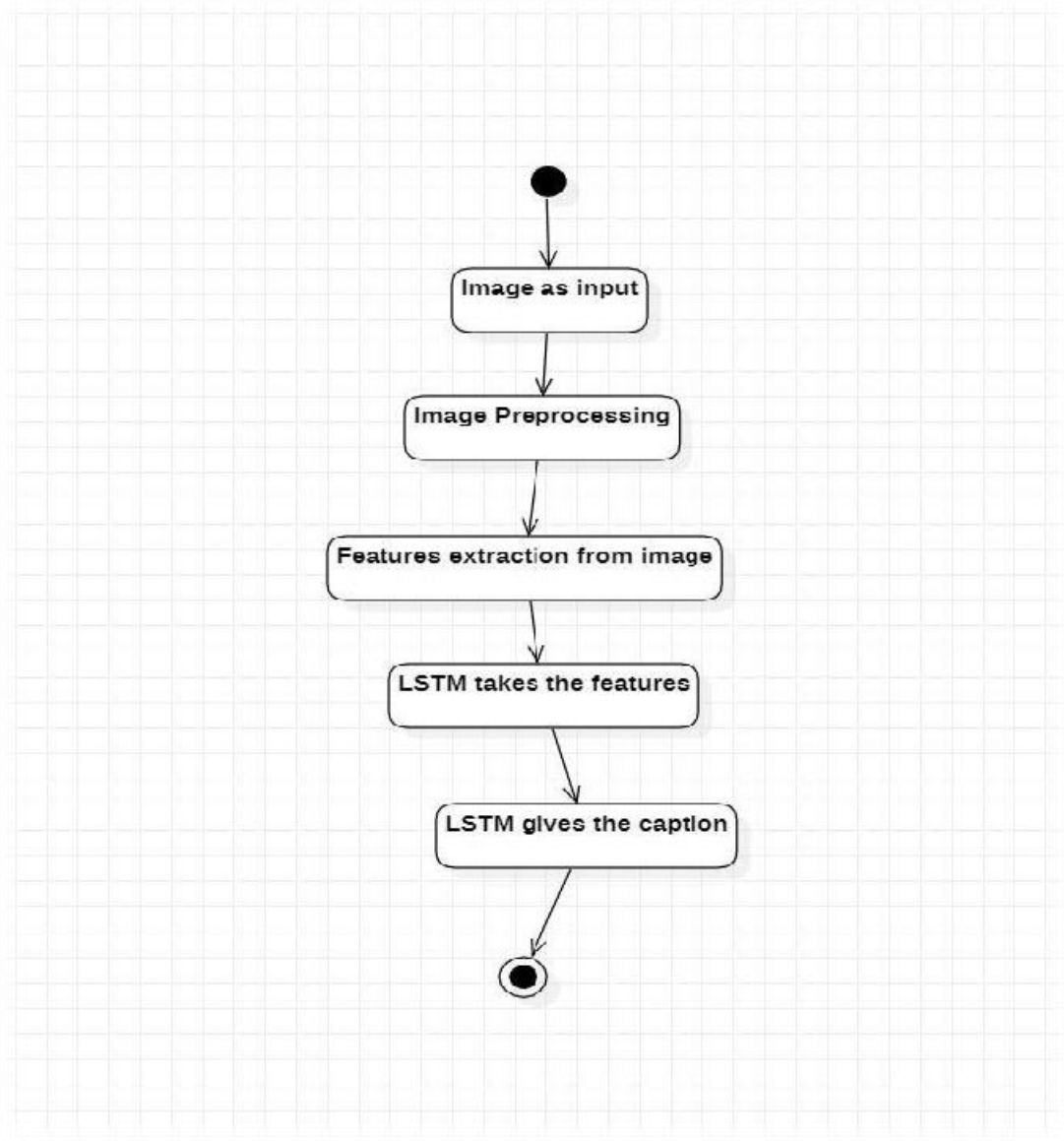


Figure 3.5: Activity Diagram For Automatic Gesture Recognition And Image Caption Generator

4.IMPLEMENTATION

4.1 SAMPLE CODE

1)import os
import pickle
import numpy as np
from tqdm.notebook import tqdm

from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Model
from tensorflow.keras.utils import to_categorical, plot_model
from tensorflow.keras.layers import Input, Dense, LSTM, Embedding, Dropout, add

2)BASE_DIR = '/kaggle/input/flickr8k'
WORKING_DIR = '/kaggle/working'

Extract Image Features

3)# load vgg16 model
model = VGG16()
# restructure the model
model = Model(inputs=model.inputs, outputs=model.layers[-2].output)
# summarize
print(model.summary())

4)features = {}
directory = os.path.join(BASE_DIR, 'Images')

for img_name in tqdm(os.listdir(directory)):
    # load the image from file
    img_path = directory + '/' + img_name
    image = load_img(img_path, target_size=(224, 224))
    # convert image pixels to numpy array
    image = img_to_array(image)
    # reshape data for model
    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))
    # preprocess image for vgg
    image = preprocess_input(image)
    # extract features
    feature = model.predict(image, verbose=0)
    # get image ID
    image_id = img_name.split('.')[0]

```
    # store feature
    features[image_id] = feature

5)pickle.dump(features, open(os.path.join(WORKING_DIR, 'features.pkl'), 'wb'))

6)with open(os.path.join(WORKING_DIR, 'features.pkl'), 'rb') as f:
    features = pickle.load(f)
```

Load the Captions Data

```
7)with open(os.path.join(BASE_DIR, 'captions.txt'), 'r') as f:
    next(f)
    captions_doc = f.read()
8)mapping = {}
# process lines
for line in tqdm(captions_doc.split('\n')):
    # split the line by comma(,)
    tokens = line.split(',')
    if len(line) < 2:
        continue
    image_id, caption = tokens[0], tokens[1:]
    # remove extension from image ID
    image_id = image_id.split('.')[0]
    # convert caption list to string
    caption = " ".join(caption)
    # create list if needed
    if image_id not in mapping:
        mapping[image_id] = []
    # store the caption
    mapping[image_id].append(caption)

9)len(mapping)

10)def clean(mapping):
    for key, captions in mapping.items():
        for i in range(len(captions)):
            # take one caption at a time
            caption = captions[i]
            # preprocessing steps
            # convert to lowercase
            caption = caption.lower()
            # delete digits, special chars, etc.,
            caption = caption.replace('[^A-Za-z]', '')
            # delete additional spaces
            caption = caption.replace('\s+', ' ')
            # add start and end tags to the caption
```

```
        caption = 'startseq ' + " ".join([word for word in caption.split() if len(word)>1]) +
' endseq'
        captions[i] = caption
```

11)mapping['1000268201_693b08cb0e']

12)clean(mapping)

13)mapping['1000268201_693b08cb0e']

```
14)all_captions = []
for key in mapping:
    for caption in mapping[key]:
        all_captions.append(caption)
```

15)len(all_captions)

16)all_captions[:10]

```
17)tokenizer = Tokenizer()
tokenizer.fit_on_texts(all_captions)
vocab_size = len(tokenizer.word_index) + 1
```

18)vocab_size

```
19)max_length = max(len(caption.split()) for caption in all_captions)
max_length
```

```
20)image_ids = list(mapping.keys())
split = int(len(image_ids) * 0.90)
train = image_ids[:split]
test = image_ids[split:]
```

```
21)def data_generator(data_keys, mapping, features, tokenizer, max_length, vocab_size,
batch_size):
    # loop over images
    X1, X2, y = list(), list(), list()
    n = 0
    while 1:
        for key in data_keys:
            n += 1
            captions = mapping[key]
            # process each caption
            for caption in captions:
                # encode the sequence
```

```
            seq = tokenizer.texts_to_sequences([caption])[0]
            # split the sequence into X, y pairs
            for i in range(1, len(seq)):
                # split into input and output pairs
                in_seq, out_seq = seq[:i], seq[i]
                # pad input sequence
                in_seq = pad_sequences([in_seq], maxlen=max_length)[0]
                # encode output sequence
                out_seq = to_categorical([out_seq], num_classes=vocab_size)[0]

                # store the sequences
                X1.append(features[key][0])
                X2.append(in_seq)
                y.append(out_seq)
        if n == batch_size:
            X1, X2, y = np.array(X1), np.array(X2), np.array(y)
            yield [X1, X2], y
            X1, X2, y = list(), list(), list()
            n = 0
```

Model Creation

```
22)inputs1 = Input(shape=(4096,))
fe1 = Dropout(0.4)(inputs1)
fe2 = Dense(256, activation='relu')(fe1)
# sequence feature layers
inputs2 = Input(shape=(max_length,))
se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
se2 = Dropout(0.4)(se1)
se3 = LSTM(256)(se2)

# decoder model
decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
outputs = Dense(vocab_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.compile(loss='categorical_crossentropy', optimizer='adam')

plot_model(model, show_shapes=True)

23)epochs = 20
batch_size = 32
steps = len(train) // batch_size

for i in range(epochs):
```

```
    # create data generator
    generator = data_generator(train, mapping, features, tokenizer, max_length,
vocab_size, batch_size)
    # fit for one epoch
    model.fit(generator, epochs=1, steps_per_epoch=steps, verbose=1)

24)model.save(WORKING_DIR+'/best_model.h5')
```

Generate Captions for the Image

```
25)def idx_to_word(integer, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == integer:
            return word
    return None

26)def predict_caption(model, image, tokenizer, max_length):
    # add start tag for generation process
    in_text = 'startseq'
    # iterate over the max length of sequence
    for i in range(max_length):
        # encode input sequence
        sequence = tokenizer.texts_to_sequences([in_text])[0]
        # pad the sequence
        sequence = pad_sequences([sequence], max_length)
        # predict next word
        yhat = model.predict([image, sequence], verbose=0)
        # get index with high probability
        yhat = np.argmax(yhat)
        # convert index to word
        word = idx_to_word(yhat, tokenizer)
        # stop if word not found
        if word is None:
            break
        # append word as input for generating next word
        in_text += " " + word
        # stop if we reach end tag
        if word == 'endseq':
            break

    return in_text

27)from nltk.translate.bleu_score import corpus_bleu
# validate with test data
actual, predicted = list(), list()
```

```
for key in tqdm(test):
    # get actual caption
    captions = mapping[key]
    # predict the caption for image
    y_pred = predict_caption(model, features[key], tokenizer, max_length)
    # split into words
    actual_captions = [caption.split() for caption in captions]
    y_pred = y_pred.split()
    # append to the list
    actual.append(actual_captions)
    predicted.append(y_pred)

# calcuate BLEU score
print("BLEU-1: %f" % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
print("BLEU-2: %f" % corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
```

28)

```
from PIL import Image
import matplotlib.pyplot as plt
def generate_caption(image_name):
    # load the image
    # image_name = "1001773457_577c3a7d70.jpg"
    image_id = image_name.split('.')[0]
    img_path = os.path.join(BASE_DIR, "Images", image_name)
    image = Image.open(img_path)
    captions = mapping[image_id]
    print('----------------------Actual----------------------')
    for caption in captions:
        print(caption)
    # predict the caption
    y_pred = predict_caption(model, features[image_id], tokenizer, max_length)
    print('--------------------Predicted--------------------')
    print(y_pred)
    plt.imshow(image)
```

29)

```
generate_caption("1001773457_577c3a7d70.jpg")
```

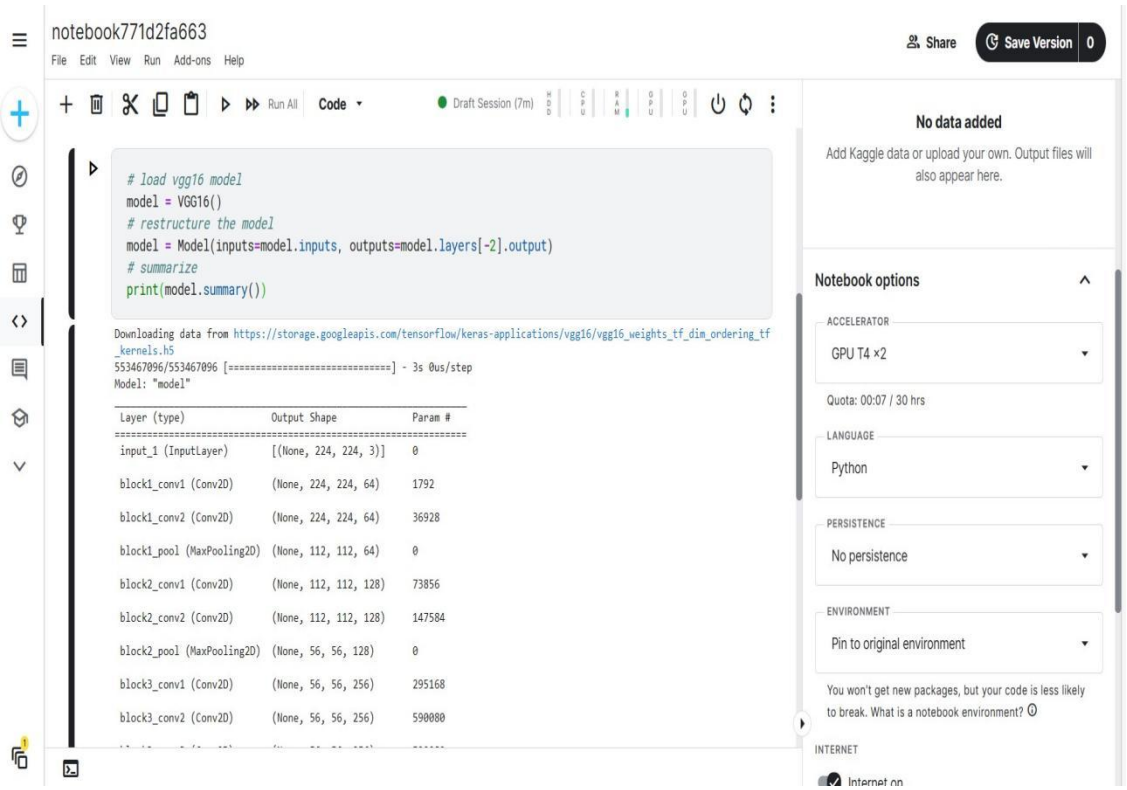
30)

```
generate_caption("1002674143_1b742ab4b8.jpg")
```

5. RESULTS

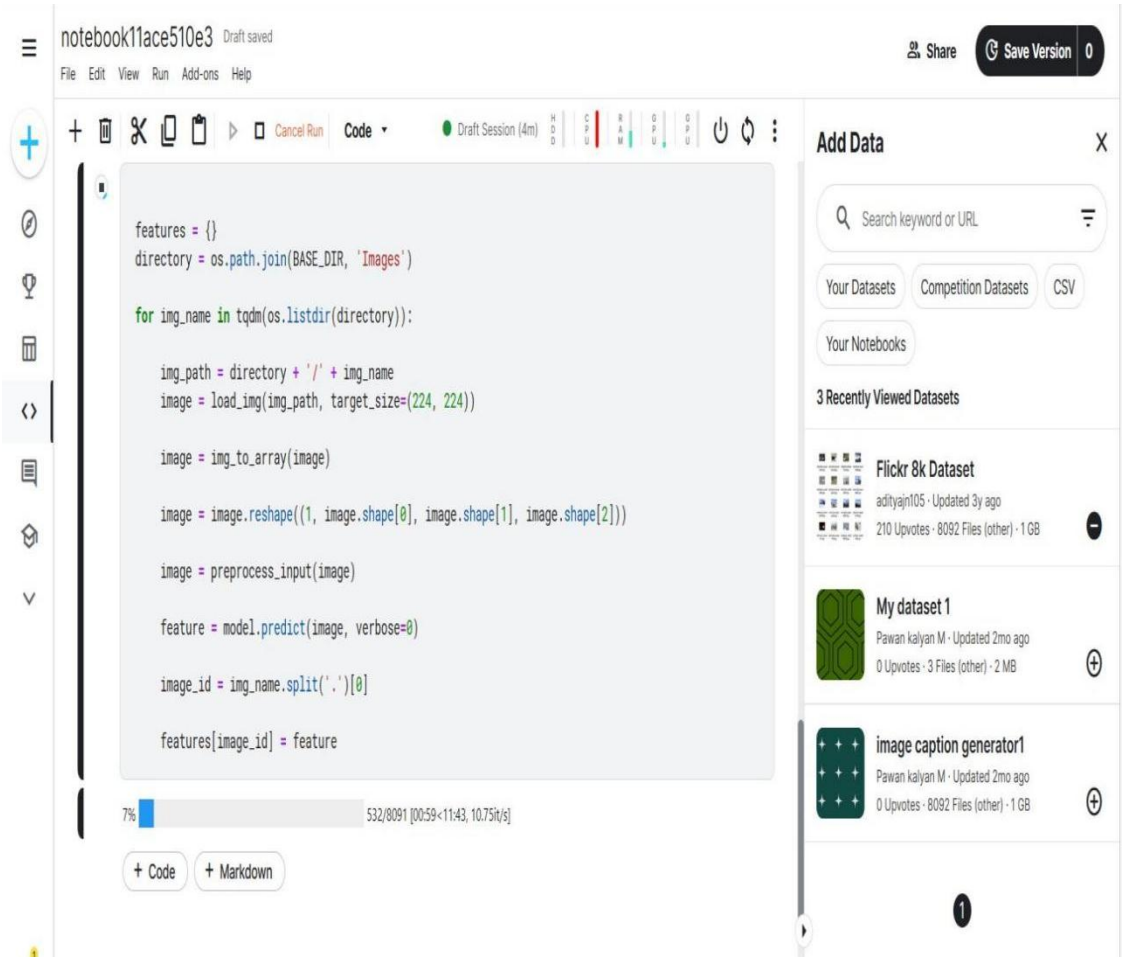
5. RESULTS

In the below Picture, the training model getting Downloading Data and Exacting Features .



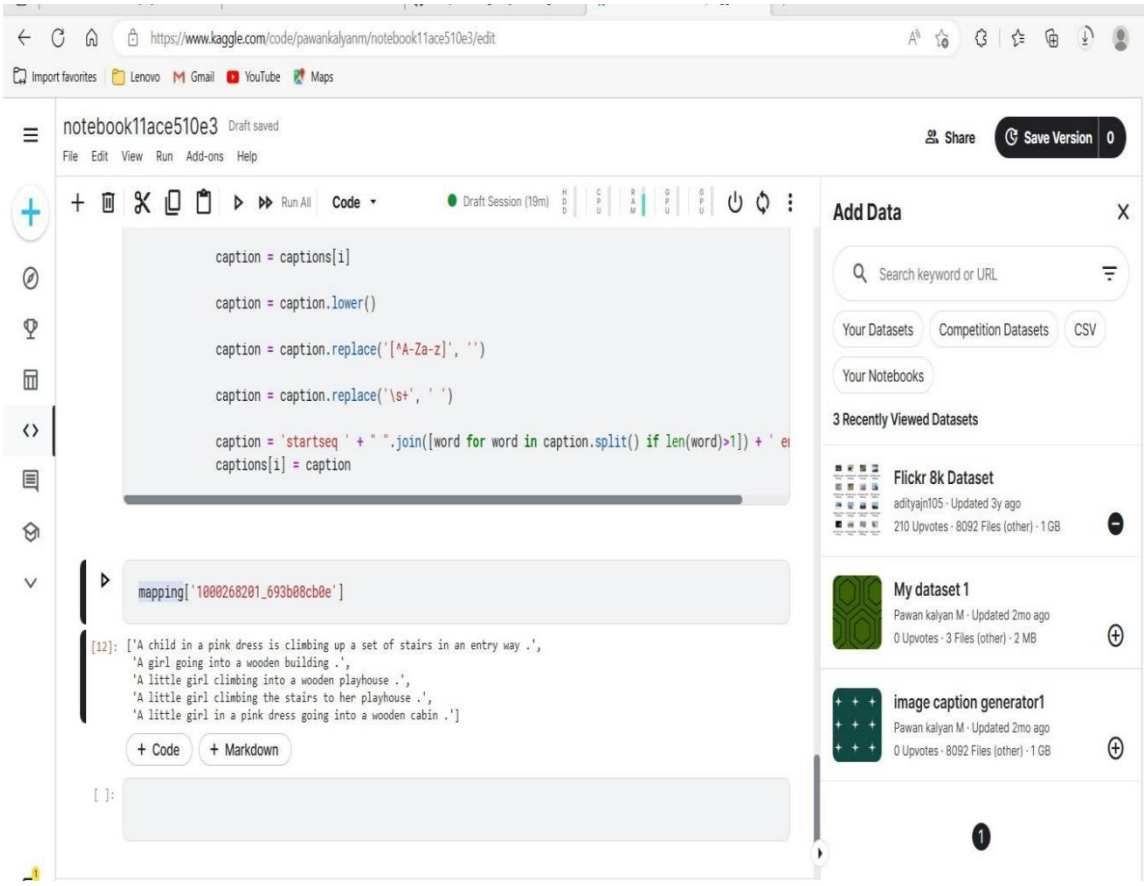
Screenshot 5.1: Exacting Features

In the below Picture, the training model Downloaded Data with Exacted Features and getting extracting the images.



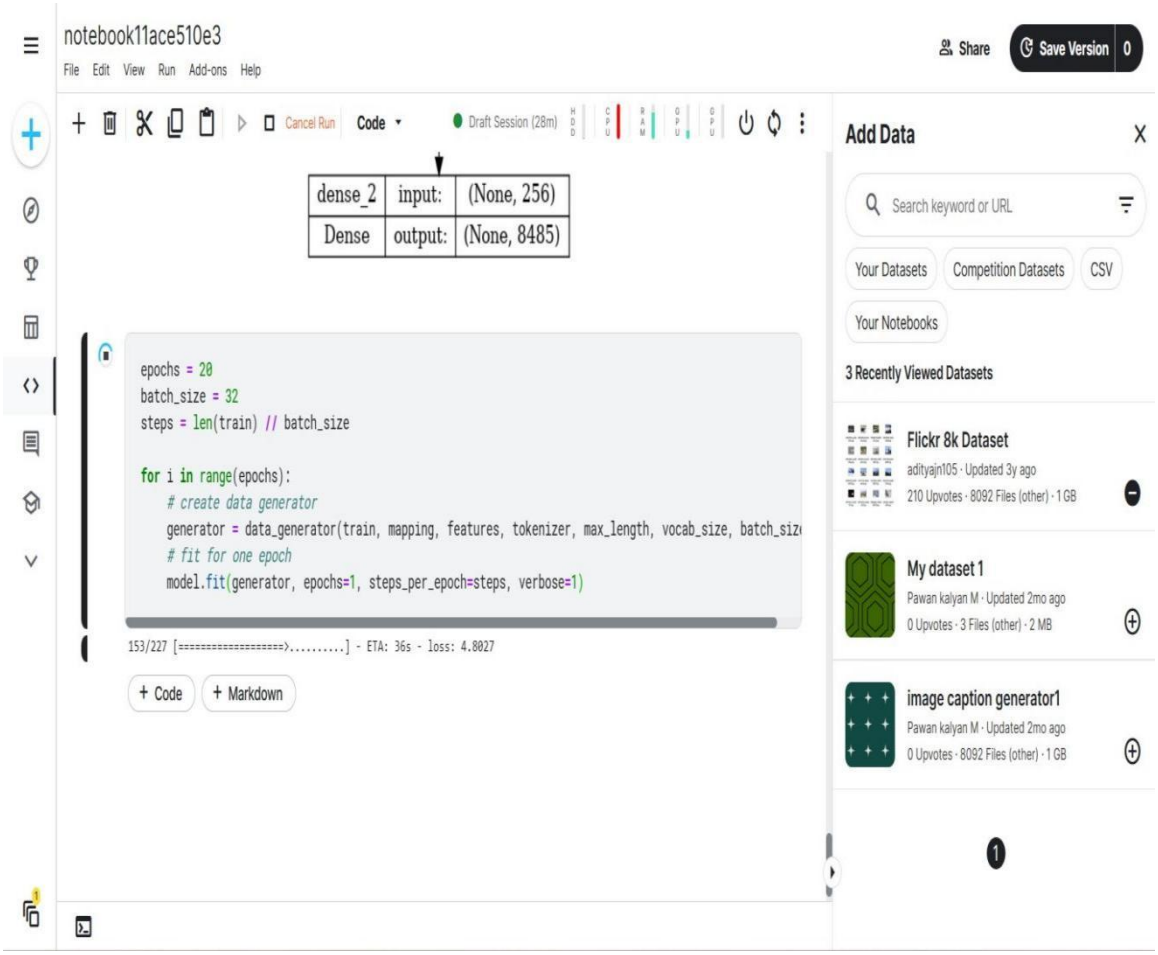
Screenshot 5.2: Exacting Images

In the below Picture, the training model Downloaded Data with Exacted Features and extracted the images from that model gets extracted the Text Data through Mapping .



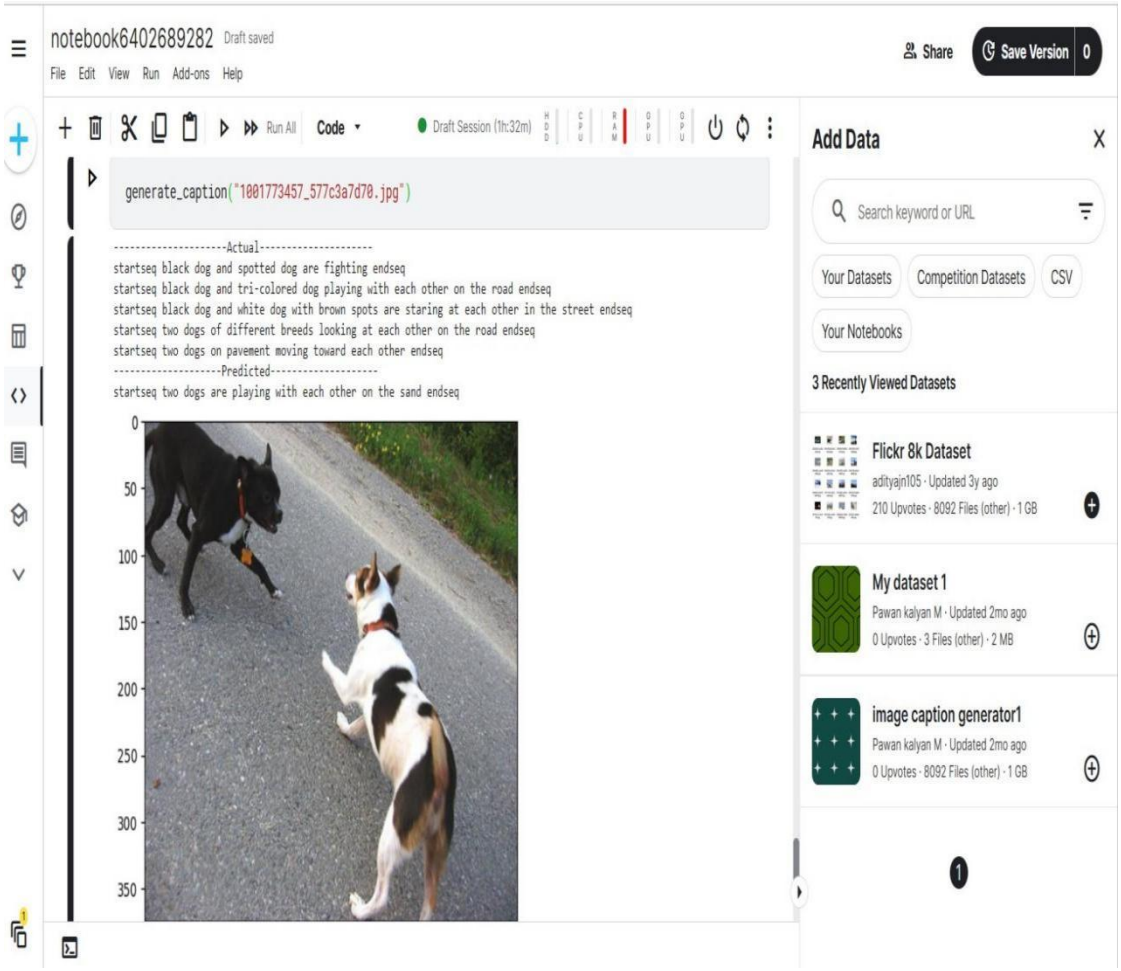
Screenshot 5.3: Pre -Processing Text Data

In the below Picture, the training model Downloaded Data with Exacted Features and extracted the images from that model gets extracted the Text Data through Mapping and through this model getting trained.



Screenshot 5.4: Model Getting Trained

In the below picture ,the model completely ready and it has given the accurate output with the downloaded data and generated captions.



Screenshot 5.5: Generation Of Caption To The Trained Model

6.TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Individual software modules or components are reviewed during unit testing, which is a sort of software testing. The goal is to make sure that every line of software code works. Unit testing is done by developers throughout the application development (code) process. Unit tests are used to isolate and verify that code is accurate. A single function, method, process, module, or object is referred to as a unit.

In the SDLC, STLC, and V Models, unit testing is the initial level of testing performed before integration testing. Unit testing is a type of WhiteBox testing performed by programmers. In the actual world, QA engineers perform testing due to time constraints or developers' dislike of it.

6.2.2 Integration Testing:

Integration testing is used to ensure that modules/components perform as intended when they are combined, i.e. to ensure that modules that work well separately do not cause problems when combined. When it comes to testing large applications with the black box testing technique, it entails the use of a number of modules that are tightly associated with one another. We may use the concepts of integration testing to test these types of scenarios.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, orspecial test cases.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	output
1	Gesture Processing Detection.	To detect Whether given image is real or fake.	The user gives the input in the form of image to the Trained Model.	An output is a text message for given image to the model.
2	Image Processing Detection.	To detect Whether given image is real or fake.	The user gives the input in the form of image to the Trained Model.	An output is a text message for given image to the model.

7.CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Different Deep learning methods for caption generation have been reviewed in this paper. Generating an appropriate and grammatically correct caption in a natural language like a human is a difficult task ,this task involves feature extraction and natural language processing concepts. the data set that has been used for this model is flickr_8k data set which consists around eight thousand images, for the given image the context of the image is recognized by targeting multiple objects. We have understood the architectures and wide ranging applications of CNN and LSTM .As the users are increasing day by day the image captioning has a vast scope in the future, the model is built over a dataset of eight thousand images to improve the accuracy and performance this can be implemented on the higher dataset.

7.2 Future enhancements

Image captioning is used in many applications including Google photos, Skin vision, Adobe photoshop, social media etc. It has recently become a significant issue, and we have explored several image generating strategies in the past, understand the process of image extractionand addressed the challenges faced by them, to improve the accuracy by extracting the features of the context of the image larger data set can be used. This project can be enhanced in the future by using CPUs and GPUs for achieving faster, cheaper and efficient algorithms, instead of limiting to the one language i.e , English the caption can be generated in different languages as required by the user.

8.BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1REFERENCES

[1] S. ALBAWI and T. A. MOHAMMED, "Understanding of a Convolutional Neural Network,"in ICET, Antalya, 2017.

[2] S. Hochreiter, "LONG SHORT-TERM MEMORY," Neural Computation, December 1997.

[3] O. Vinyals, A. Toshev, S. Bengio and D. Erhan,"A Neural Image Caption Generator," CVPR
2015 Open Access Repository, vol. Xiv, 17 November 2014.

[4] D. S. Whitehead, L. Huang, H. and S.-F. Chang, "Entityaware Image CaptionGeneration,"inEmpirical Methods in Natural Language Processing, Brussels, 2018.

[5] C. Elamri and T. Planque, "Automated Neural Image Caption Generator for Visually Impaired
People," California, 2016.

[6] G. Ding, M. Chen, S. Zhao, H. Chen, J. Han and Q. Liu, "Neural Image Caption Generation
with Weighted Training and Reference," Cognitive Computation, 08 August 2018.

[7] J. Chen, W. Dong and M. Li, "Image Caption Generator Based On Deep Neural Networks,"
March 2018. [8] S. Bai and S. An,"A Survey on Automatic Image Caption Generation,"
Neurocomputing, 13 April 2018. [9] R. Staniute and D. Sesok, "A Systematic Literature Review
on Image Captioning," Applied Sciences, vol. 9, no. 10, 16 March 2019.

[8] J. Hessel, N. Savva and M. J. Wilber, "Image Representations and New Domains in Neural
Image Captioning," ACL Anthology, vol. Proceedings of the Fourth Workshop on Vision and
Language, p. 29–39, 18 September 2015.

[9] M. Z. Hossain, F. SOHEL, M. F. SHIRATUDDIN and H. LAGA, "A Comprehensive

Survey of Deep Learning for Image Captioning," ACM Journals, vol. 51, no. 6, 14 Octomber 2018.

[10] A. Farhadi, . M. Hejrati, . M. A. Sadeghi and . P. Young, "Every Picture Tells a Story: Generating Sentences from Images," in ACM Digital Library, 2010. for Visual Recognition and Description," CVPR 2015, vol. 14, 31 May 2016

8.2 GHITHUB LINK:

kawshikbhyroju/Major-Project (github.com)

9. PAPER PUBLICATION

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 21st Feb 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 03

10.48047/IJIEMR/V12/ISSUE 02/88

Title DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR

Volume 12, ISSUE 02, Pages: 573-579

Paper Authors
B.Kawshik, G.Sai Mahesh, M.Sai kumar,
Jonnadula Narasimharao

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR

B.Kawshik[1], G.Sai Mahesh[2], M.Sai kumar[3], Jonnadula Narasimharao[4]

[1,2,3] B.Tech Student, Department of Computer Science and Engineering, CMR Technical Campus,Medchal, Hyderabad,Telangana, India,
[1]kbhyroju@gmail.com, [2]gajula797@gmail.com, [3]saikumar10684@gmail.com,
[4]Associate Professor, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Hyderabad, Telangana, India,
jonnadula.narasimharao@gmail.com

ABSTRACT: In the past few years, the problem of generating descriptive sentences automatically for images has garnered a rising interest in natural language processing and computer vision research. An image caption is something that describes an image in the form of text. It is widely used in programs where one needs information from any image in automatic text format. Image captioning is a fundamental task which requires semantic understanding of images and the ability of generating description sentences with proper and correct structure. With the exponential development in the field of artificial intelligence in recent years, many researchers have focused their attention towards the topic of image caption generation. With advanced deep learning techniques, accessibility of big datasets and computer power one can build an efficient model to generate captions. Hence, in this work Deep Learning based Automated image Caption Generator is presented. The model is trained in such a way that if input image is given to model it generates captions which nearly describes the image. In this approach, two deep learning algorithms like LSTM (Long Short Term Memory) and CNN (Convolutional Neural Networks) are used. Feature extraction is done first and then captions are generated. The flickr_8k dataset is used for training the model. The dataset which we are using contains 8000 images and each image is mapped with five different captions.

KEYWORDS: Image Caption Generator, Convolutional Neural Network (CNN), Long Short Term Network (LSTM).

I. INTRODUCTION

Image captioning is the process of generating a textual description of an image that aims to describe the salient parts of the given image.

Image caption includes the multi-level use of image information. From the target in the image, the relationship between the targets, to the description of the image, and the construction of the scene graph, all belong to the category of image description research. Each task in image caption has great research value and great practical application value [4].

Caption generation is an interesting artificial intelligence problem where a descriptive sentence is generated for a given image. It involves the dual techniques from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order. Image captioning has various applications such as recommendations in editing applications, usage in virtual assistants, for image indexing, for visually impaired persons, for social media, and several other natural language processing applications.

Automatic caption generation is a tough undertaking that can aid visually challenged persons in understanding the content of web images. It may also have a significant impact on search engines and robots. This problem is substantially more difficult than image categorization or object recognition, both of which have

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

been extensively researched [2].Gesture recognition becomes a popular analytics tool for extracting the characteristics of user movement and enables numerous practical applications in the biometrics field. Despite recent advances in this technique, complex user interaction and the limited amount of data pose serious challenges to existing methods [1].

With the growing advancement in in-depth reading techniques, the availability of large databases, and the ability to integrate, models are often developed to produce image captions. Image captioning is a process that involves image processing and natural language processing concepts to identify the context of an image and interpret it in a natural language such as English or any other language. Image caption generator incorporates the concept of reconfiguring an image and interpreting it in the native language such as English [3].

Recent advancements in language modeling and object recognition have made image captioning an essential research area in computer vision and natural language processing. Caption generation of an image has a great impact by helping visually impaired people to better understand the contents on the web [5].

Image caption methods can be divided into template-based methods, retrieval-based methods and deep learning-based methods. The template-based method first obtains some visual concepts for the image, and then generates a sentence through sentence templates, syntactic rules, or combined methods. Retrieval-based methods usually need to save a large database, and then obtain a sentence or a group of sentences through image retrieval, and then obtain a complete image description. The image description based on deep learning mainly uses the structure of codec to complete the image caption task. Further, the effect of image description can be improved through the attention machine or other methods of enhancing the deep learning model.

Recently, deep learning methods have achieved state-of the-art results on examples of this problem. It has been demonstrated that deep learning models are able to achieve optimum results in the field of caption generation problems. Hence in this work, deep learning based automated image caption generator is presented. The rest of the work s organized as follows: The section II demonstrates literature survey. The section III presents deep learning based automated image caption generator.The section IV evaluates the result analysis of presented approach. Finally the work is concluded in section V.

II. LITERATURE SURVEY

Xiangqing Shen, Bing Liu, Yong Zhou & Jiaqi Zhao et. al., [6] describes Remote sensing image caption generation via transformer and reinforcement learning. A new model using the Transformer to decode the image features to target sentences is presented. Reinforcement Learning is then applied to enhance the quality of the generated sentences. We demonstrate the validity of our proposed model on three remote sensing image captioning datasets. This model obtains all seven higher scores on the Sydney Dataset and Remote Sensing Image Caption Dataset (RSICD), four higher scores on UCM dataset, which indicates that the proposed methods perform better than the previous state of the art models in remote sensing image caption generation.

Nayan Mehta, Suraj Pai,Sanjay Singh et. al., [7] describes Automated 3D sign

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

language caption generation for video. This paper aims to present a useful technology that can be used to leverage online resources and make them accessible to the hearing-impaired community in their primary mode of communication. First, the video gets converted to text via subtitles and speech processing methods. The generated text is understood through NLP algorithms and then mapped to avatar captions which are then rendered to form a cohesive video alongside the original content. We validated our results through a 6-month period and a consequent 2-month study, where we recorded a 37% and 70% increase in performance of students taught using Sign captioned videos against student taught with English captioned videos.

Chetan Amritkar, Vaishali Jabade et. al., [8] describes Image Caption Generation using Deep Learning Technique.This model is used to generate natural sentences which eventually describe the image. This model consists of Convolutional Neural Network(CNN) as well as Recurrent Neural Network(RNN). The CNN is used for feature extraction from image and RNN is used for sentence generation. The model is trained in such a way that if input image is given to model it generates captions which nearly describes the image. The accuracy of model and smoothness or command of language model learns from image descriptions are tested on different datasets. These experiments show that model is frequently giving accurate descriptions for an input image.

Philip Kinghorn, Li Zhang, Ling Shao et. al., [9] presents A region-based image caption generator with refined descriptions. A novel region-based deep learning architecture for image description generation is presented. It employs a regional object detector, recurrent neural network (RNN)-based attribute prediction, and an encoder–decoder language generator embedded with two RNNs to produce refined and detailed descriptions of a given image. Most importantly, the proposed system focuses on a local based approach to further improve upon existing holistic methods, which relates specifically to image regions of people and objects in an image. Evaluated with the IAPR TC-12 dataset, the presneted system shows impressive performance and outperforms state-of-the-art methods using various evaluation metrics

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio et. al., [10] describes Neural Image Caption Generation with Visual Attention. An attention based model is described that automatically learns to describe the content of images. Authors described how this model is trained in a deterministic manner using standard back-propagation techniques and stochastically by maximizing a variational lower bound. They also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. They validate the use of attention with state-of-theart performance on three benchmark datasets: Flickr9k, Flickr30k and MS COCO.

III. DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR

In this section, Deep learning based automated image caption generator is presented. The block diagram of presented approach is shown in Fig. 1.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

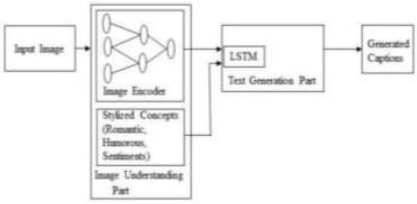


Fig. 1:Architecture of Deep learning based automated image caption generator

Image captioning isa more complicated but meaningful task in the age of artificial intelligence. In this an Image caption generator, basis on the provided oruploaded image file it will generate the caption from a trained model which is trainedusing algorithms and on a large dataset. The main idea behind this is that users will get automated captions when they use or implement it on social media or on any applications.

This project is totally based upon to generate the relevant natural languagecaption to the given input image, instead of just describing a single target object themodel detects multiple target objects for generating grammatically correct caption. Wehave used flickr_8k data set .The given input image is pre-processed.The flickr_8k dataset is used for training the model. The dataset which weare using contains 8000 images and each image is mapped with five differentcaptions.These applications in image captioning have important theoretical andpractical research value. Image captioning is a more complicated but meaningful task inthe age of artificial intelligence.

Given a new image, an image captioning algorithmshould output a description about this image at a semantic level. In this an Imagecaption generator, basis on our provided or uploaded image file It will generate thecaption from a trained model which is trained using algorithms and on a large dataset.The main idea behind this is that users will get automated captions when we use orimplement it on social media or on any applications.

Data preprocessing is done in this step includes Data cleaning, Data reduction, Image data preparation. For instance, punctuations, digits, single length words are removed from the text dataset. Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. Feature extraction helps to reduce the amount of redundant data from the data set. In the end, the reduction of the data helps to build the model with less machine effort and also increases the speed of learning and generalization steps in deep learning process.

Two deep learning models have been selected i.e, CNN and LSTM. Firstly, CNN takes image as input and extract features such as background, objects in the image.In this approach, CNN (Convolutional Neural Networks)and LSTM are used to generate captions for this Python-based application (Long Short TermMemory). The photograph features will be taken from VGG16, a CNN version trainedon the image net dataset, and then fed into the LSTM model, which will be responsible for creating the photograph captions. Convolutional neural networks are deep neural networks that have been customized to process data in the form of a second matrixPhotographs are easy. It is able to handle the photos that have been translated, circled,scaled and modifications in angle.

Long ShortTerm Memory (LSTM) is a form of RNN(Recurrent Neural Network) that excels at sequence prediction. Based on the prior textual content, one can estimate what the next phrase will be.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

LSTM may performrelevant statistics throughout the input processing and, using an overlook gate, it caneliminate irrelevant data. So that it will be easy for the user to recognize the image inless time.

Convolutionneural network (CNN) is used to extract features from the provided input image .Theinformation from the CNN is used by the LSTM for generating the relevant caption forthe given image. The proposed model is to generate the relevant natural languagecaption to the given input image, instead of just describing a single target object themodel detects multiple target objects for generating grammatically correct caption.

IV. RESULT ANALYSIS

Deep learning based automated image caption generator is implemented in this section. The result analysis of presented approach is evaluated in this section.This approach includes different phases, feature extraction, image extraction, data pre-processing and image caption generation. These process implemented results are shown as follows: The Fig. 2 shows the exacting features.

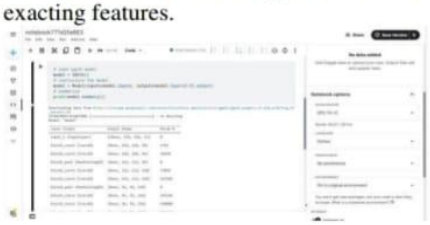


Fig. 2:Features Extraction Process

The Fig. 3 shows the images extraction phase.

Fig. 3: Image Extraction Process

The Fig. 4 shows the data pre-processing phase implementation.

Fig. 4: Data-Pre-processing Process results

The Fig. 5 shows the training phase of presented approach.

Fig. 5: Training Phase

The Fig. 5 shows the output of trained presented mode i.e. image caption generation.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Fig. 5: Image Caption Generation

Hence, this approach has generated the captions for different images very effectively and accurately.

V. CONCLUSION

In this work,Deep learning based automated image caption generator is presented.Generating an appropriate and grammatically correct caption in a naturallanguage like a human is a difficult task ,this task involves feature extraction and natural language processing concepts. CNN and LSTM have worked well together in synchronization, they were able to find a connection between objects in pictures.In this analysis, The flickr_8k dataset is used for training the model which contains 8000 images and each image is mapped with five differentcaptions. The Convolutional Neural Network is used to extract features from the provided input image and Long Short Term memory algorithm is used to generate the relevant caption forthe given image. From the results,it can beconcluded that this model can be used to generate image captions for multiple images in real time.

VI. ACKNOWLEDGEMENT

We thank CMR Technical Campus for supporting this paper titled "Deep Learning based Automated Image Caption Generator", which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans, Head Of the Department, Department Of Computer Science and Engineering, Guide and Teaching and Non- Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work

VII. REFERENCES

[1]Hao Zhou, Wei Huang, Zhuo Xiao, Shichuan Zhang, Wangzhan Li, Jinhui Hu, Tianxing Feng, Jing Wu, Pengcheng Zhu, Yanchao Mao, "Deep-Learning-Assisted Noncontact Gesture-Recognition System for Touchless Human-Machine Interfaces", Advanced Functional Materials, 2022,doi:10.1002/adfm.202208271

[2]SaiTeja. N.R, Rashmitha Khilar, "Implementing Complexity in Automatic Image Caption Generator using Recurrent Neural Network over Long Short-Term Memory", Journal of Pharmaceutical Negative Results, Volume13,Special Issue , 2022

[3] Peerzada Salman syeed, Dr.Mahmood Usman, "Image Caption Generator UsingDeep Learning", Neuroquantology, October 2022, Volume 20, Issue 12, Page 2682-2691, Doi: 10.14704/Nq.2022.20.12.Nq77261

[4] SiZhen Li, Linlin Huang, "Context-based Image Caption using Deep Learning", 2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing (ICSP 2021), DOI: 10.1109/ICSP51882.2021.9408871

[5] Santosh Kumar Mishra, Rijul Dhir, Sriparna Saha, And Pushpak Bhattacharyya, "A Hindi Image Caption Generation Framework Using Deep Learning", Trans. Asian Low-Resour. Lang. Inf. Process. 20, 2, Article 32 (March 2021), 19 pages, doi:10.1145/3432246

[6] Xiangqing Shen, Bing Liu, Yong Zhou & Jiaqi Zhao, "Remote sensing image caption generation via transformer and reinforcement learning", Multimedia Tools and Applications,

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

volume 79, pages26661–26682 (2020), doi: 10.1007/s11042-020-09294-7

[7] Nayan Mehta, Suraj Pai,Sanjay Singh, "Automated 3D sign language caption generation for video",Universal Access in the Information Society volume 19, pages725–738 (2020)

[8] Chetan Amritkar, Vaishali Jabade, "Image Caption Generation using Deep Learning Technique",2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018 IEEE, 978-1-5386-5257-2/18

[9]Philip Kinghorn, Li Zhang, Ling Shao, "A region-based image caption generator with refined descriptions",Neurocomputing, Volume 272, 10 January 2018, Pages 416-424, Elsevier, Doi:10.1016/j.neucom.2017.07.014

[10] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio, "Neural Image Caption Generation with Visual Attention", Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

10. CERTIFICATES

International Journal for Innovative Engineering and Management Research

IMPACT FACTOR 7.812

(ISSN 2456 - 5083)

CERTIFICATE

ELSEVIER SSRN

DOI: 10.48047/IJIEMR/V12/I02/88

This is to Certify that Prof./Dr./Mr./Jonnadula Narasimharao From CMR Technical Campus,Medchal, Hyderabad,Telangana, India. Participated in the International Journal of Engineering and Management Research.Presented a Paper Entitled"DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-02 Feb-2023

Hyderabad 2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com

International Journal for Innovative Engineering and Management Research

IMPACT FACTOR 7.812

(ISSN 2456 - 5083)

CERTIFICATE

ELSEVIER SSRN

DOI: 10.48047/IJIEMR/V12/I02/88

This is to Certify that Prof./Dr./Mr./B.KawshikFrom CMR Technical Campus,Medchal, Hyderabad,Telangana, India. Participated in the International Journal of Engineering and Management Research.Presented a Paper Entitled"DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-02 Feb-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com

International Journal for Innovative Engineering and Management Research

IMPACT FACTOR 7.812

(ISSN 2456 - 5083)

CERTIFICATE

ELSEVIER SSRN

DOI: 10.48047/IJIEMR/V12/I02/88

This is to Certify that Prof./Dr./Mr./G.Sai Mahesh From CMR Technical Campus,Medchal, Hyderabad, Telangana, India. Participated in the International Journal of Engineering and Management Research. Presented a Paper Entitled"DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-02 Feb-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com

International Journal for Innovative Engineering and Management Research

IMPACT FACTOR 7.812

(ISSN 2456 - 5083)

CERTIFICATE

ELSEVIER SSRN

DOI: 10.48047/IJIEMR/V12/I02/88

This is to Certify that Prof./Dr./Mr./M.Sai kumar From CMR Technical Campus,Medchal, Hyderabad, Telangana, India. Participated in the International Journal of Engineering and Management Research. Presented a Paper Entitled"DEEP LEARNING BASED AUTOMATED IMAGE CAPTION GENERATOR"In the Organizing Commitee of ROYAL SOCIETY FOR SCIENCE.

Published in International Journal of Innovation Engineering and Management Research (IJIEMR), Vol-12,Issue-02 Feb-2023

Hyderabad
2023

RAKESH ORUGANTI
Editor In Chief

+91 9989292561 Huda Techo Enclave, HiTech City Madhapur, Hyderabad, 500081 info.ijiemr@gmail.com