

Generating Image Captions based on Deep Learning and Natural language Processing

by

M. Bhargavi

Roll No. 204G1A0521

G. Jasmin

Roll No. 204G1A0542

P. Manjusha

Roll No. 204G1A0552

A. Anulekha Sai

Roll No. 214G5A0504

Under the guidance of

Mrs. P. Rohini M. Tech

Assistant Professor



Department of Computer Science and Engineering

Srinivasa Ramanujan Institute of Technology

(Autonomous)

2023 - 2024

Contents

- ✍ Abstract
- ✍ Introduction
- ✍ Literature survey
- ✍ Existing System
- ✍ Proposed System
- ✍ Planning
- ✍ Design
- ✍ Implementation
- ✍ Conclusion
- ✍ Research Paper
- ✍ References

Abstract

Humans and computers are attempting to communicate because everything in today's society depends on systems like computers, mobile phones, etc. This is how our project is visualized. Our undertaking People with visual impairments can benefit from the creation of image captions. Computers are unable to distinguish objects, things, or activities with the same ease as humans. To recognize them, they require some training. The suggested method is used to identify activities or similar items. We offer several deep neural network-based models for creating captions for images, with a particular emphasis on CNNs (Convolutional Neural Networks) that extract characteristics from the image.

Continue....

Using LSTM (Long Short-Term Memory) techniques, RNNs (Recurrent Neural Networks) create captions based on the image's attributes. and examining how they affect the construction of sentences. Here, encoder-decoders are used to create a link between descriptions from natural language processing and visual information such as image features. The process of generating a caption's sequence is handled by the decoder, while the encoder extracts features. In order to determine which feature extraction and encoder model produces the best results and accuracy, we have also created captions for sample photos and compared them with one another. We also introduce Deep Voice, a text-to-speech system of production quality that uses only deep neural networks to generate captions based on visual attributes. The evaluation of our project will be conducted utilizing several machine learning methods and Python.

Introduction

- It is relatively easy for humans to describe the environments they are living. It is normal for a human to be able to quickly describe a vast amount of information about an image. This is a fundamental human ability. The ability to identify objects and describe images is facilitated by the human brain. Artificial Intelligence introduces numerous algorithms that are based on the architecture of the brain. To allow a computer to automatically explain an image, Image captioning came into picture.
- The existing image captioning systems face challenges in generating accurate and contextually relevant descriptions, often resulting in inadequate or inaccurate captions.
- The main problem addressed by this project is the development of an innovative system to produce comprehensive and precise image captions by exploiting both global and specific image features. Also, provide the audio for the captions.

Literature survey

[1]. In this paper, deep neural networks have enabled the captioning of images. Based on the dataset, the photo caption generator assigns a suitable title to an applied input image. The proposed study suggests a deep learning-based model and applies it to produce a caption for the input image. The model uses both CNN and LSTM algorithms. This CNN model recognizes the objects in the picture, and the Long Short-Term Memory (LSTM) algorithm generates text as well as caption that fits the project. Thus, the proposed model combines both object recognition and title generation for the input images.

[2]. In this paper, The keras framework's TensorFlow backend has been utilized in this study's model evaluation. Utilizing assessment measures that were appropriate for the problem's nature allowed for an understanding of how The model has made correct predictions. This paper presents the results of mathematical computations performed on the confusion matrix.

Literature survey

[3]. The proposed system for the project “Design and implementation of text to speech conversion for visually impaired people” is a sophisticated and versatile solution that developed a useful text-to-speech synthesizer in the form of a simple application that converts inputted text into synthesized speech and reads out to the user.

[4]. The main difficulties in this research include identifying the objects in an image and their characteristics, which are challenging computer vision problems, as well as figuring out how the objects interact and what relationships exist between them. Automatic image description is not without its difficulties. To improve the model's performance, the authors trained it over several layers (or levels) using CNN .

[5]. The project specifies the use of the Flickr8k dataset for training and testing which is significantly good. The BLEU score is used as an evaluation metric to assess the quality of generated captions.

Existing System

In the existing system, image captioning typically involves a two-step process: feature extraction using Convolutional Neural Networks (CNNs) and caption generation utilizing recurrent neural networks (RNNs).

DISADVANTAGES

- **Limited Understanding of Long-Term Dependencies:** RNNs suffer from difficulties in capturing long-term dependencies in sequential data. In the context of image captioning, where the relationship between words in a sentence is crucial, this limitation may result in the model struggling to maintain context over extended captions.
- **Inability to Capture Global Context:** CNNs are excellent at extracting local features from images, but they might lack the ability to capture global context and relationships between different objects or scenes within an image.

Continue...

This limitation can impact the model's understanding of complex visual scenes, potentially leading to inaccurate or incomplete captions.

- **Fixed-size Image Representations:** CNNs produce fixed-size feature vectors regardless of the input image size. This fixed-size representation may not fully capture the diversity of visual content, leading to information loss for images with varying complexities or compositions.
- **Difficulty in Handling Rare Words:** RNNs may struggle with generating rare or unseen words, as they heavily rely on the training data. Uncommon words or specific vocabulary may not be adequately represented in the training set, leading to challenges in captioning novel or specialized images.

Proposed System

Our proposed system employs combination of ResNet-50 and LSTM ensures a seamless fusion of visual and linguistic information. The ResNet-50 feature vector serves as a foundation for the LSTM to generate contextually relevant captions, effectively marrying the strengths of both modalities. The proposed architecture aims to overcome challenges associated with understanding complex visual scenes and maintaining linguistic context, ultimately leading to improved image captioning performance. The use of ResNet-50 as a feature extractor and LSTM for sequence modeling represents a state-of-the-art approach in the field, aligning with contemporary advancements in deep learning for multimodal tasks. And our proposed system provides audio for the relevant audio for the generated captions utilizing text-to-speech library which will help to visually impaired people.

Continue

Advantages:

1. Rich Visual Representations: ResNet-50: ResNet-50, a powerful convolutional neural network, excels at capturing rich visual features from images. Its deep architecture allows it to learn hierarchical representations, enabling the extraction of intricate details and patterns.

2. High-Level Semantic Features: ResNet-50: ResNet-50 provides high-level semantic features that go beyond simple object detection. This is crucial for generating captions that not only describe objects but also capture the semantic context and relationships within a scene.

3. Effective Handling of Varied Image Sizes: ResNet-50: ResNet-50 can handle images of various sizes without requiring manual resizing. This flexibility is advantageous when working with datasets containing images of different resolutions.

Planning

- **Objective-1:** To develop a deep learning-based image captioning model that surpasses the current state-of-the-art performance in terms of captioning accuracy. We are using flickr30k model to provide accurate captions.
- **Objective-2:** To provide audio for the generated captions. Generated textual captions into spoken audio using a Text-to-Speech (TTS) system. TTS systems take text input and produce corresponding speech output.

Scope: Scope of this project are:

- Accessibility Enhancement
- Multimodal AI Development
- Real-world Applications
- Deep Learning Benchmarking

Planning

Functional Requirements:

- Authentication of user whenever he/she logs into the system.
- System shutdown in case of a cyber-attack.
- A verification email is sent to user whenever he/she register for the first time on some software system.

Non-functional Requirements:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Flexibility

Planning

Software Requirements:

- Operating System : Windows 7+
- Server side Script : HTML, CSS, Bootstrap & JS
- Programming Language : Python
- Libraries : Flask, Pandas, Mysql.connector, Os, Smtplib, Numpy
- IDE/Workbench : PyCharm
- Technology : Python 3.6+
- Server Deployment : Xampp Server
- Database : MySQL

Hardware Requirements:

- Processor - I3/Intel Processor
- Hard Disk - 160GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA
- RAM - 8GB

Planning

1 Long Short Term Memory:

1.1 Sequence Modeling:

In the context of image captioning, LSTMs are used to model sequential information, such as generating a sequence of words in a sentence. The LSTM is employed as the decoder part of the image captioning model, taking as input the features extracted from the image.

1.2 Image Feature Input: The LSTM receives the image features (extracted by a pre-trained CNN like ResNet) as an initial input.

1.3 Word Generation:

The LSTM generates words one at a time, considering the context provided by the image features and the previously generated words. At each time step, the LSTM produces a probability distribution over the vocabulary, and a word is sampled from this distribution.

Planning

1.4 Recurrent Connection:

LSTMs have recurrent connections that allow them to maintain and update an internal memory state, which helps capture long-term dependencies in the sequence. The internal state is updated at each time step based on the input features and the previously generated word.

1.5 Training: During training, the model is optimized to minimize the difference between the predicted caption and the ground truth caption.

Design



Figure1.: Block Diagram

Design

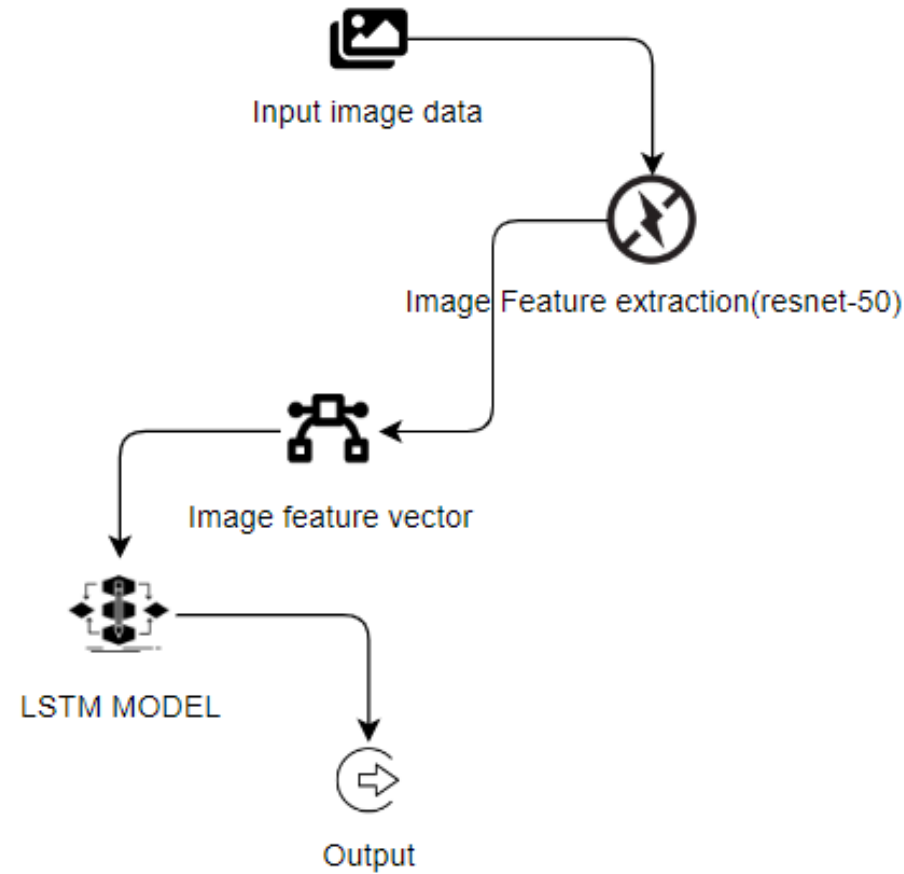


Figure1.: Architecture

Design

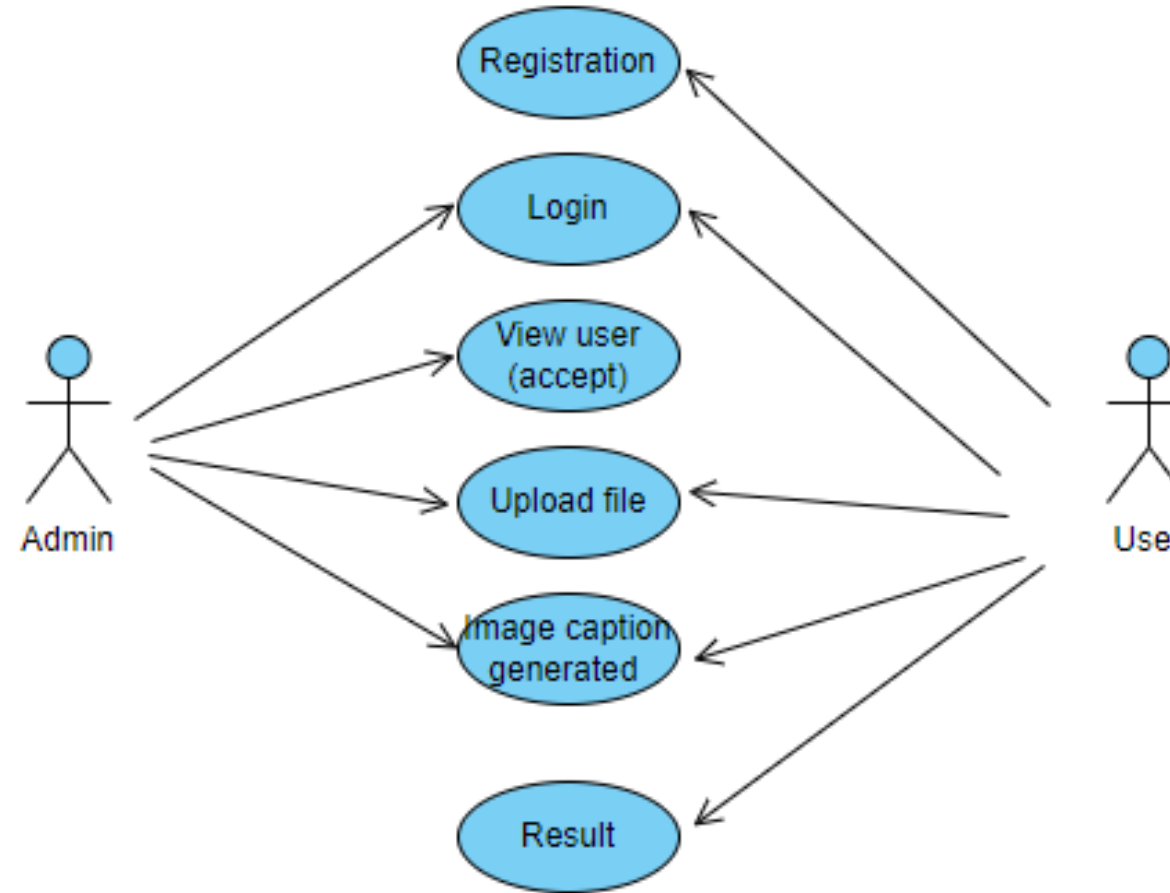


Figure2.: Use Case Diagram

Design

1 System

- **1.1 Create Dataset:** The dataset containing images and text data of the desired objects to be captioned is split into training and testing dataset with the test size of 20-30%.
- **1.2 Pre-Processing:** Data preprocessing is a crucial step in the data analysis pipeline where raw data is transformed, cleaned, and organized to make it suitable for analysis by machine learning algorithms. Resizing and reshaping the images into appropriate format to train our model.
- **1.3 Training:** Data training, often referred to simply as training, is a fundamental concept in machine learning and artificial intelligence. Use the pre-processed training dataset is used to train our model using RESNET-50 and LSTM algorithm.

Design

- **1.4 Pre-Training:** The ResNet model is used as a feature extractor for images. The model is typically pre-trained on a large dataset for image classification tasks (e.g., ImageNet). The weights learned during pre-training capture hierarchical and abstract features in images.
- **1.5 Feature Extraction:** Given an input image, the pre-trained ResNet model is used to extract features from intermediate layers. The features represent high-level visual information present in the image.
- **1.6 Integration with Captioning Model:** The extracted image features are then passed to the decoder part of the image captioning model. The decoder, often implemented as a recurrent neural network (RNN) or transformer, generates a textual description of the image based on the input features.

Design

2 User

- **2.1 Register:** The user needs to register and the data stored database.
- **2.2 Admin login:** Admin logs in into the administrator login and views the user registered list, once he accepts the user data only then the user will be allowed to login.
- **2.3 Login:** A registered user can login using the valid credentials to the website to use a application.
- **2.4 Upload Image:** The user has to upload an image which needs to be Captioned the images.
- **2.5 Prediction:** The results of our model will display the caption of image we have assigned to it.
- **2.6 Logout:** Once the prediction is over, the user can logout of the application.

Implementation

```
import pandas as pd
from flask import Flask, render_template, redirect, url_for, flash
from forms import LoginForm # Assuming your form is in a file named forms.py
from flask import Flask, render_template, request, url_for, redirect, flash, send_from_directory, session
from forms import RegistrationForm, LoginForm
import mysql.connector
import pyttsx3

import torch
from transformers import VisionEncoderDecoderModel, ViTFeatureExtractor, AutoTokenizer
#from app import app
import os
from PIL import Image

import json
from keras.models import load_model
import pickle
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from keras.preprocessing.sequence import pad_sequences
import collections
from keras.preprocessing import image
from keras.applications.resnet50 import ResNet50, preprocess_input, decode_predictions
from keras.models import Model
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField, BooleanField
from wtforms.validators import DataRequired, Length, Email, EqualTo
```

Figure.: Importing libraries

Implementation

```
import matplotlib.pyplot as plt
import re
import cv2
```

[1]

Python

```
# Open the file and read its data
def readFile (path):
    with open(path, encoding="utf8") as file:
        data = file.read()
    return data;
```

[4]

Python

```
# Read captions from the file.token.txt
data = readFile ("data/textFiles/30k_captions.txt")
```

```
# Split the data into each line, to get a list of captions
captions = data.split('\n')
# Remove the last line since it is blank
captions = captions[:-1]
```

[5]

Python

```
print("Total number of caption = " + str(len(captions)))
```

[6]

Python

```
... Total number of caption = 158915
```

```
print(captions[0])
```

Cell 3 of 18

```
# Store the captions in a dictionary
# Each imageID will be mapped to a list of its captions
```

```
content = {}
```

```
for line in captions:
```

```
    imageID, caption = line.split('\t')
```

```
    imageID = imageID.split('.')[0]
```

```
# If the imageID doesn't exist in the dictionary, create a blank entry
```

```
if content.get(imageID) is None:
```

```
    content[imageID] = []
```

```
# Append the current caption to the list of the corresponding image
```

```
content[imageID].append(caption)
```

Figure.: Text data processing

Implementation

```

+ Code + Markdown ...
# Choose a random number, say 50

IMG_PATH = "data/Images/"
image_id = captions[50].split('.')[0]

img = cv2.imread(IMG_PATH + image_id + ".jpg")
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
plt.axis("off")
plt.show()

print("CAPTIONS - ")
for caption in content[image_id]:
    print(caption)

[10]
...
... CAPTIONS -
Five ballet dancers caught mid jump in a dancing studio with sunlight coming through a window .
Ballet dancers in a studio practice jumping with wonderful form .
Five girls are leaping simultaneously in a dance practice room .
Five girls dancing and bending feet in ballet class .
A ballet class of five girls jumping in sequence .

```

```

topn = 50 # Taking top 50 words
def plthist(dfsub, title):
    plt.figure(figsize=(20,3))
    plt.bar(dfsub.index,dfsub["count"])
    plt.yticks(fontsize=15)
    plt.xticks(dfsub.index,dfsub["word"],rotation=90,fontsize=15)
    plt.title(title,fontsize=20)
    plt.show()
plthist(dfword.iloc[:topn,:],title="Top 50 Most frequently appearing words")
plthist(dfword.iloc[-topn,:],title="Least 50 appearing words")

```

Figure.: Mapping captions to the images

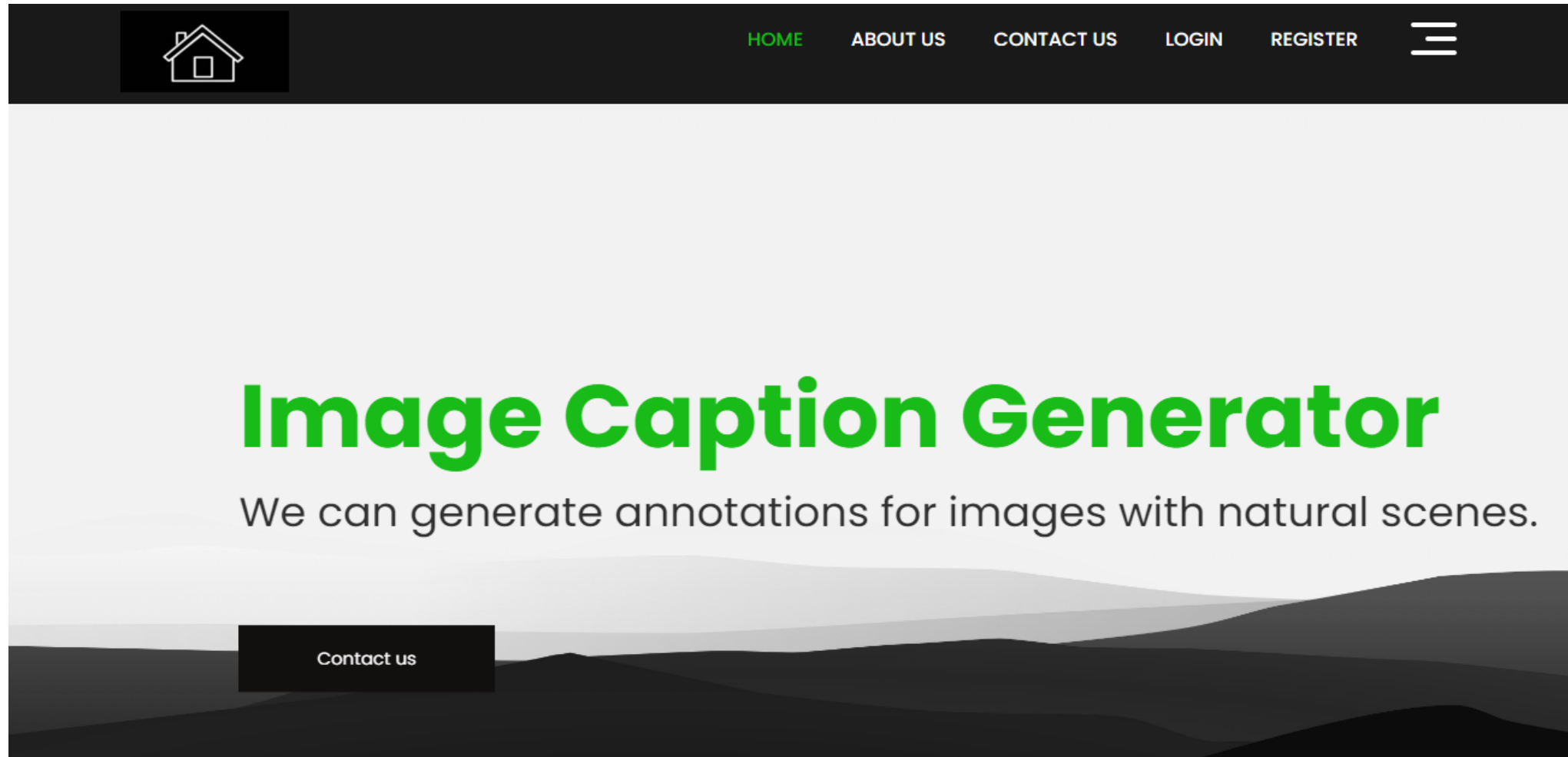
Implementation

```
def text_to_speech(text):  
    engine = pyttsx3.init()  
    engine.say(text)  
    engine.runAndWait()  
  
def Xception(image_path):  
    caption = generate.runModel(image_path)  
    return caption
```

Figure.: Generating audio for the caption

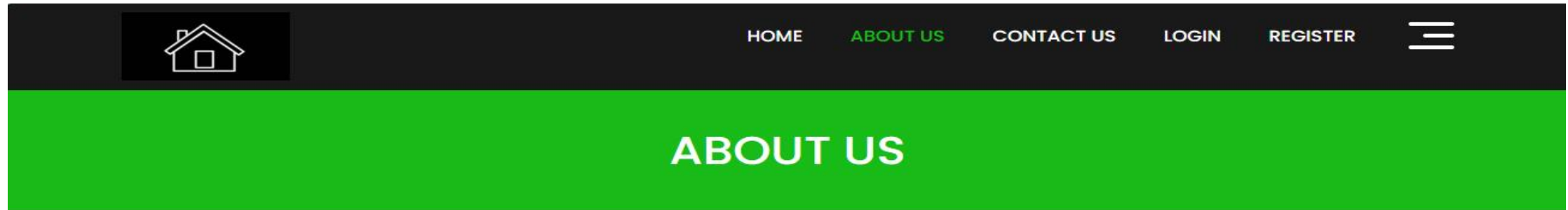
Results

Home Page:



Results

About Us:



Based on the content of an Image, we can generate a caption for any natural image. These captions are widely used for searching of images, tagging in social media etc. Such an application can also help the visually impaired to see the world full with images.


When we see an image, we can quickly recognize what is going on in the image, what objects are present and what they are doing. With the progress in Artificial Intelligence (AI), we are trying to do the same automatically by our computers.

The need for such a system is increasing especially due to the advent of autonomous vehicles / semi-autonomous vehicles which involves reading and understanding millions of images.




Results

Register Page:



HOMEABOUT USCONTACT USLOGINREGISTER



JOIN TODAY

Username

Email

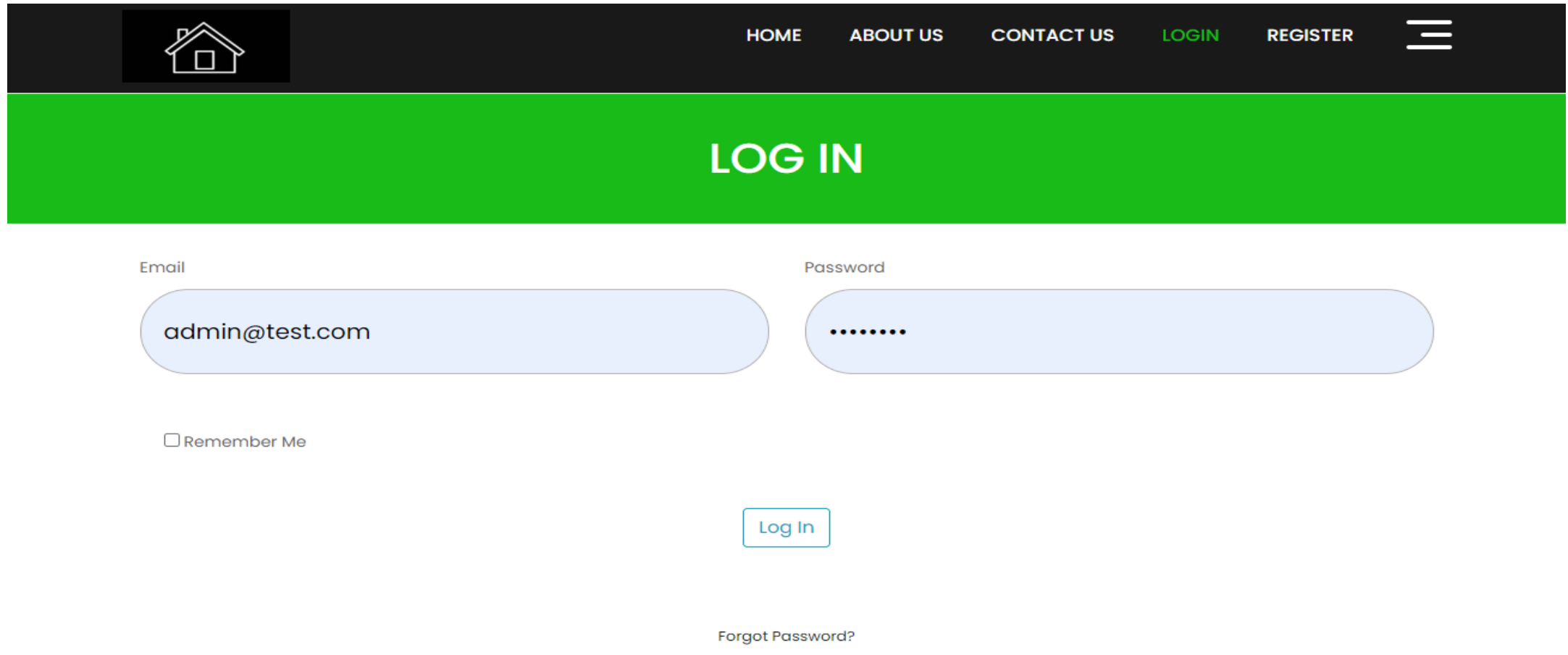
Password

Confirm Password

Sign Up

Results

Login Page:



The screenshot shows a web application's login page. At the top, there is a dark navigation bar with a home icon, links for HOME, ABOUT US, CONTACT US, LOGIN (highlighted in green), and REGISTER, along with a hamburger menu icon. Below this is a large green banner with the text "LOG IN" in white. The main content area contains two input fields: "Email" with the value "admin@test.com" and "Password" with masked characters. Below the email field is a checkbox labeled "Remember Me". A "Log In" button is centered below the inputs. At the bottom, there is a link for "Forgot Password?".

HOME ABOUT US CONTACT US LOGIN REGISTER

LOG IN

Email Password

admin@test.com

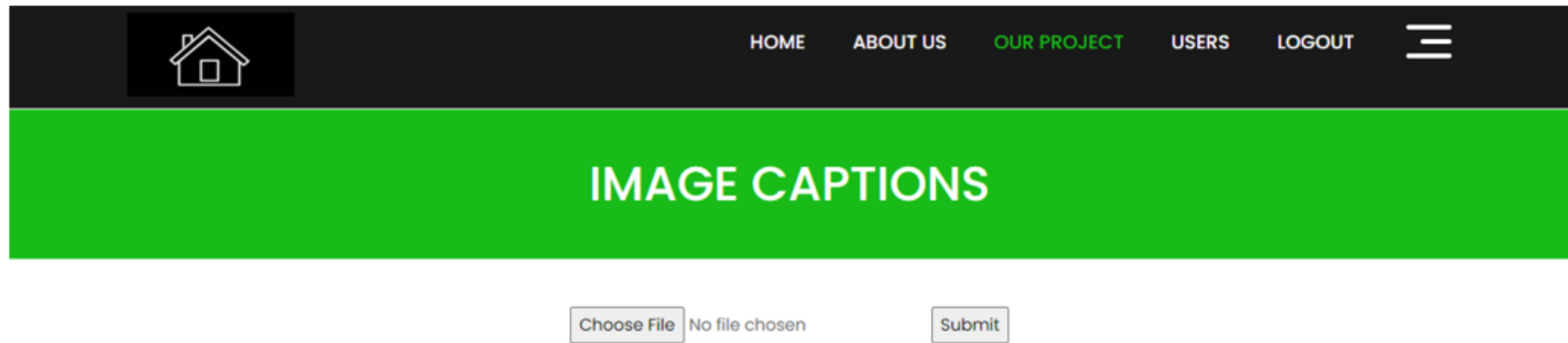
☐ Remember Me

Log In

Forgot Password?

Results

Upload page: Here user uploads the image and caption is generated




HOME ABOUT US OUR PROJECT USERS LOGOUT

IMAGE CAPTIONS


Choose File No file chosen Submit

Results

Result:



HOMEABOUT USOUR PROJECTUSERSLOGOUT





Generate CaptionsUpload a different Image

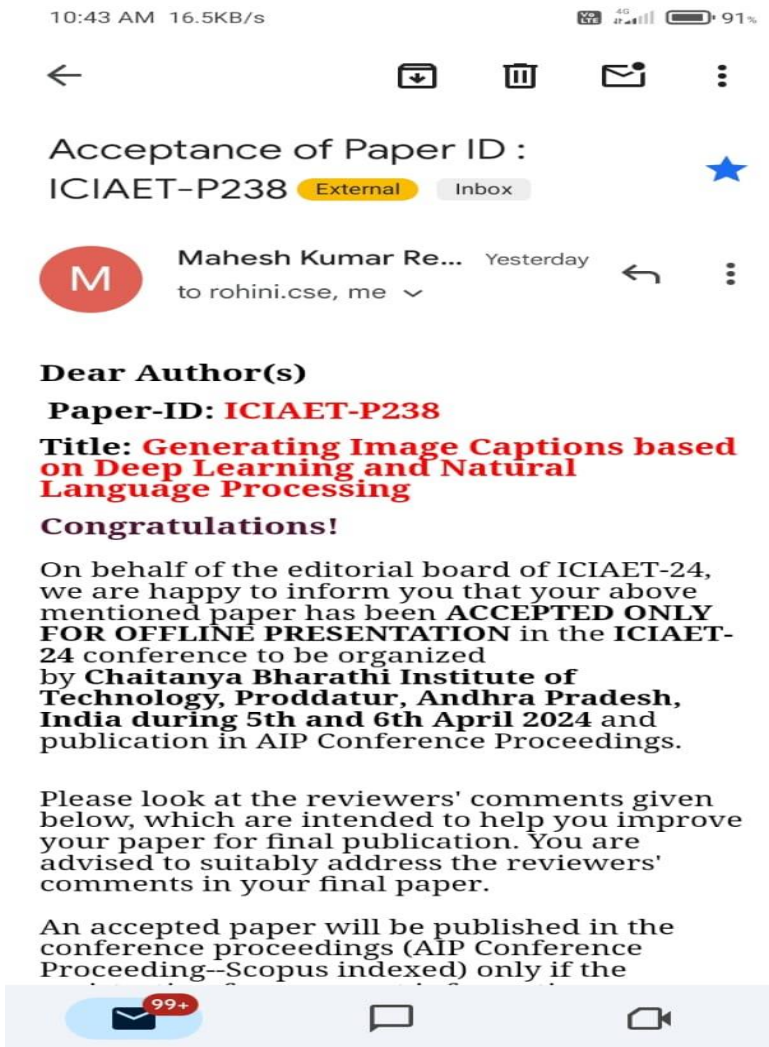
a dog runs through the snow

Tirunelveli(+91)99999xxxxxDemel@gmail.com

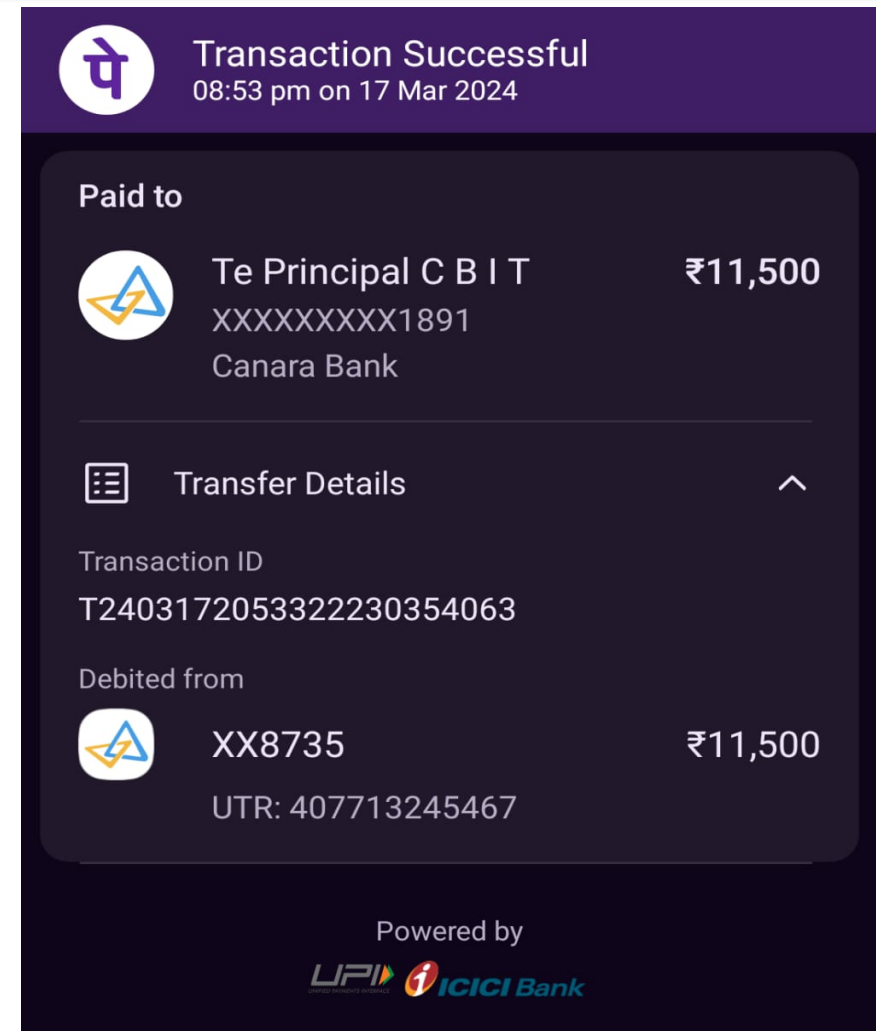
conclusion

The issue of creating meaningful captions for images has been found to be powerfully and effectively solved by the image caption generator that combines Long Short-Term Memory (LSTM) networks with Convolutional Neural Networks (CNNs). Using CNN layers to obtain relevant characteristics and capture spatial information, the CNN-LSTM model showed how to generate contextually relevant captions by efficiently utilizing LSTM layers. Visual perception and sequential data processing work together to address the problems of picture understanding and natural language synthesis through the integration of these two architectures. This work emphasizes the need of merging all the layers to produce better results from challenging tasks like image captioning.

Research Paper



Acceptance from CBIT



Payment Proof

References

- [1]. M Sailaja, K Harika, B Sridhar. Rajan Singh, “[Image Caption Generator using Deep Learning](#)”, 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC).
- [2]. Lakshmi Narasimhan Srinivasan, Dinesh Sreekanthan and A.L Amutha, “[Image captioning - A Deep Learning Approach](#)”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp.
- [3]. Itunuoluwa Isewon, Jelili Oyelade, Olufunke Oladipupo, “[Design and implementation of text to speech conversion for visually impaired people](#)”, IJAIS, Volume 7– No. 2, ISSN : 2249-0868, 2023.

References

- [4]. D. Elliott, F. Keller, “Image Description using Visual Dependency Representations”, [Conference on Empirical Methods in Natural Language Processing](#).
- [5]. Dr. Savita Sangam, Abhijeet Sawant, Abhishek Malap, Deepak Yadav, “[Automated Image Captioning Using Deep Learning](#)”, International Research Journal of Engineering and Technology, vol.07, pp.7310-7312, May 2020.

Git Hub Dashboard

The screenshot shows a web browser window displaying a GitHub repository page. The browser's address bar shows the URL `https://github.com/204g1a0521/CSE-2020-24-Batch-A8`. The repository name is **CSE-2020-24-Batch-A8** and it is marked as **Public**. The repository has 1 branch (main) and 0 tags. The commit history shows 6 commits, with the most recent one being `b996416` 3 minutes ago. The commit message is `204g1a0521 Update Review-0 Project Overview`. The commit details table lists the following files and their commit messages:

File	Commit Message	Time
P1.pdf	I have uploaded Research papers	9 hours ago
P2.pdf	I have uploaded Research papers	9 hours ago
P3.pdf	I have uploaded Research papers	9 hours ago
P4.pdf	I have uploaded Research papers	9 hours ago
README.md	Initial commit	4 days ago
Review-0 Project Overview	Update Review-0 Project Overview	3 minutes ago

The README.md file content is displayed below the commit history, showing the title **CSE-2020-24-Batch-A8**. The right sidebar contains sections for **About** (No description, website, or topics provided), **Releases** (No releases published), **Packages** (No packages published), and **Contributors** (4 contributors).

Any Queries?