IMAGE CAPTION GENERATOR

Name : Aaditry choudhary -20BCE2618
Dheeraj.L -20BCA0030
Prachi Balodia -20BDS0177
Dhanjay -20BCT0320

1 INTRODUCTION

1.1 Oerview

Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe them in a natural language like English. In this Python based project, we will have implemented the caption generator using CNN (Convolutional Neural Networks) and LSTM (Long short term memory). The image features will be extracted from Xception which is a CNN model trained on the imagenet dataset and then we feed the features into the LSTM model which will be responsible for generating the image captions. Convolutional neural networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easy. It can handle the images that have been translated, rotated, scaled and changes in perspective. LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information

1.2 Purpose

The Image Caption Generator project utilizes deep learning techniques to automatically generate descriptive captions for images, bridging the gap between visual content and textual understanding. It has applications in accessibility for the visually impaired, improving image search engines, and enhancing content understanding and summarization.

2 LITERATURE SURVEY

2.1 Existing problem

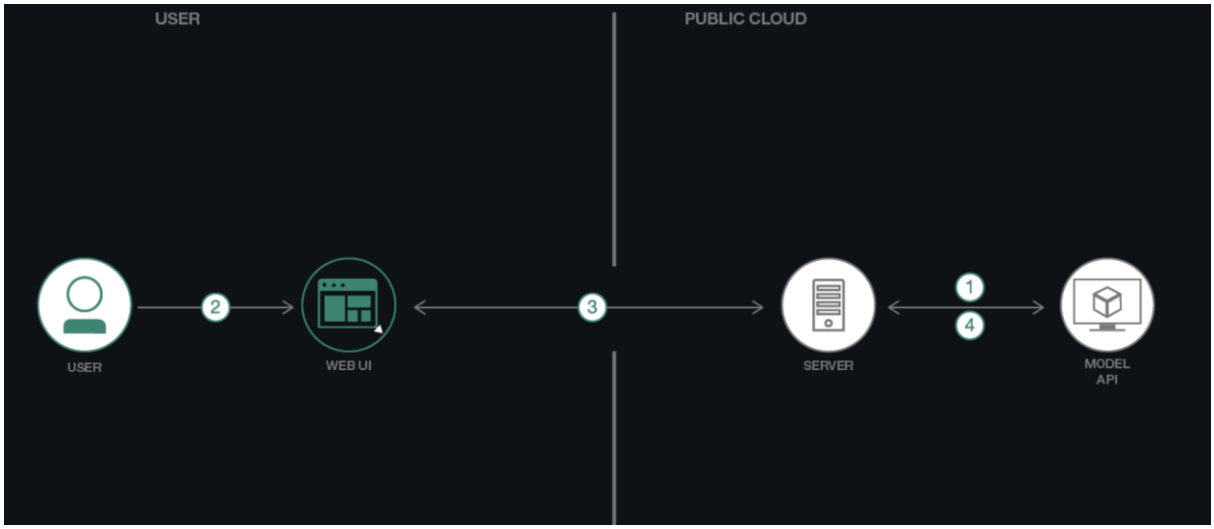
1. SkinVision: SkinVision is a mobile application available for Android and IOS which lets you confirm whether a skin condition can be skin cancer or not.
2. Google Photos: Google Photos is an android application which makes use of an image caption generator to classify photos into Mountains, Sea, etc.
3. Picasa: Picasa is a cross platform image organizer which organizes images and identifies you and your friends in a group picture.
4. Tesla/Google Self Drive Cars: All Self driving cars are using image caption generators so that computers can drive cars safely and efficiently.
5. Adobe Photoshop: It is an image editing application which uses image captioning for providing editing recommendations.
6. Facebook: Facebook is an online social media and social networking service which automatically generates captions of images you have uploaded.
7. ShutterStock: ShutterStock is a stock photography application where you can sell images and it automatically generates tags for images using an image caption generator.

2.2 Proposed solution

This project is proposed to be used in robotics, using image caption generator robotics can determine the surrounding conditions and show intelligence like humans, using image caption generator robots can make better decisions by understanding the environmental conditions.

3 THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

The Image Caption Generator is primarily built using deep learning frameworks such as TensorFlow,

PyTorch, or Keras. These frameworks provide the necessary tools and APIs for constructing and training complex neural network models.

The image data is preprocessed using libraries like OpenCV or PIL, which handle tasks such as resizing, normalization, and feature extraction. These libraries ensure the input images are properly prepared for analysis.

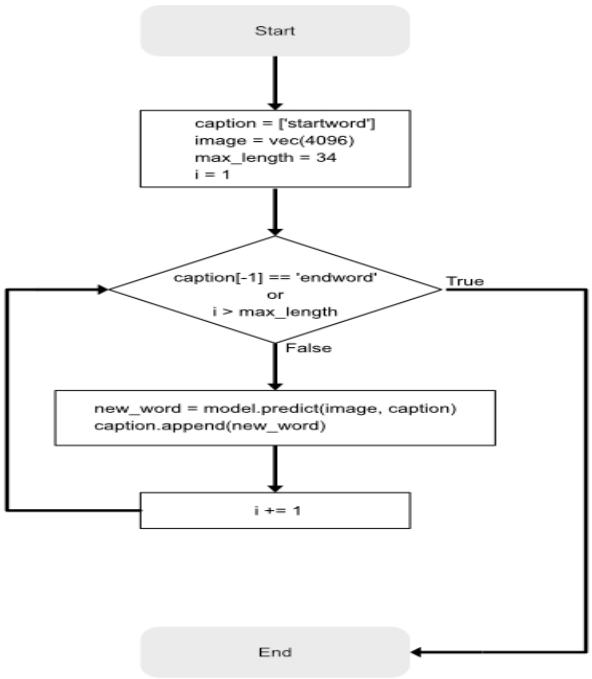
Neural network architectures, such as Convolutional Neural Networks (CNNs) for image analysis and Recurrent Neural Networks (RNNs) or Transformer models for language modeling, are implemented using the chosen deep learning framework. These architectures are responsible for extracting image features and generating descriptive captions.

Training data, often obtained from datasets like MSCOCO or Flickr, is used to train the Image Caption Generator. These datasets consist of a large collection of images with corresponding captions, allowing the model to learn the relationship between visual content and textual descriptions.

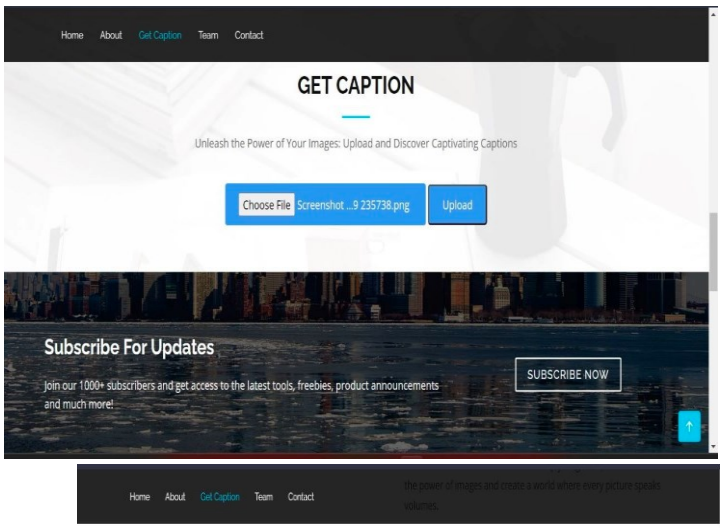
4 EXPERIMENTAL INVESTIGATIONS

Experimental investigations are conducted to gather empirical evidence and test hypotheses. They involThey follow a structured methodology, including hypothesis formulation, data collection, analysis, and interpretation. These experiments aim to provide objective and reliable evidence to support or refute scientific theories and understand cause-and-effect relationships in various fields of study.ve systematically manipulating variables, measuring outcomes, and analyzing data to draw conclusions.

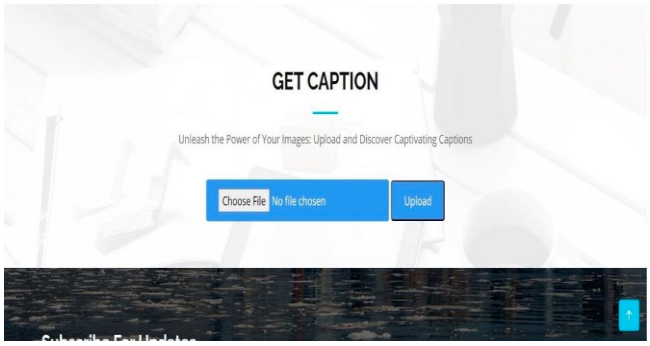
5 FLOWCHART

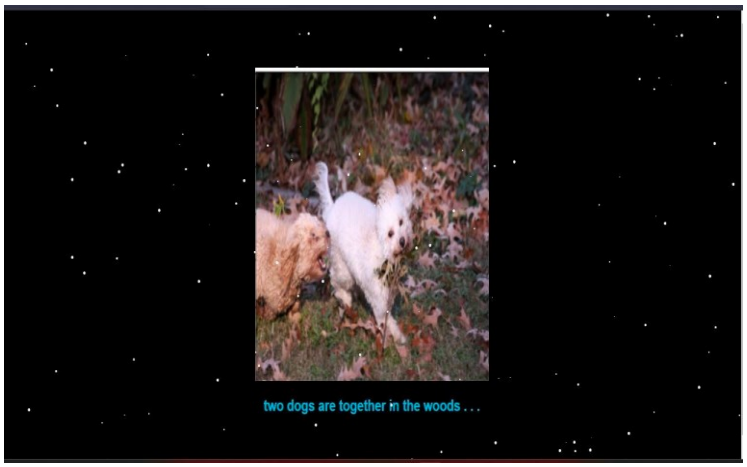


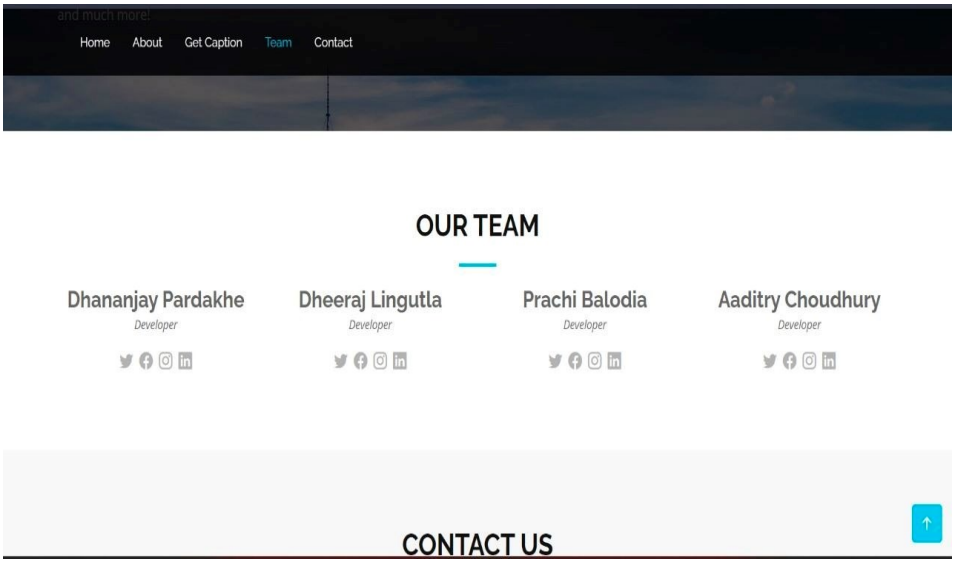
6 RESULT


Home
About
Get Caption
Team
Contact
GET CAPTION
Unleash the Power of Your Images: Upload and Discover Captivating Captions
Choose File
Screenshot ...9 235738.png
Upload
Subscribe For Updates
Join our 1000+ subscribers and get access to the latest tools, freebies, product announcements and much more!
SUBSCRIBE NOW

Home
About
Get Caption
Team
Contact


GET CAPTION
Unleash the Power of Your Images: Upload and Discover Captivating Captions
Choose File
No file chosen
Upload


two dogs are together in the woods . . .



7 ADVANTAGES & DISADVANTAGES

ADVANTAGES

1. Assistance for visually impared
   a. The advent of machine learning solutions like image captioning is a boon for visually impaired people who are unable to comprehend visuals.
   b. With AI-powered image caption generators, image descriptions can be read out to the visually impaired, them to get a better sense of their surroundings.
2. Recommendations in editing
   a. The image captioning model automates and accelerates the closed captioning process for digital content production, editing, delivery, and archival.
   b. Well-trained models replace manual efforts for generating quality captions for images as well as videos.
3. Media and Publishing Houses
   a. The media and public relations industry circulate tens of thousands of visual data across borders in the form of newsletters, emails, etc.
   b. The image captioning model accelerates subtitle creation and enables executives to focus on more important tasks.
4. Self driving cars
   a. By using the captions generated by image caption generator self driving cars become aware of the surroundings and make decisions to control the car.
5. Reduce vehicle accidents
   a By installing an image caption generator in the vehicles, vehicles can stop by applying the automatic brake when an object in the surrounding is detected.

DISADVANTAGES

1) Limited Contextual Understanding: Image caption generators may not capture the full context of the surroundings, leading to potential misunderstandings or misinterpretations of the visual scene. This can result in incorrect or unsafe decisions made by the self-driving car.

2) Ambiguity and Inaccurate Captions: Image caption generators can sometimes produce ambiguous or inaccurate captions, which can lead to confusion or incorrect interpretations by the self-driving car's decision-making algorithms. This can pose risks and potentially compromise the safety of the vehicle and its passengers.

3) Lack of Real-Time Adaptability: Image caption generators typically operate in a batch processing manner, generating captions based on static images. However, real-world driving scenarios require dynamic and real-time adaptability to changing road conditions, traffic situations, and unexpected events. The fixed captions may not adequately address these dynamic aspects.

4) Dependence on Image Quality and Recognition: The accuracy of the generated captions heavily relies on the quality

8 APPLICATIONS

1) Accessibility for the Visually Impaired: By providing textual descriptions of images, the solution can enhance accessibility for visually impaired individuals, enabling them to understand and engage with visual content.

2)Image Search and Organization: The generated captions can improve image search engines by enabling more a accurate and relevant search results based on image content. They can also assist in organizing and categorizing large image databases.

3)Content Generation for Social Media: Social media platforms can leverage image captions to automatically generate descriptive and engaging content for image-based posts, enhancing user experience and facilitating content creation.

4)E-commerce Product Descriptions: The solution can aid in generating detailed and informative captions for products in e-commerce platforms, improving product descriptions and helping users make informed purchasing decisions.

5)Automatic Video Captioning: The image caption generator can be extended to generate captions for videos, enabling automatic video captioning for improved accessibility and content understanding.

6)Content Moderation and Filtering: The generated captions can be used as additional information for content moderation, aiding in identifying inappropriate or harmful images and videos.

7)Medical Imaging Analysis: In the medical field, the solution can assist in analyzing medical images by generating captions that describe the visual features and abnormalities present, supporting doctors and medical professionals in diagnosis and treatment planning.

9 CONCLUSION

Image caption generator is a task that involves computer vision and natural language processing concepts to recognize the context of an image and describe them in a natural language like English. So, to make an image caption generator model, have merged these architectures. It is also called a CNN-RNN model.

- CNN is used for extracting features from the image. We will use the pre-trained model Xception.
- LSTM will use the information from CNN to help generate a description of the image.

Convolutional Neural networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images.CNN is basically used for image classifications and identifying if an image is a bird, a plane or Superman, etc.It scans images from left to right and top to bottom to pull out important features from the image and combines the feature to classify images. It can handle the images that been translated, rotated, scaled and changes in perspective.

LSTM stands for Long short term memory, they are a type of RNN (recurrent neural network) which is well suited for sequence prediction problems. Based on the previous text, we can predict what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information.

10 FUTURE SCOPE

1. Sheer complexity
   a. Everything has the potential to be scaled up in terms of power and complexity. With technological advancements,we can make CPUs and GPUs cheaper and/or faster, enabling the production of bigger, more efficient algorithms.We can also design neural nets capable of processing more data, or processing data faster, so it may learn to recognize patterns with just 1,000 examples, instead of 10,000. Unfortunately, there may be an upper limit to how advanced we can get in these areas—but we haven't reached that limit yet, so we'll likely strive for it in the near future.

2. New applications
   a. Rather than advancing vertically, in terms of faster processing power and more sheer complexity, neural nets could (and likely will) also expand horizontally, being applied to more diverse applications. Hundreds of industries could feasibly use neural nets to operate more efficiently, target new audiences, develop new products,or improve consumer safety—yet it's criminally underutilized. Wider acceptance, wider availability, and more creativity from engineers and marketers have the potential to apply neural nets to more applications.

3. Integration
   a. The weaknesses of neural nets could easily be compensated if we could integrate them with a complementary technology, like symbolic functions. The Hard part would be finding a way to have these systems work together to produce a common result—and engineers are already working on it.

11 BIBILOGRAPHY

1. HaoranWang , Yue Zhang, and Xiaosheng Yu, “An Overview of Image Caption Generation Methods”, (CIN-2020)
2. B.Krishnakumar, K.Kousalya, S.Gokul, R.Karthikeyan, andD.Kaviyarasu, “IMAGE CAPTION GENERATOR USIN
   DEEP LEARNING”, (international Journal of Advanced Science and Technology- 2020)
3. MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga, “A Comprehensive Survey of Deep Learning for Image Captioning” ,(ACM-2019)
4. Rehab Alahmadi, Chung Hyuk Park, and James Hahn, “Sequence-tosequence image caption generator”, (ICMV-2018)
5. Oriol Vinyals, Alexander Toshev, SamyBengio, and Dumitru Erhan, “Show and Tell: A Neural Image Caption Generator”,(CVPR 1, 2-2015)
6. Priyanka Kalena, Nishi Malde, Aromal Nair, Saurabh Parkar, and Grishma Sharma, “Visual Image Caption Generator Using Deep Learning”, (ICAST-2019)
7. Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra, andNand Kumar Bansode,“Camera2Caption: A Real-Time Image Caption Generator”, International Conference on Computational Intelligence in Data Science(ICCIDS) - 2017
8. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, et al., "Show, attend and tell: Neural image caption generation with visual attention", Proceedings of the International Conference on Machine Learning (ICML), 2015.
9. J. Redmon, S. Divvala, Girshick and A. Farhadi, "You only look once: Unified real-time object detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
10. D. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate.arXiv:1409.0473”, 2014.
11. Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi,

APPENDIX

A. Source Code

```
import cv2
from tensorflow.keras.models import load_model
import numpy as np
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Dense, Flatten, Input, Convolution2D,
        Dropout, LSTM,TimeDistributed, Embedding, Bidirectional,
        Activation, RepeatVector,Concatenate
from tensorflow.keras.models import Sequential, Model
```

```
from tensorflow.keras.preprocessing import image, sequence
import cv2
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tqdm import tqdm

from tensorflow.keras.applications import ResNet50

resnet = ResNet50(include_top=False,weights='imagenet',
                  input_shape=(224,224,3),pooling='avg')
print("="*50)
print("resnet loaded")
vocab = np.load('vocab.npy', allow_pickle=True)
vocab = vocab.item()
inv_vocab = {v:k for k,v in vocab.items()}

embedding_size = 128
max_len = 40
vocab_size = len(vocab)

image_model = Sequential()

image_model.add(Dense(embedding_size, input_shape=(2048,),
              activation='relu'))
image_model.add(RepeatVector(max_len))



language_model = Sequential()

language_model.add(Embedding(input_dim=vocab_size,
              output_dim=embedding_size, input_length=max_len))
language_model.add(LSTM(256, return_sequences=True))
language_model.add(TimeDistributed(Dense(embedding_size)))
```

```
conca = Concatenate()([image_model.output, language_model.output])
x = LSTM(128, return_sequences=True)(conca)
x = LSTM(512, return_sequences=False)(x)
x = Dense(vocab_size)(x)
out = Activation('softmax')(x)
model = Model(inputs=[image_model.input, language_model.input], outputs
            = out)

# model.load_weights("../input/model_weights.h5")
model.compile(loss='categorical_crossentropy', optimizer='RMSprop',
            metrics=['accuracy'])
model.load_weights('mine_model_weights.h5')
print("="*50)
print("model loaded")

app = Flask(_name_)
app.config['SEND_FILE_MAX_AGE_DEFAULT'] = 1

@app.route('/')
def index():
    return render_template('index.html')



@app.route('/after', methods=['GET', 'POST'])
def after():
    global model, vocab, inv_vocab

    file = request.files['file1']
    file.save('static/file.jpg')
    img = cv2.imread('static/file.jpg')
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (224,224,))
```

```
img = np.reshape(img,(1,224,224,3))
features = resnet.predict(img).reshape(1,2048)
text_in = ['startofseq']
final = ""
print("="*50)
print("GETING Captions")

count = 0
while tqdm(count < 20):

    count += 1

    encoded = []
    for i in text_in:
        encoded.append(vocab[i])

    padded = pad_sequences([encoded], maxlen=max_len,
          padding='post', truncating='post').reshape(1,max_len)

    sampled_index = np.argmax(model.predict([features, padded]))

    sampled_word = inv_vocab[sampled_index]

    if sampled_word != 'endofseq':
        final = final + ' ' + sampled_word

    text_in.append(sampled_word)



return render_template('predict.html',final = final)
```

```
if _name_ == "_main_":
    app.run(debug=True)
```

Thank you