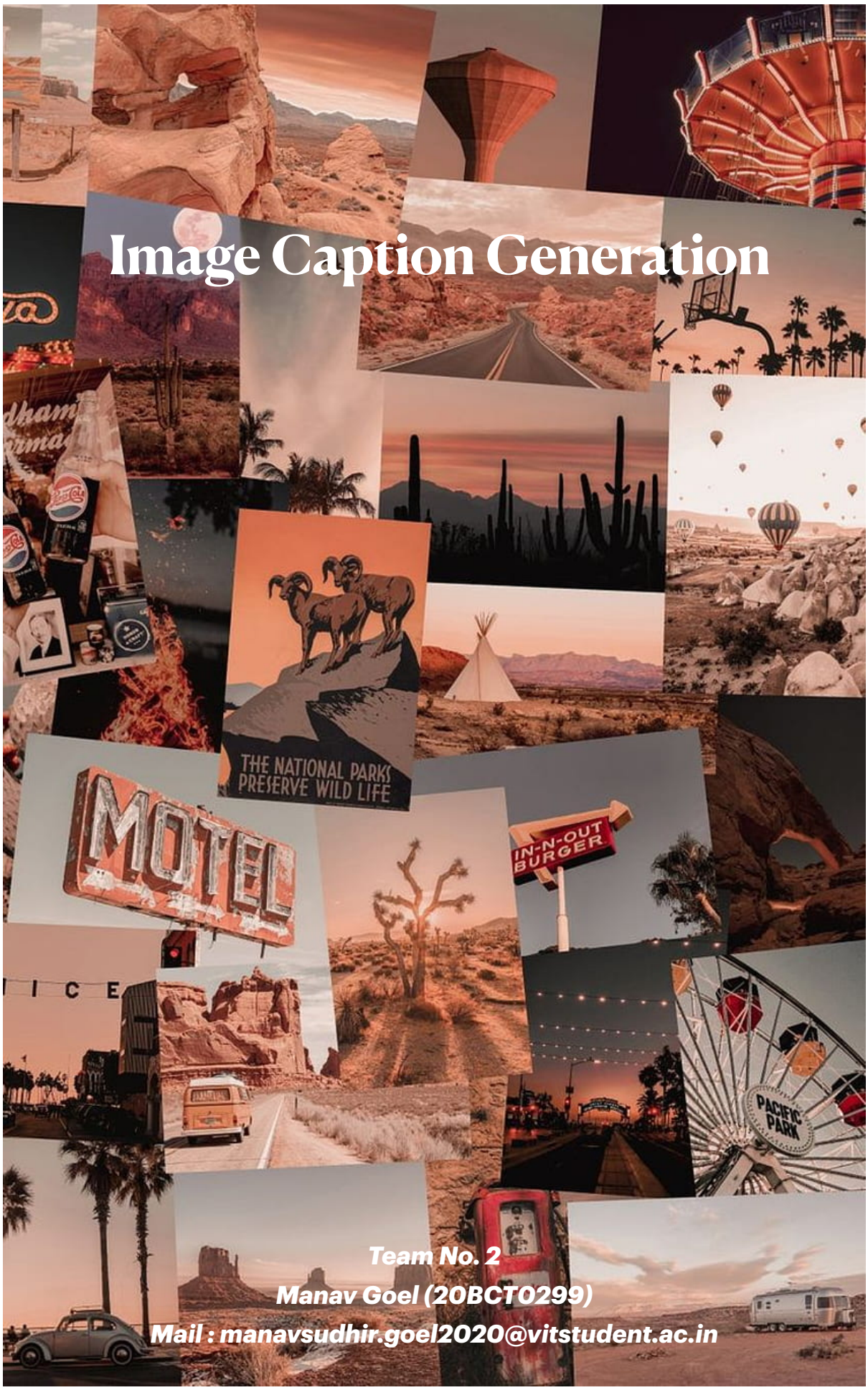


Image Caption Generation

Team No. 2

Manav Goel (20BCT0299)

Mail : manavsudhir.goel2020@vitstudent.ac.in

Introduction

1.1 Overview

"Image Caption Generator" utilises the Xception model as a feature extractor to extract meaningful features from images. These extracted features, along with processed image captions, are then fed into a model composed of Dense and LSTM layers. The objective is to train this model to understand the relationship between the image features and their corresponding captions. The extracted image features and processed image captions are combined and passed through a series of Dense and LSTM layers. The Dense layers help in transforming and reducing the dimensionality of the combined features, while the LSTM layers enable the model to capture sequential information and dependencies within the captions.

1.2 Purpose

Image caption generation has several practical applications in various domains. Image caption generators can assist visually impaired individuals in understanding the content of images on social media, news articles, and other platforms. The caption generated could be verbalised for them. It can automatically generate concise and informative summaries of images. This can be particularly useful in scenarios where large amounts of visual content need to be processed and summarised, such as in news articles, social media feeds, or image collections. Captioning images enables efficient indexing and retrieval of visual data. By associating descriptive captions with images, it becomes easier to search and retrieve specific images based on their content, making image libraries and databases more organised and searchable.

Literature Survey

2.1 Existing problem

1. Show and Tell: A Neural Image Caption Generator" by Vinyals et al. (2015): This paper introduced an early approach to image captioning using a deep learning model. The model consisted of a the Inception V3 CNN model for image feature extraction and a recurrent neural network (RNN) for generating captions. It is designed to efficiently capture visual features from images and has been widely used in various computer vision tasks, including image captioning. The Inception V3 model provides a powerful feature extraction backbone for the image caption generator in the "Show and Tell" approach.

2. Image Caption Generator Based on Deep Neural Networks" by Jianhui Chen, Wenqiang Dong, and Minchen Li. The project focuses on leveraging the power of deep learning, specifically deep neural networks, to automatically generate descriptive captions for images. They have carried out a comparative study for different CNN model such as VGGNet and GoogLeNet, with RNNs LSTM or GRU. Experiment results show that: first the VGGNet outperforms the AlexNet and GoogLeNet in BLEU score measurement; second, the simplified GRU model achieves comparable results with more complicated LSTM model; third, increasing the beam size increase the BLEU score in general but does not necessarily increase the quality of the description which is judged by humans

3. "Enhanced Image Captioning with Colour Recognition Using Deep Learning Methods" published in the journal of Applied Sciences (2021) : The study focuses on enhancing the accuracy and descriptive quality of image captions by considering colour information. The authors propose a methodology that combines object detection, colour analysis, and deep learning-based image captioning. The incorporation of colour recognition aims to enhance the image captioning process by adding an additional dimension of information. By considering colour attributes, the system aims to provide more accurate and descriptive captions that capture both the visual content and colour characteristics of the image.

4. "Image Caption Generator Using Attention Mechanism" by V. Agrawal, S. Dhekane, N. Tuniya, and V. Vyas (2021) : The main focus of the paper is to address the challenge of generating descriptive captions for images by incorporating an attention mechanism. The attention mechanism allows the model to focus on relevant image regions while generating captions, improving the alignment between visual content and textual descriptions. However, attention mechanisms are commonly used in conjunction with recurrent neural networks (RNNs), such as long short-term memory (LSTM), to generate captions by attending to specific image regions at each time step of caption generation.

2.2 Proposed Solution

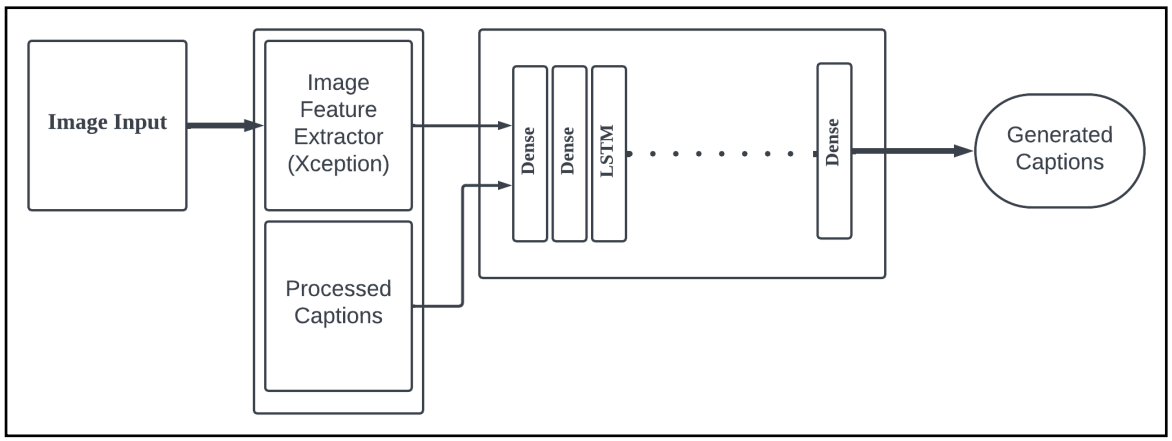
My proposed model for image caption generation consists of the basic structure of the above papers. Where a CNN model called Xception is used to extract features. Xception model is considered to be a good and effective deep learning model for image classification and feature extraction tasks. It has achieved state-of-the-art performance on various benchmark datasets and is widely used in computer vision applications. The Xception model has been shown to outperform previous CNN architectures on benchmark datasets like ImageNet, achieving higher accuracy with fewer parameters. Its ability to capture fine-grained details and learn discriminative features has made it a popular choice in computer vision research and applications. Thus, I chose this.

The features that are extracted are then coupled with the caption that have been processed as follows, change casing, remove punctuations, split, add start and end sequences, and finally vectorise it. Now, the coupled data is fed into the custom RNN model, which first processes the image features once and then along with vectorised caption undergoes several hidden layers like LSTM, Add, Dense and Dropout, finally outputting the caption words.

The input is fed as follows, first a `*startseq*` is fed along side features extracted from the test image. It then gives a vector, from which the word index is determined, and that word is added to sequence, and this goes on in until `*endseq*` or no discernible output is received.

Theoretical Analysis

3.1 Block Diagram



3.2 Hardware / Software designing

Hardware Requirements :

- Needs to have a dedicated GPU

Software Requirements :

- Python 3 is required

Libraries in python that were used during the project :

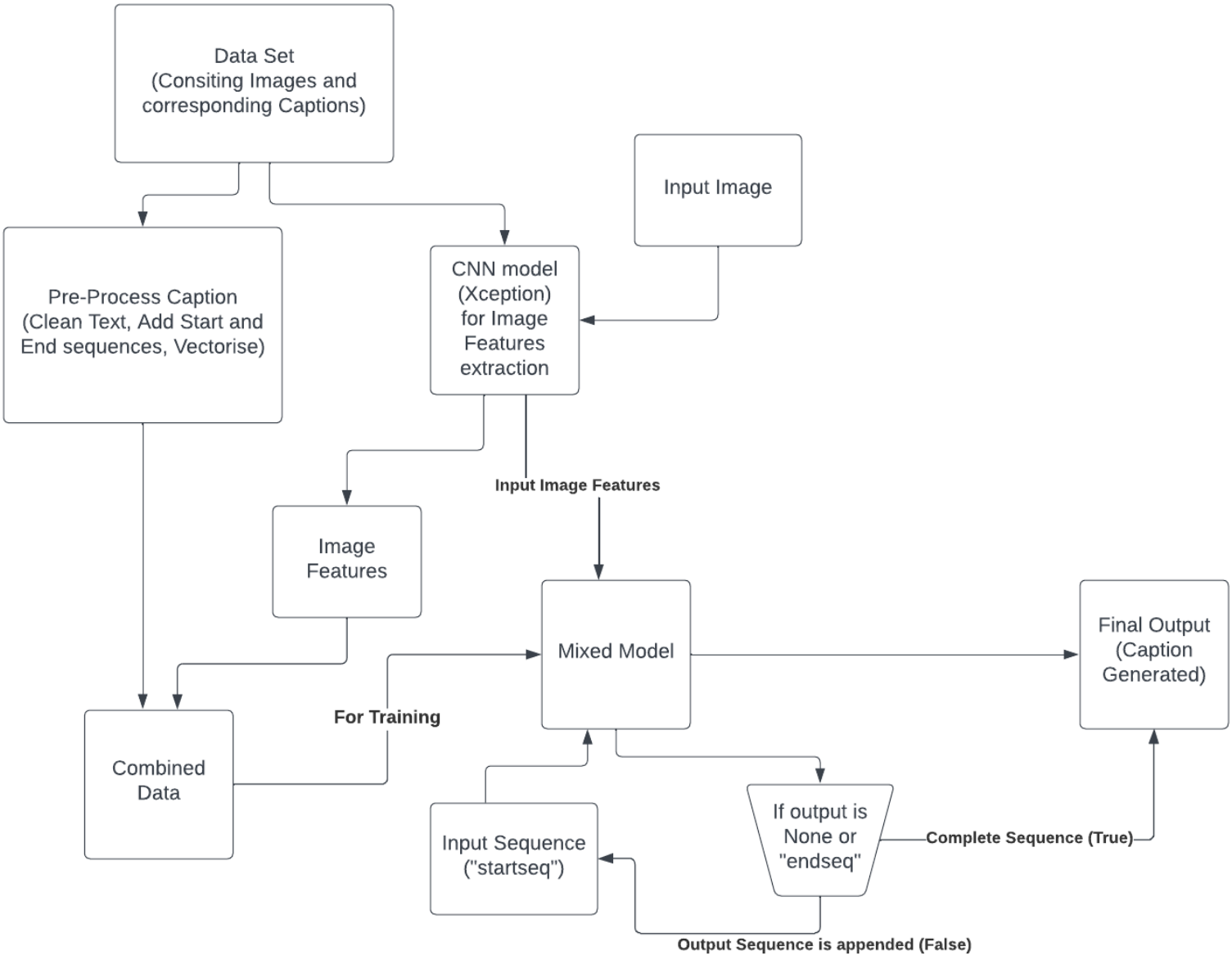
- Tensorflow
- Pickle
- Keras
- Numpy
- Matplotlib
- Pandas
- Flask
- Nltk

Experimental Investigations

Some of the observations I made during the development phase of the project are as follows :

- It is better to use transfer learning for image feature extraction, and use highly accurate and efficient models instead of building your own, due to the training time that it takes to make your own and the complexity required.
- Another observation I made was that my model started overfitting over 30 epochs for 64 steps an epoch, while it was in accurate for anything under 25 epochs.
- I have used the BLEU -1 and -2 to score the captions generated against the test captions, for BLEU - 1, the accuracy was approximately about 0.65 for 450 test cases. And 0.45 for BLEU -2. In general, the rule of thumb is that the BLEU score should be around 0.5 to 0.9 for it to be considered a good model. As 0.65 lies in that range, my model can be considered pretty good. Comparing this to the model in “Image Caption Generator Based on Deep Neural Networks” paper, in which the best model also had about this range but slightly less. Hence, my model is better.
- Another insight I got while researching for the project and coming up with a solution, is that all the generators are powered by a combination architecture of CNN + RNN. Now, CNN is for the computer vision aspect. But, why RNN, is what I question to myself, and on why make RNNs when they are only used for sequential data. So, I went ahead with a custom model which has mixed layers like Dense and an LSTM layer (to capture sequential dependencies in caption words).

Flowchart


Data Set
(Consiting Images and
corresponding Captions)
Input Image
Pre-Process Caption
(Clean Text, Add Start and
End sequences, Vectorise)
CNN model
(Xception)
for Image
Features
extraction
Input Image Features
Image
Features
Mixed Model
Final Output
(Caption
Generated)
For Training
Combined
Data
Input Sequence
("startseq")
If output is
None or
"endseq"
Complete Sequence (True)
Output Sequence is appended (False)

Result

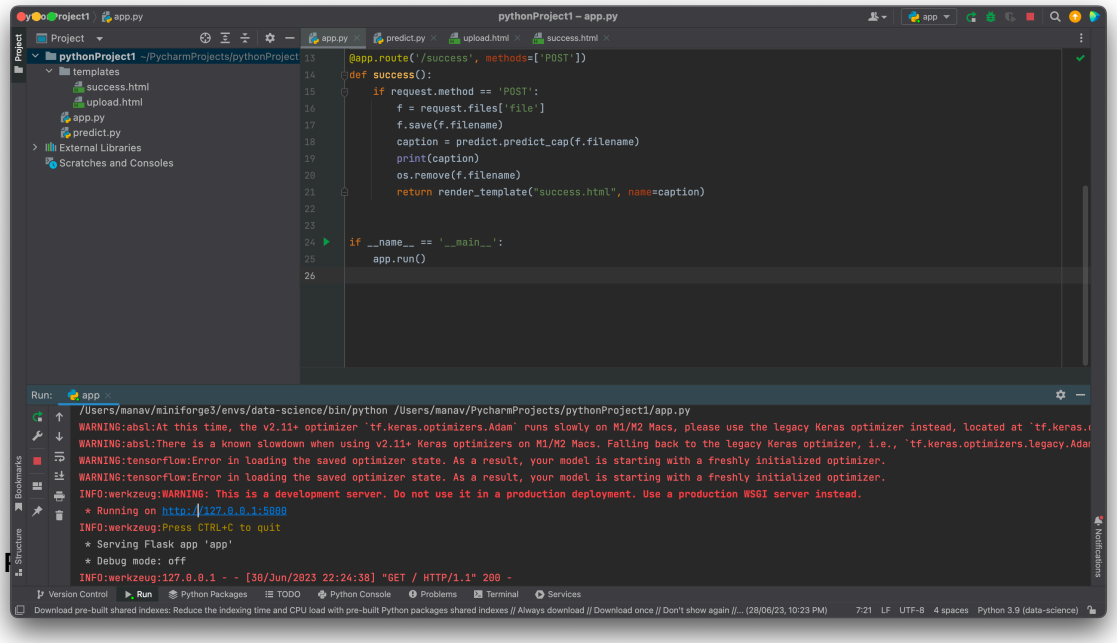
Here are few of the testing phase results, as it can be seen though the captions are not completely correct, they do give some gist of the picture.

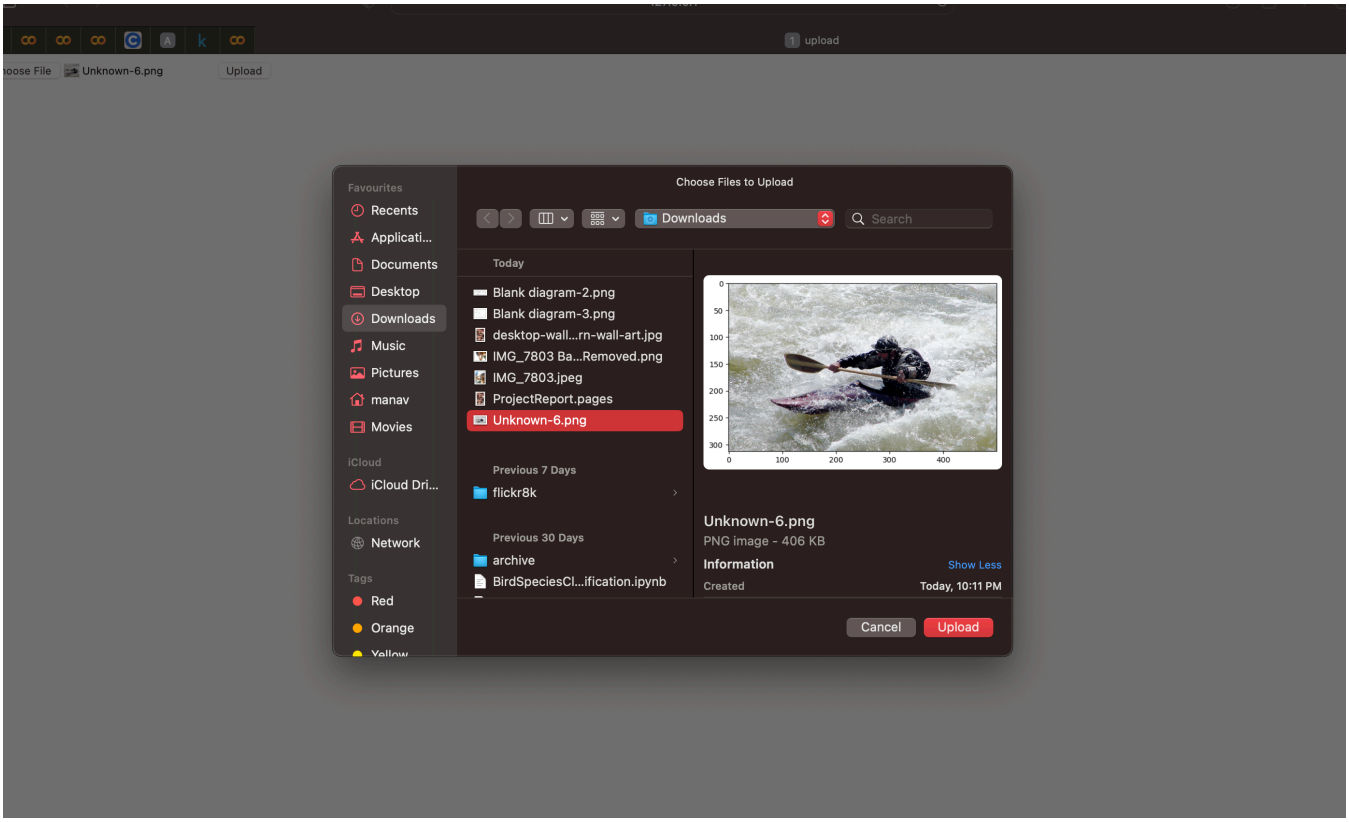
```
Actual :
startseq football player in red jersey getting his knee looked at by another man endseq
startseq man is checking out an injured football player at game endseq
startseq an injured football player is being nursed on the field in the middle of game endseq
startseq two men examine football players leg endseq
startseq two men help an injured player on the field endseq

Predicted :
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 51ms/step
startseq man is sitting in the the middle of the the the the the catcher endseq
```

```
Actual :
startseq group of backpackers lay on the dry ground endseq
startseq group of hikers are resting on the ground in front of some mountains endseq
startseq people camp with the mountains in the background endseq
startseq several hikers rest with their gear in front of mountain endseq
startseq the group of hikers is resting in front of mountain endseq

Predicted :
1/1 [==============================] - 0s 45ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 42ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 46ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 39ms/step
startseq group of people are climb the hill endseq
```

```
Actual :
startseq kayaker is battling waves in purple boat while wearing rain gear endseq
startseq man is paddling in whitewater rapids endseq
startseq man kayaks in rough water endseq
startseq man in kayak swimming in rough waters endseq
startseq person kayaking in white water endseq

Predicted :
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 28ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 31ms/step
startseq man is riding the wave in the water endseq
```

Here are some of the screen shots from the Flask integrated version, displaying the same picture





File uploaded successfully

Generated Caption: surfer is surfing in the wave

Advantages and Disadvantages

7.1 Advantages

1. I have used Xception model, which is not only has good accuracy, it is also pretty efficient. It also outperforms some other well known CNN models that are used for the same project.
2. The model combines image features extracted by a CNN and textual input to generate captions. This integration allows the model to leverage both visual and semantic information, potentially leading to more informative and accurate captions.
3. The LSTM layer in the model enables sequential processing of the caption input, allowing the model to capture temporal dependencies and generate captions that consider the context of previous words.
4. The inclusion of dropout layers (dropout_3 and dropout_4) helps to prevent overfitting by randomly disabling neurons during training. This regularisation technique can improve the model's generalisation ability and reduce the risk of overfitting to the training data.

7.2 Disadvantages

1. This model though needs a lot of fine tuning, and had a very time consuming processing of testing each model variation based on the amount of epochs that it got trained.
2. From my perspective, it seems that another thing is that it falls prey to bias, if the photos of someone wearing red T-shirt is overwhelming high then it will categorise other similar photos as such too. Making it less detailed.

Applications

Even the its current accuracy my model can be helpful in various domains such as :

- In the e-commerce industry, it can automatically generate captions for product images. This can improve search engine optimization (SEO), facilitate product discovery, and enhance the overall shopping experience for users. Provided it is trained on specific dataset
- It can aid in indexing and retrieval tasks by automatically generating captions for images in large image databases. This allows for efficient and accurate searching and filtering based on image content.
- Content management systems used by websites or blogs can integrate this to automatically generate captions for images uploaded by content creators. This can improve the overall quality and consistency of image captions across the platform.
- It can also be used for categorisation of visual data, as it generates caption that can then be used for making categories.

Conclusion

In conclusion, The goal was to automatically generate descriptive captions for images, leveraging the power of neural networks and sequential processing.While the project yielded promising results, it is important to acknowledge certain limitations. The model's efficiency and performance were influenced by factors such as the quality and diversity of the training data, the choice of hyperparameters, and the training process itself. Further optimization and fine-tuning of the model could have potentially improved its accuracy and generalisation ability. Despite these limitations, the developed

image caption generator has practical applications in various domains. Overall, the project represents a step towards automating the process of generating descriptive captions for images, and it provides a foundation for further research

Future Scope

To further enhance the project, future work could focus on incorporating attention mechanisms to improve the model's ability to focus on relevant image regions, exploring alternative architectures or pre-trained models, and conducting extensive evaluation and user studies to assess the quality and effectiveness of the generated captions. Better datasets, with diverse images and less bias is also necessary.

Bibliography

Some of the material resources that I referred to are :

- Kaggle.com
- AnalyticVidhya.com
- deeplearning.ai
- medium.com
- ieeexplore.ieee.org
- python.org

Appendix

The link to my code which is in GitHub is : https://github.com/manavgoel472003/SmartInternzProjectandAssignment

The link to my demo video and some files used is : https://drive.google.com/drive/folders/1r2B4CfLQMbn0Y_75-givZp52AGI3TYq2?usp=share_link