

***A Mini-Project Report On***

**“****Image Captioning”**

***Submitted By***

**Kunal Dhavale**

**PRN:** **1132220648**

**Sakshi Salvi**

**PRN: 1132221110**

**Shrutika Daundkar**

**PRN: 1132220969**

**Aniket Lavhe**

**PRN: 1132220262**

**S.Y. M.Sc. (Data Science and Big Data Analytics)**

**School of Computer Science**

**Faculty of Engineering and Technology**

**Dr. Vishwanath Karad MIT – World Peace University**

**Pune - 411038**

**Academic Year 2023-2024**

**Nov - 2023**

**Dr. Vishwanath Karad MIT WORLD PEACE UNIVERSITY, PUNE**

**SCHOOL OF COMPUTER SCIENCE**

**Certificate**

This is to certify that

**Kunal Dhavale**

**PRN:** **1132220648**

**Sakshi Salvi**

**PRN: 1132221110**

**Shrutika Daundkar**

**PRN: 1132220969**

**Aniket Lavhe**

**PRN: 1132220262**

Of **M.Sc. (Data Science and Big Data Analytics)** successfully completed his/her

Mini-Project in

**“Image Captioning”**

to our satisfaction and submitted the same during the academic year 2023-2024 towards the partial fulfilment of degree of **Master of Science in Data Science and Big Data Analytics of** Dr Vishwanath Karad MIT World Peace University under the School of Computer Science, MIT WPU, Pune.

**Prof. Dr. Shubhalaxmi Joshi Prof. Riddhi Panchal Prof. Riddhi Panchal**

**Associate Dean Program Head Assistant Professor**

**Faculty of Science School of Computer School of Computer**

**Science Science**

**MITWPU MIT WPU MIT WPU**

**ACKNOWLEDGEMENT**

In the accomplishment of this project, I would like to express my special thanks of gratitude to my teachers **Prof. Riddhi Panchal**, School of Computer Science, Dr. Vishwanath Karad MIT World Peace University whose valuable guidance has been the ones that helped me patch this project and make it full proof success. His/her suggestions and instructions have served as the major contributor towards the completion of the project.

As we were working in a group, I would like to thank my group members for their fabulous support throughout the completion of the project. We learned a lot of things during this period, as it was hard to work in this time of adversity; we were in touch with each other throughout the period and shared everything which was important from the aspect of our project. As this project was completed by staying at home, I would also like to thank our families for their cooperation and for providing facilities to us.

**Kunal Dhavale**

**PRN: 1132220648**

**Sakshi Salvi**

**PRN: 1132221110**

**Shrutika Daundkar**

**PRN: 1132220969**

**Aniket Lavhe**

**PRN: 1132220262**

**Contents**

|  |  |
| --- | --- |
| **Introduction** |  |
| Domain Name | 4 |
| Motivation | 4 |
| Problem Statement | 5 |
|  |  |
| **Literature Survey** | 6-7 |
|  |  |
| **Solution Design** |  |
| Solution Approach | 8-9 |
| Technology Stack | 10 |
| Design Model | 11-12 |
|  |  |
| **Solution Implementation and Results** |  |
| Obtaining Data | 13 |
| EDA | 13 |
| Pre-Processing | 14-15 |
| Algorithms Used | 16-17 |
| Results | 18 |
|  |  |
| **Conclusion and Future Work** |  |
| Conclusion | 19 |
| Future Work | 19 |
|  |  |
| **References** | 20 |

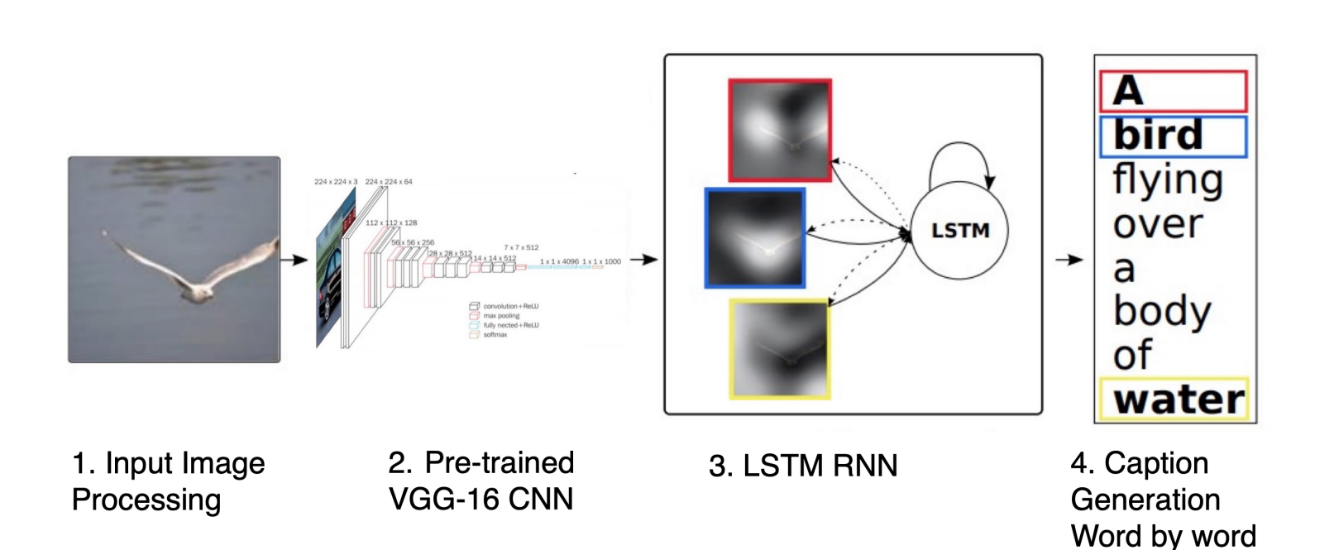
1. **INTRODUCTION:**

Your brain can quickly deduce the meaning of an image that we viewed, but can a computer also discern the meaning of an image? Researchers in computer vision spent a lot of time working on this and thought it was impossible until recently! We can create models that can provide captions for images thanks to advances in deep learning techniques, large datasets, and computing power.

This is what we intend to do in this Python-based project, which will combine the deep learning methods of LSTMs—a sort of recurrent neural network—and convolutional neural networks.

The goal of an image caption generator is to identify the context of a picture and provide a natural language description—such as English—using principles from computer vision and natural language processing.

Image captioning models can be advanced by using the MS COCO dataset as a catalyst. Deep neural networks require a rich tapestry of textual and visual data, which is provided by this collection of photos that is both wide and painstakingly annotated. The objective of this project is to use deep learning to interpret the complex contextual relationships and variety of objects found in everyday scenarios as represented in the MS COCO dataset.



**Motivation:**

We think deep learning can have a significant impact on how humans perceive and interact with images, which is why we have chosen to employ it in this image captioning project.

Our main aims are:

Identifying DL techniques for object detection and relation finding mechanism.

Identifying DL methods to handle challenges of image captioning.

Identifying DL techniques for Language generation as well as object detection.

Comparison between several techniques.

Identifying different datasets used for image captioning.

Identifying different methods to evaluate image captioning models.

**Problem Statement:**

Creating a natural language description of an image is a difficult task that is associated with image captioning. Although humans can perform this task with relative ease, machines face significant challenges due to the complex and diverse nature of natural language. Deep learning techniques have advanced significantly in the modern era, with a particular emphasis on convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These developments have demonstrated a lot of promise for the field of picture captioning. However, creating accurate and semantically meaningful captions continues to be a significant challenge. Even with the availability of rich datasets like MS COCO and recent advances in deep learning, existing models frequently fail to capture subtleties like handling unseen objects, addressing potential biases, and handling diverse linguistic styles. By utilizing the MS COCO dataset to create a novel deep learning-based image captioning model, this project seeks to address these issues by pushing the limits of automated image descriptions' precision, variety, and contextual richness.

**2**.**Literature Review:**

1. Title: **Image Captioning using Deep Learning**

Year: 2023

Authors: Mr. Raghu, Sai Srikar, Aaftaab, Ruthvik Sai

Algorithms Used: Convolution neural network, long short-term memory, Recurrent Neural Network, VGG16

Methodology: Many methodologies for image caption generation utilizing deep learning techniques are categorized as novel caption generation such as Attention based method, Novel based object captioning techniques, semantic concept-based approach.

Result: Produced relevant descriptions for a given image such as two people are hiking snowy mountain.

2.Title: **Automatic image captioning combining natural language processing and deep neural networks**

Year: 2023

Authors: Antonio M. Rinaldi, Cristiano Russo, Cristian Tommasino

Algorithms Used: CNN, RNN, Top-Down Approach

Methodology: Detection module, combining module and Captioning module used along with MASK-RCNN, YOLOv3, RetinaNet.

Result: They have generated results used by means of a captioning module to generate a description of the image content in natural language.

3.Title: **Generating image captions using deep learning algorithm RNN and LSTM**

Year: 2023

Authors: Talakoti Mamatha, Paidimarla Naveen Reddy, Kondlay Laxmi Ganesh3, Chepyala, Sathwik, A Balaram

Algorithms Used: CNN, RNN, Resnet, LSTM

Methodology: RESNET structure is utilized for extricating the photograph highlights and this picture highlights are given as enter to lengthy short term memory units and descriptions are produced.

Result: ResNet-LSTM display has better exactness in comparison to CNN-RNN and VGG show.

4. Title: **Visual Image Caption Generator Using Deep Learning**

Year: 2019

Authors: Priyanka Kalena, Nishi Malde, Aromal Nair, Saurabh Parkar

Algorithms Used: Convolution neural network, long short-term memory, Recurrent Neural Network, Gated Recurrent Unit, VGG, Encoder-Decoder.

Methodology: The system combines three models for optimizing image caption description: Feature Extraction, Encoder, and Decoder models.

Result: The model is imperfect and may generate incorrect captions. In the next phase, we will develop models using Inceptionv3 instead of VGG as the feature extractor.

5. Title: **Apply Deep Learning-based CNN and LSTM for Visual Image Caption Generator**

Year: 2023

Authors: N. Indumathi, R.J. Divyalakshmi, J. Stalin, V. Ramachandran, P. Rajaram

Algorithms Used: Deep Learning, Convolution neural network, long short-term memory, Recurrent Neural Network

Methodology: CNN is useful for image processing, categorizing images, and determining image types; LSTM is a recurrent neural network for sequence prediction, outperforming regular RNN with short-term memory.

**3. Solution Design:**

**3.1 Solution Approach:**

The solution approach for your image captioning project using deep learning involves several steps.

**1.Data Collection:**

We collected our data by downloading MSCOCO dataset. We used this dataset for our project as it offers a variety of data for training image captioning models, making it one of the most useful datasets for image captioning.

**2.Data Preprocessing:**

We cleaned and preprocessed the images and captions to guarantee consistency and quality.

Image Preprocessing-We preprocessed images by altering their image sizes i.e., Resizing image.

Captions Preprocessing-caption preprocessing was done using tokenization.

**3.Model Implementation:**

Convolutional neural networks (CNNs) for image feature extraction.

We used a pretrained CNN model named VGG16.

We implemented every layer of VGG16 Model and omitted dense layer.

Recurrent neural networks (RNNs) or transformers for generating captions.

Here we used LSTM for generating accurate captions for images.

**4.Data Generator:**

We used data generator to load data as we are working which large dataset which utilizes large amount of memory space. To avoid the problem which arises due to memory space, loading data in batches essential and it is done using data generator.

**5.Loss Function:**

We used categorical cross entropy as our loss function to measures the difference between the predicted caption and the actual caption.

Categorical cross entropy- Used for multi-class classification problems. Similar to binary cross-entropy but extended to multiple classes.

**6.Optimization Algorithm**:

Backpropagation and optimization algorithm used by us to update the model parameters is Adam.

Adam (Adaptive Moment Estimation)-Adam combines ideas from RMSprop and Momentum. It adapts the learning rates of each parameter based on both the first-order momentum and the second-order acceleration of the gradients. It is widely used and often performs well across different types of deep learning tasks.

**7.Evaluation Metrics:**

We used BLEU for defining appropriate evaluation metrics to assess the performance of your model.

BLEU-It is a metric that measures the similarity between the generated caption and one or more reference captions. It is based on precision, where precision accounts for the number of overlapping n-grams (contiguous sequences of n items) between the generated and reference captions.

**8.Hyperparameter Tuning:**

Fine-tune the hyperparameters of your model to optimize its performance. This may involve adjusting learning rates, batch sizes, or other parameters specific to your chosen architecture.

**9.Validation and Testing:**

Validate your model on a separate validation dataset to ensure it generalizes well. Test the model on unseen data to evaluate its real-world performance.

**Technology Used:**

**1.Deep Learning Framework:**

TensorFlow: Choose one of these popular deep learning frameworks for building and training your image captioning model.

**2.Model Architecture:**

CNN (Convolutional Neural Network) for Image Feature Extraction:

Use a pre-trained CNN 🡪VGG16 & Inception to extract features from images.

RNN (Recurrent Neural Network) or Transformer for Caption Generation:

Implement an RNN-based sequence-to-sequence model or a transformer architecture for generating image captions.

**3.Data Preprocessing:**

🡪Python: Utilize Python for data preprocessing tasks.

🡪OpenCV: For image processing tasks such as resizing, normalization, and augmentation.

🡪NLTK: For natural language processing tasks like tokenization.

**4.Model Training:**

GPU Support:

Leverage GPUs (Graphics Processing Units) to accelerate model training. Platforms like Google Collab, CUDA Toolkit provide GPU support.

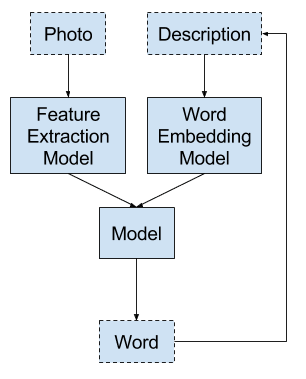
**3.3 Models:**

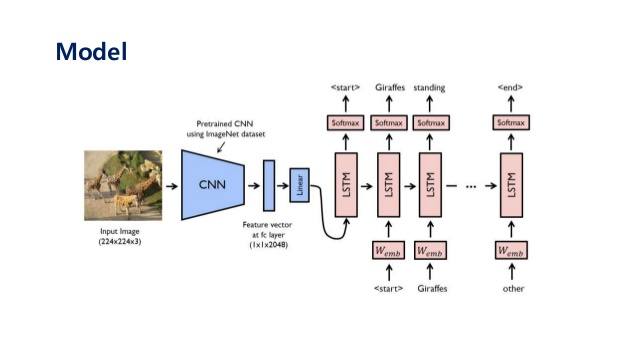
**Image caption generator model:**

To make image caption generator model, we will be merging CNN (convolutional neural network) and RNN (Recurrent neural network). It is also called a CNN-RNN model.

**CNN**: It is used for extracting features from the image. We are using a pretrained CNN model called VGG16 for extracting features from our data.

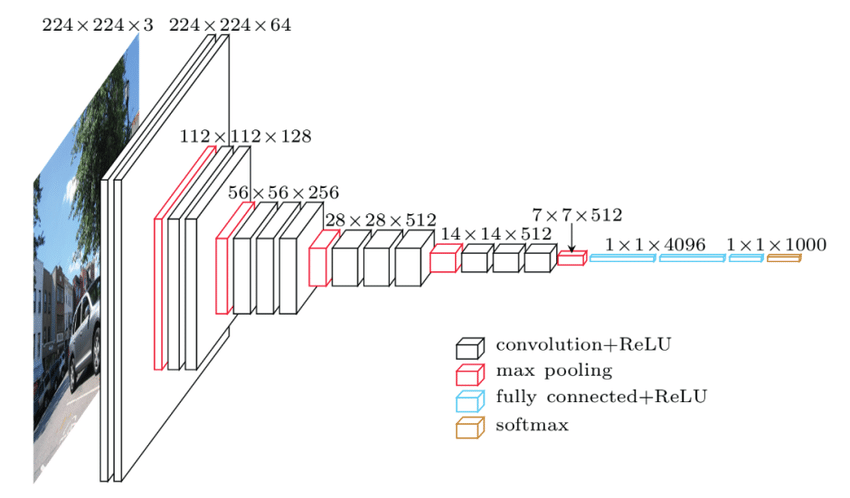
**RNN**: It is used for generating information for extracted features. We are using LSTM which will use the information from CNN to help generate a description of the image.

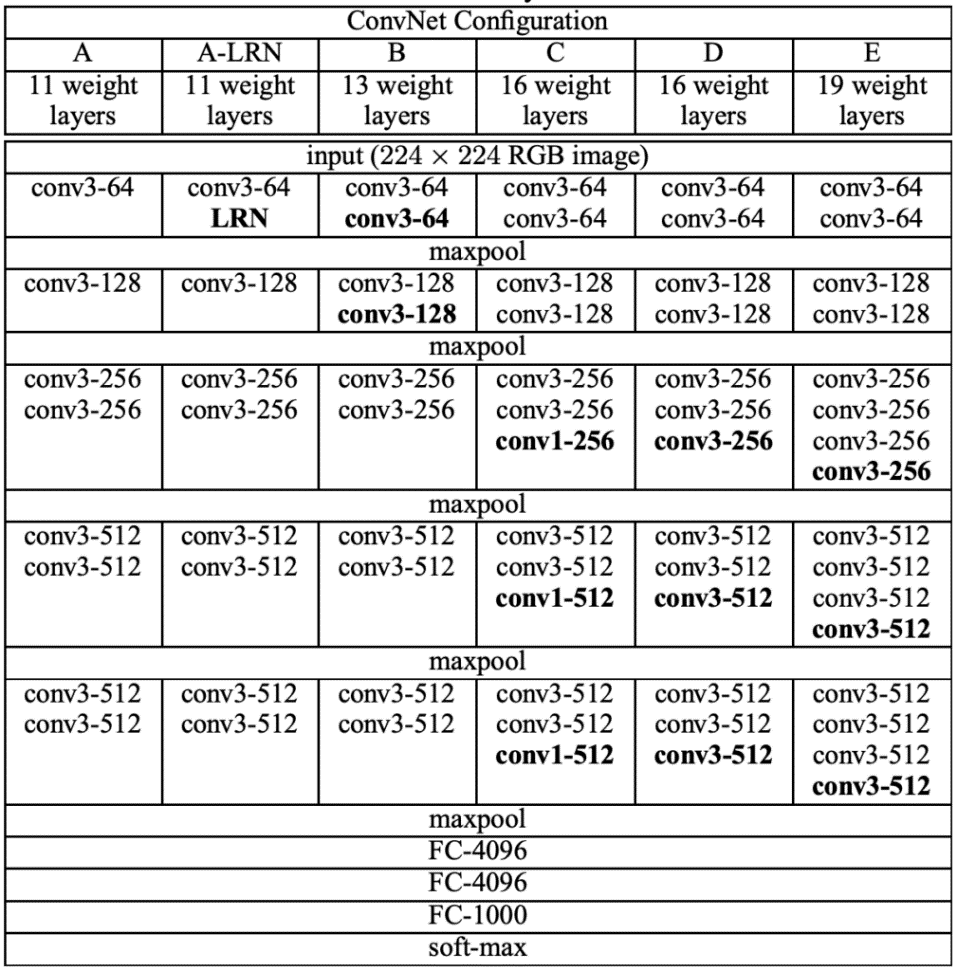




**VGG16**:

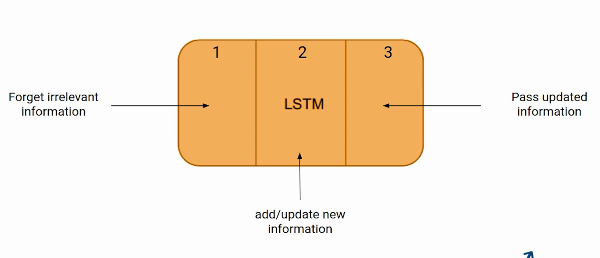
The VGG16 represents a Convolutional Neural Network (CNN) architecture that emerged as the winner of the 2014 ILSVR (Imagenet) competition. This vision model architecture has gained recognition as a paradigm of exceptional quality. One notable characteristic of the VGG16 model is its emphasis on utilizing 3x3 filter convolution layers with a stride of 1, in lieu of a vast number of hyper-parameters. Additionally, the model consistently employs identical padding and a 2x2 filter stride 2 maxpool layer. In the architectural design, the convolution and maximum pooling layers are organized in a congruous manner. The object under consideration possesses a pair of FCs. The ultimate layer comprises fully interconnected layers, culminating in a softmax function for output. Notwithstanding, the ultimate layer shall be eliminated as solely the image's features are requisite. The nomenclature "VGG16" denotes that the model comprises a total of sixteen stratified layers, each endowed with variant weight parameters. The magnitude of this network is formidable, as it encompasses an estimated 138 million parameters. The provided model was trained using the Imagenet dataset, which consists of 14 million photographs that are classified into 1000 distinct categories.

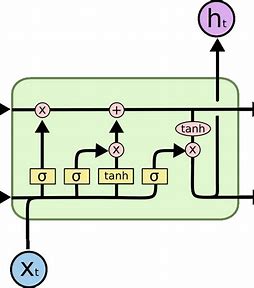




**LONG SHORT-TERM MEMORY (LSTM**):

Long short-term memory (LSTM) is a kind of RNN (recurrent neural network) that excels at sequence prediction tasks. Based on the previous paragraph, we can estimate what the next word will be. In terms of overcoming the restrictions of RNNs with short term memory, it beat ordinary RNNs. The LSTM may carry out useful information throughout input processing and discard irrelevant information using a forget gate. LSTMs, as opposed to traditional RNNs, are designed to avoid the problem of vanishing gradients and store information over long periods of time. LSTMs can maintain their error constant, allowing them to learn back propagation across time and layers over many time steps.





**INCEPTION V3 MODEL:**

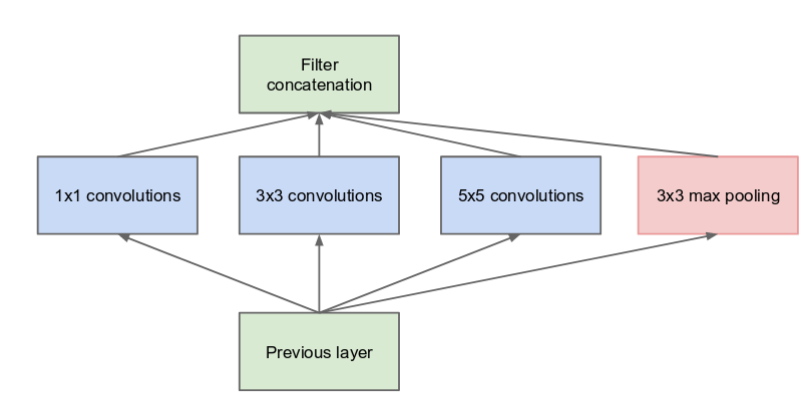
The Inception V3 is a deep learning model based on Convolutional Neural Networks, which is used for image classification. The inception V3 is a superior version of the basic model Inception V1 which was introduced as Google Net in 2014. As the name suggests it was developed by a team at Google. When multiple deep layers of convolutions were used in a model it resulted in the overfitting of the data. To avoid this from happening the inception V1 model uses the idea of using multiple filters of different sizes on the same level. Thus, in the inception models instead of having deep layers, we have parallel layers thus making our model wider rather than making it deeper. Inception model is made up of multiple Inception modules.

The basic module of the Inception V1 model is made up of four parallel layers.

1. 1×1 convolution
2. 3×3 convolution
3. 5×5 convolution
4. 3×3 max pooling

**Convolution** - The process of transforming an image by applying a kernel over each pixel and its local neighbors across the entire image.

**Pooling** - Pooling is the process used to reduce the dimensions of the feature map. There are different types of pooling but the most common ones are max pooling and average pooling.

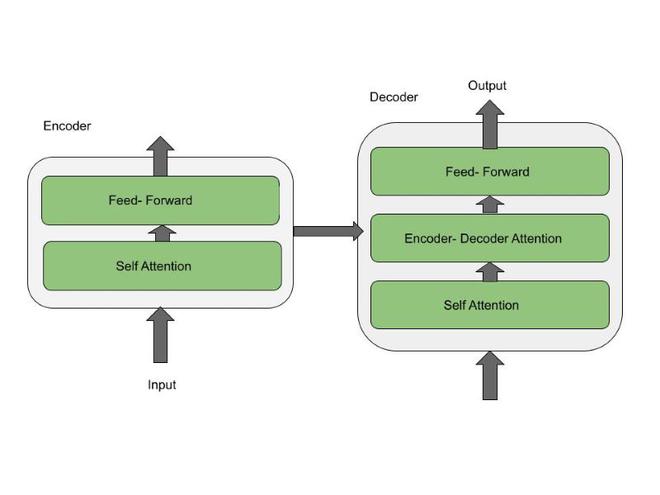


**TRANSFORMER:**

### Attention Mechanism:

The[attention mechanism](https://www.geeksforgeeks.org/ml-attention-mechanism/) in transformers employs a scaled dot-product approach, where the computation involves scaled dot products between the query, key, and value vectors. This produces weighted values that are then summed to yield the attention output. To enhance the model’s capacity to capture diverse relationships within the input, the multi-head attention mechanism is introduced. This involves applying the attention mechanism multiple times concurrently, each with distinct learned linear projections of the input. The resulting outputs from these parallel computations are concatenated and undergo a linear transformation to generate the final attention result. This multi-head approach allows the model to attend to different aspects of the input sequence simultaneously, promoting richer and more nuanced representations.

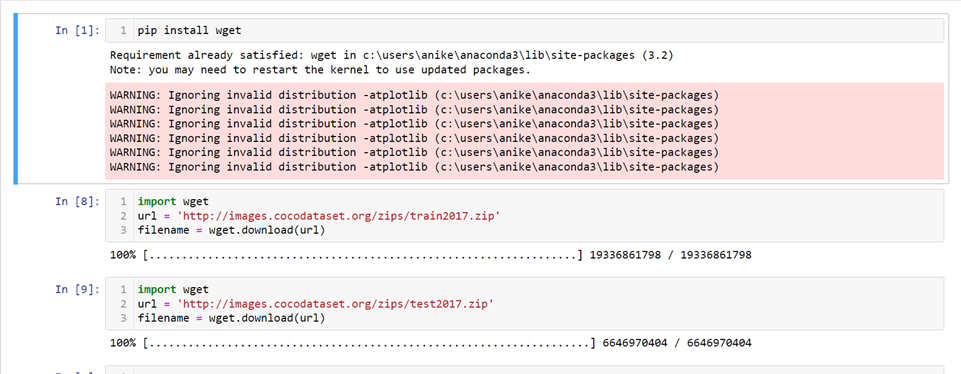
The decoder architecture has three layers: **Self Attention, Encoder-decoder attention,** and **Feed Forward**. The decoder has both the self-attention and feed-forward layer, which is also present in the encoder, but between them is an attention layer that helps the decoder focus on relevant parts of the input sentence.

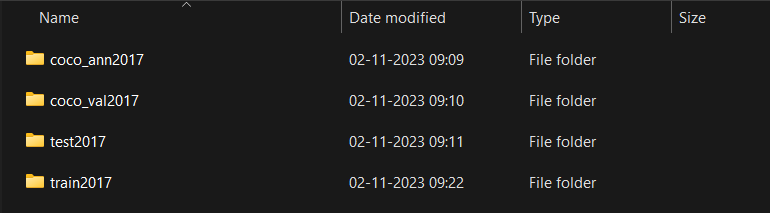


**Solution Implementation and Results:**

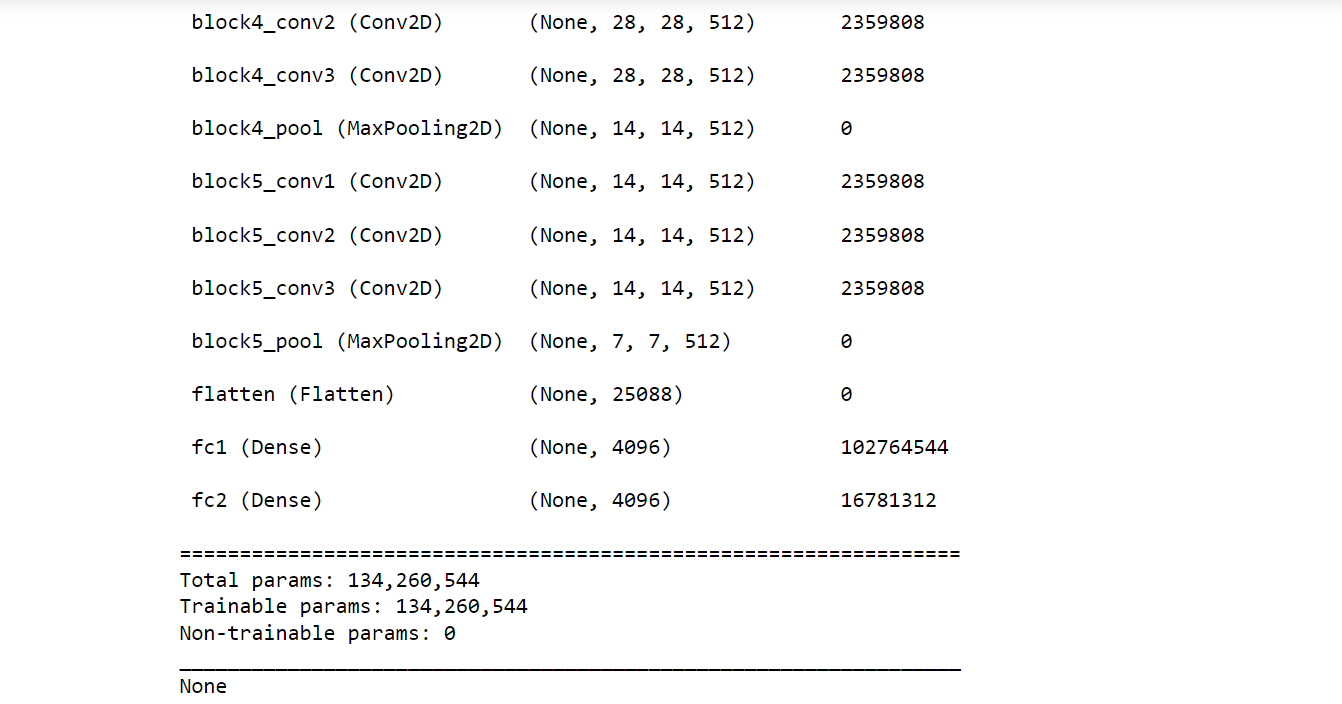
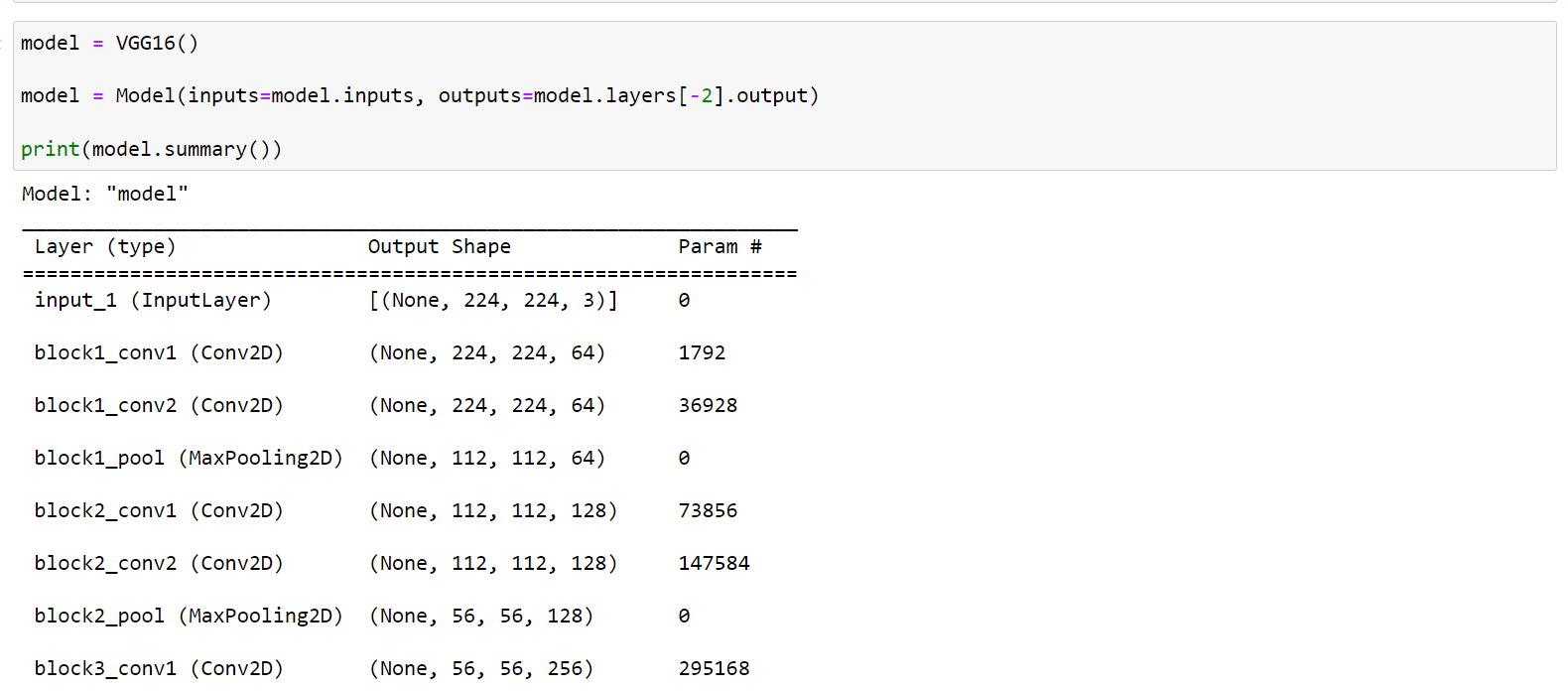
**Data Collection:**

We target the dataset link in our code to download it.

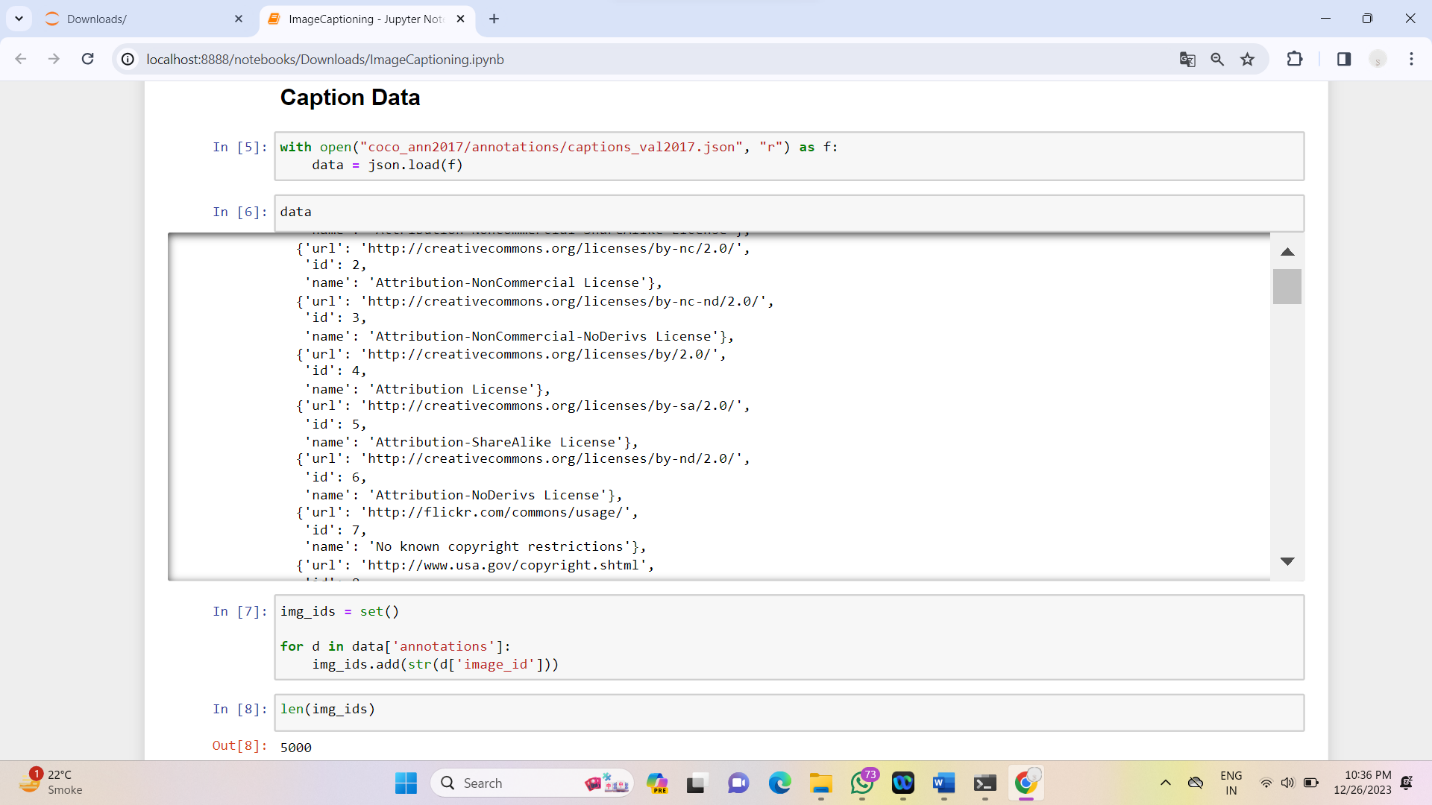




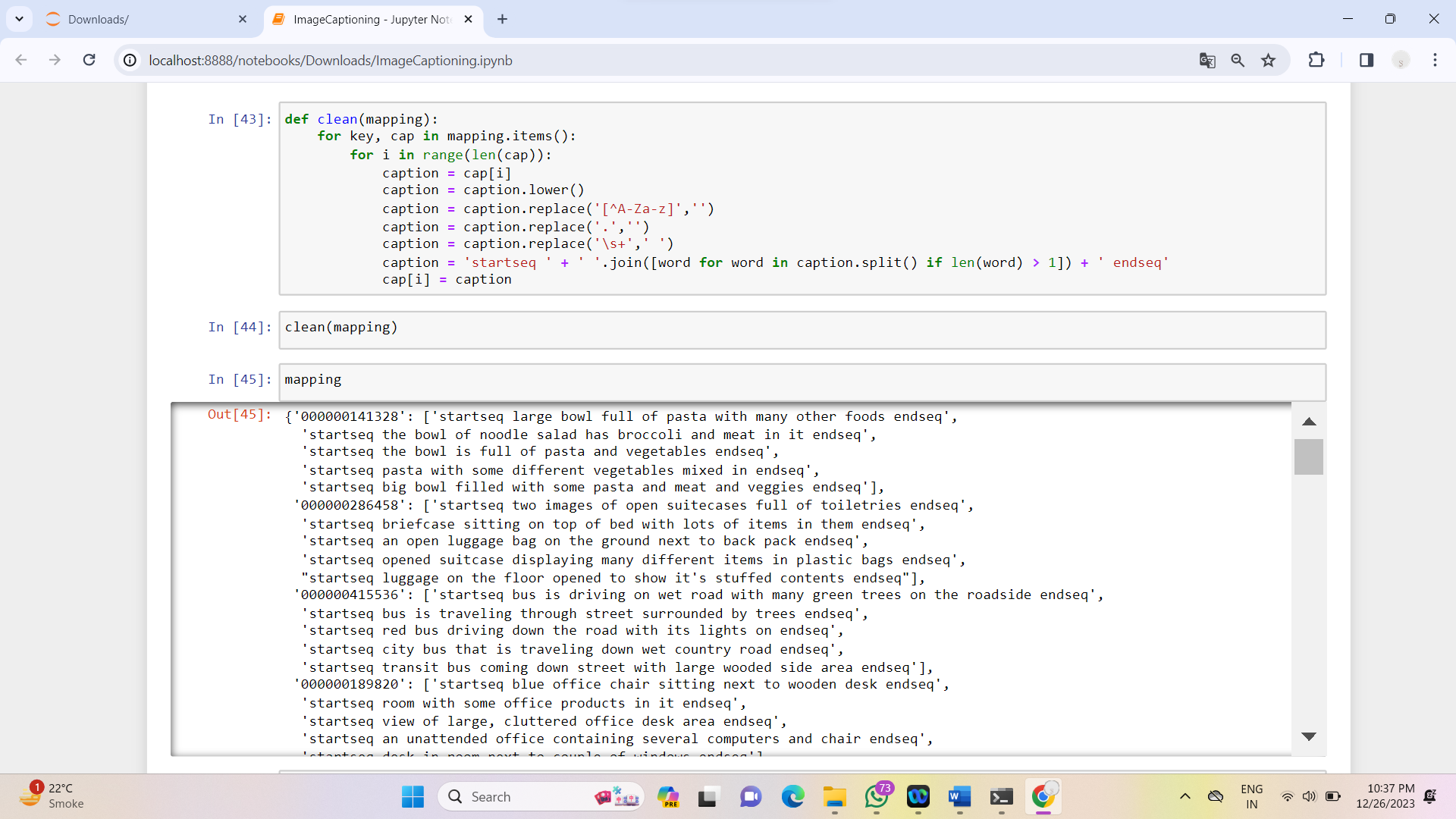
**Importing VGG16 Model :**

****

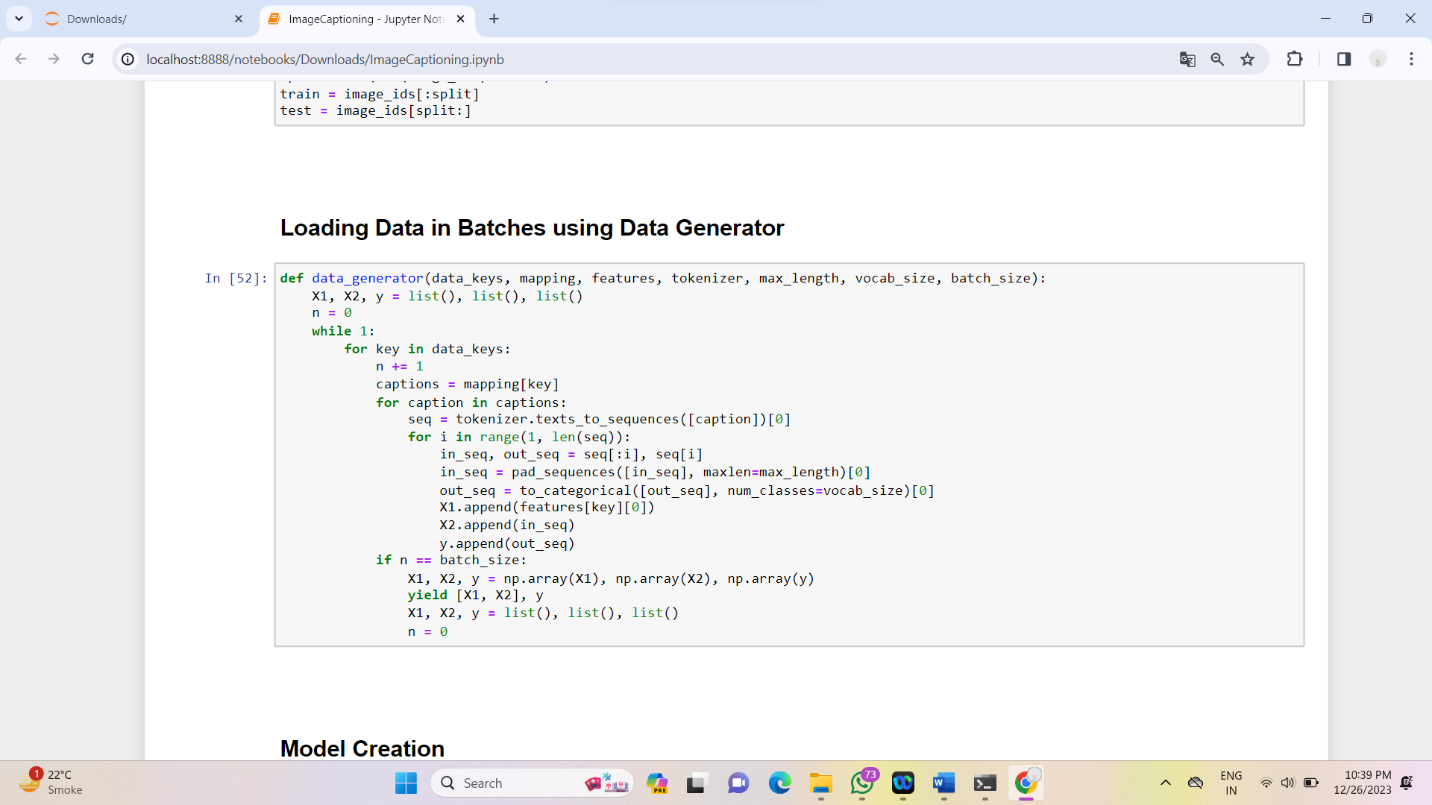
**Working with caption data:**

****

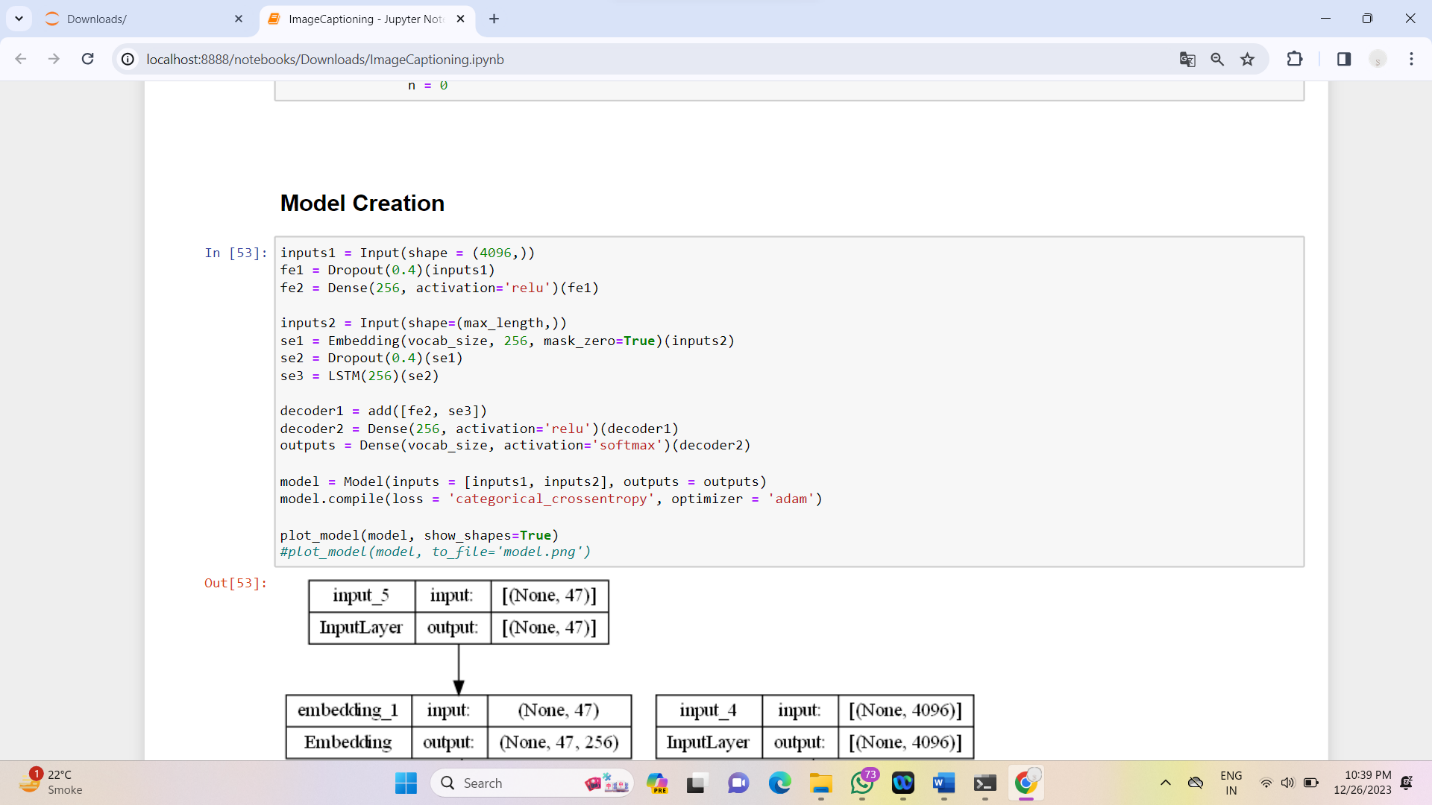
**Pre-processing of data :**

****

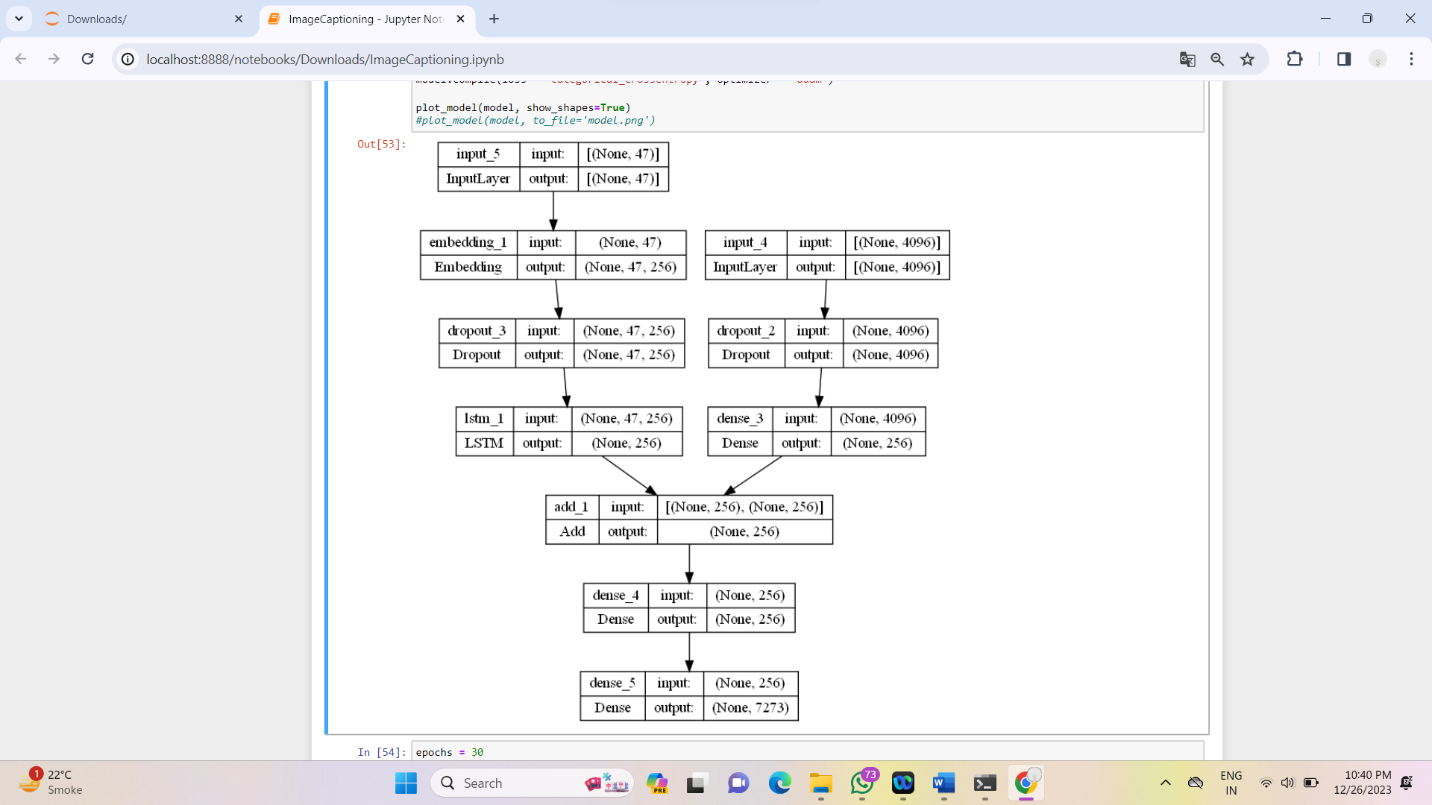
**Creation of Data Generator :**

****

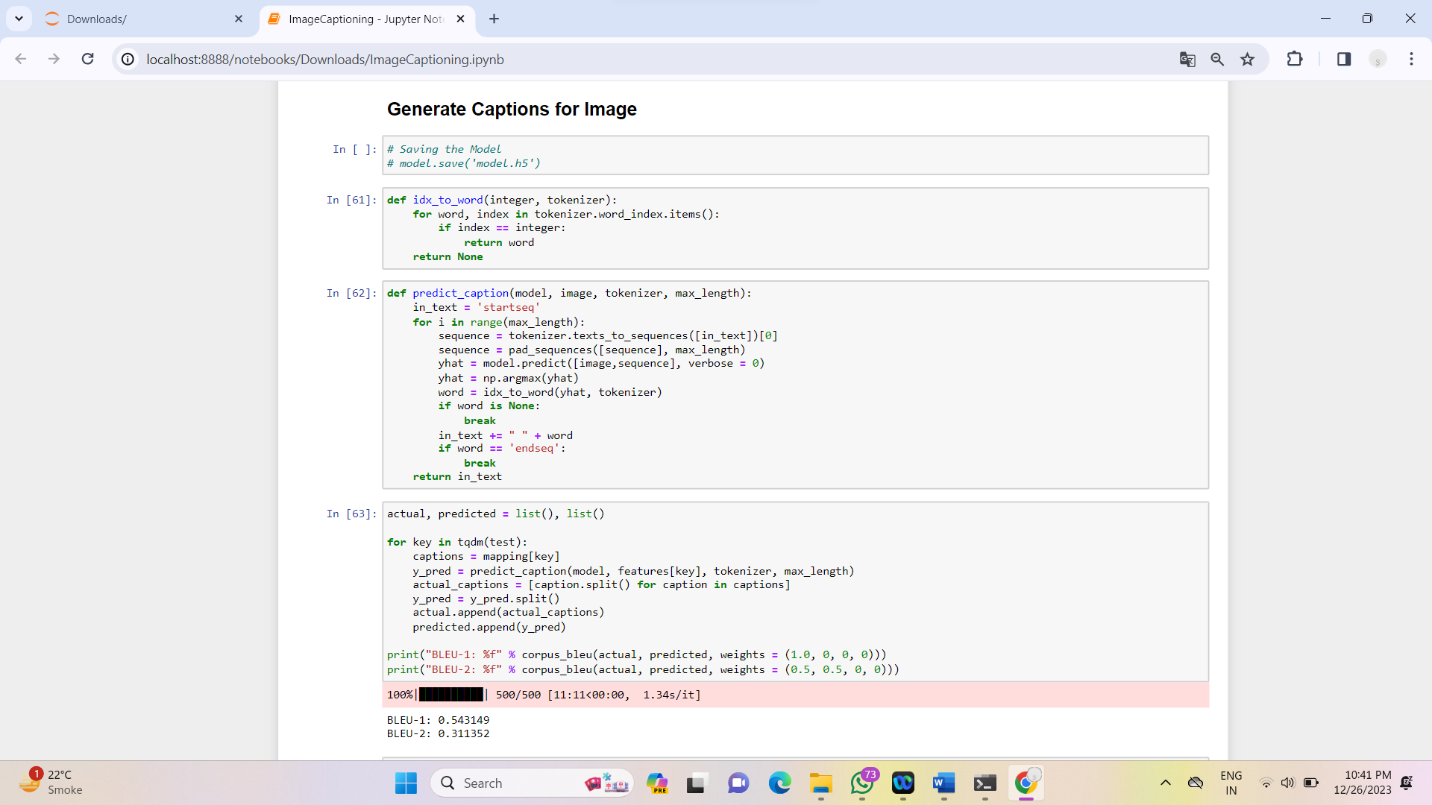
**Generating Model :**

****

**Model created :**

****

**Caption generation for Image data :**

****

**Captions Generated :**

****

**Conclusion & Future scope:**

Though there are many active research initiatives aimed at more accurate image feature extraction and semantically superior sentence creation, automatically captioning images is still far from mature.We effectively finished the tasks we indicated in the project proposal.If given additional time, there might be improvements.First off, the network does not adjust to our particular training dataset because we employed a pre-trained CNN network straight into our pipeline without any fine-tuning. We therefore anticipate getting a marginally higher BLEU4 score by experimenting with various CNN pre-trained networks and turning on fine-tuning. A MSCOCO training program is another possible way to improve.

Generally speaking, the output will be more accurate the more different training datasets the network has encountered.

We can all agree that this research has piqued our curiosity in using deep learning techniques to computer vision, and we plan to investigate more in the future.

**For the domain of Image Captioning, the quality of Captions generated is a more reliable measure of the model’s accuracy. The Transformer architecture generates quite reliable captions consistently.**

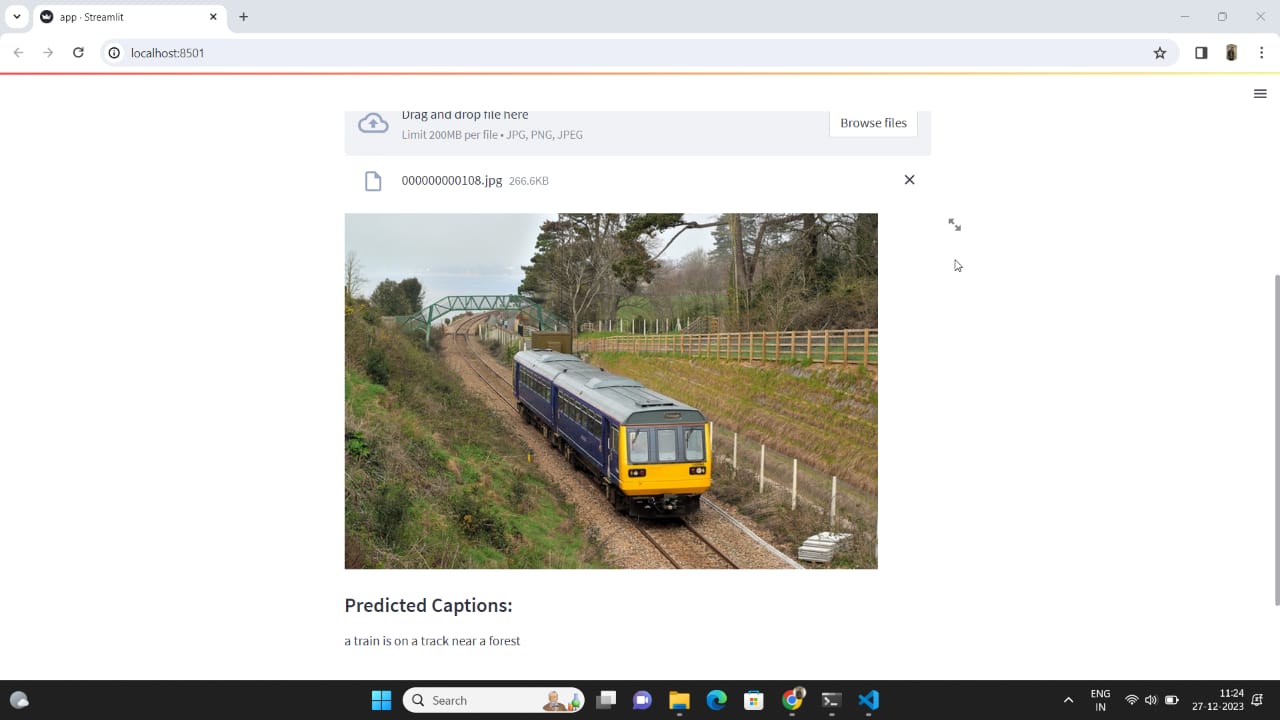
**About Streamlit:**

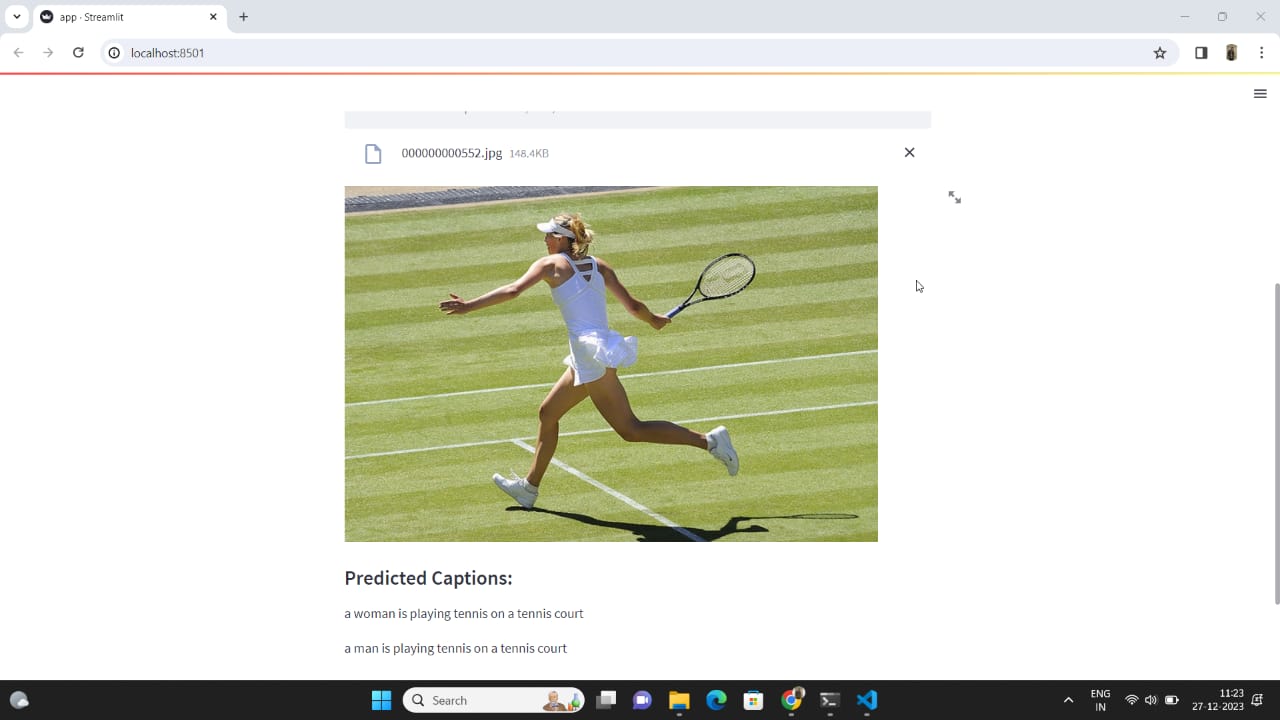
With Streamlit, an open-source Python framework, web apps for data science and machine learning projects may be quickly and easily created. It makes it simple and quick for developers and data scientists to transform data scripts into shareable web applications. Because of its straightforward and intuitive design, Streamlit is used even by people without a lot of web development knowledge.

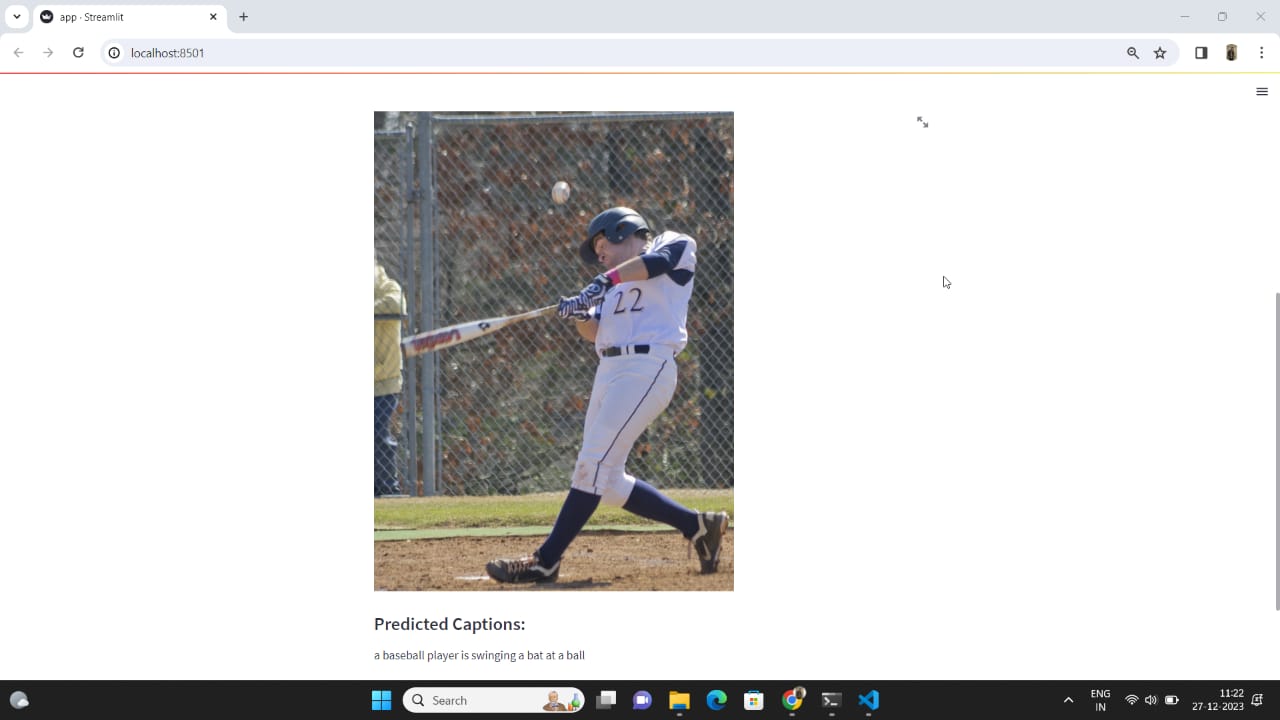
Streamlit offers a straightforward Python script-based methodology that makes it possible to construct interactive web apps quickly. Only a few lines of code are needed to convert a Python script into a web application.

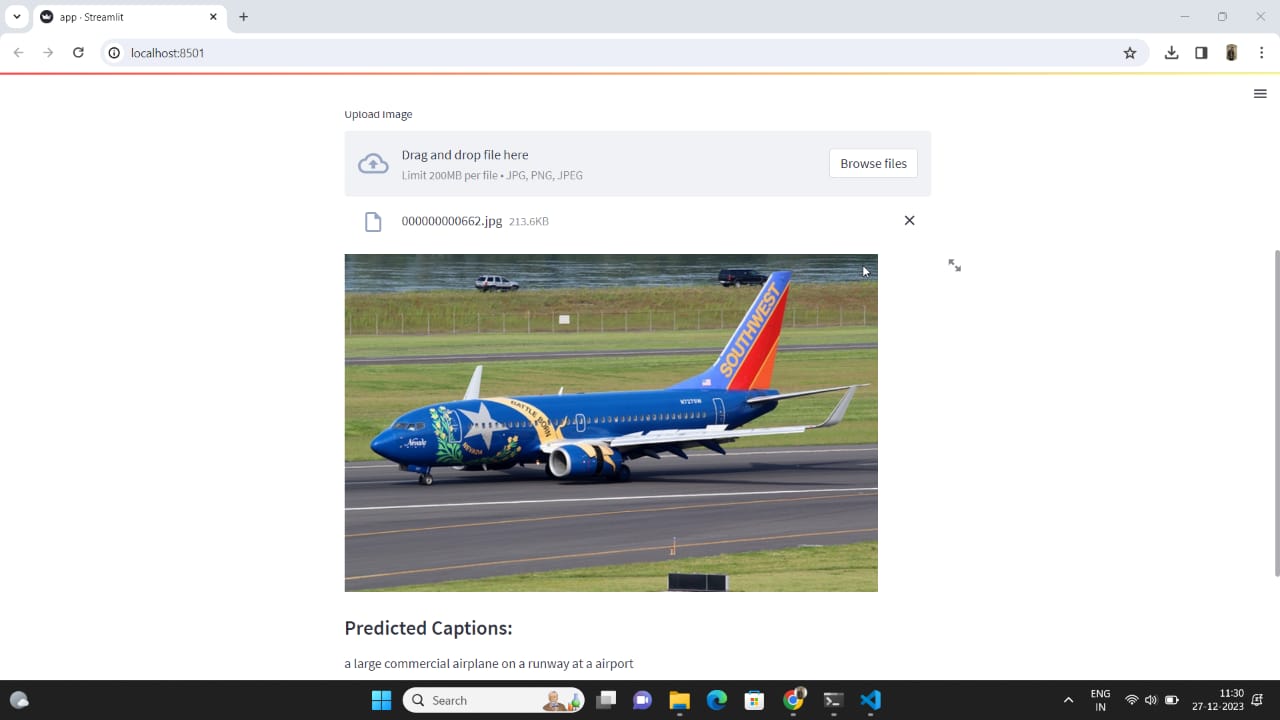
Frequently employed in the development of interactive dashboards and visualizations. It makes it simple to view and study datasets by supporting a number of data visualization libraries, including Matplotlib, Plotly.

**Predicted Output:**

****

****

****

****