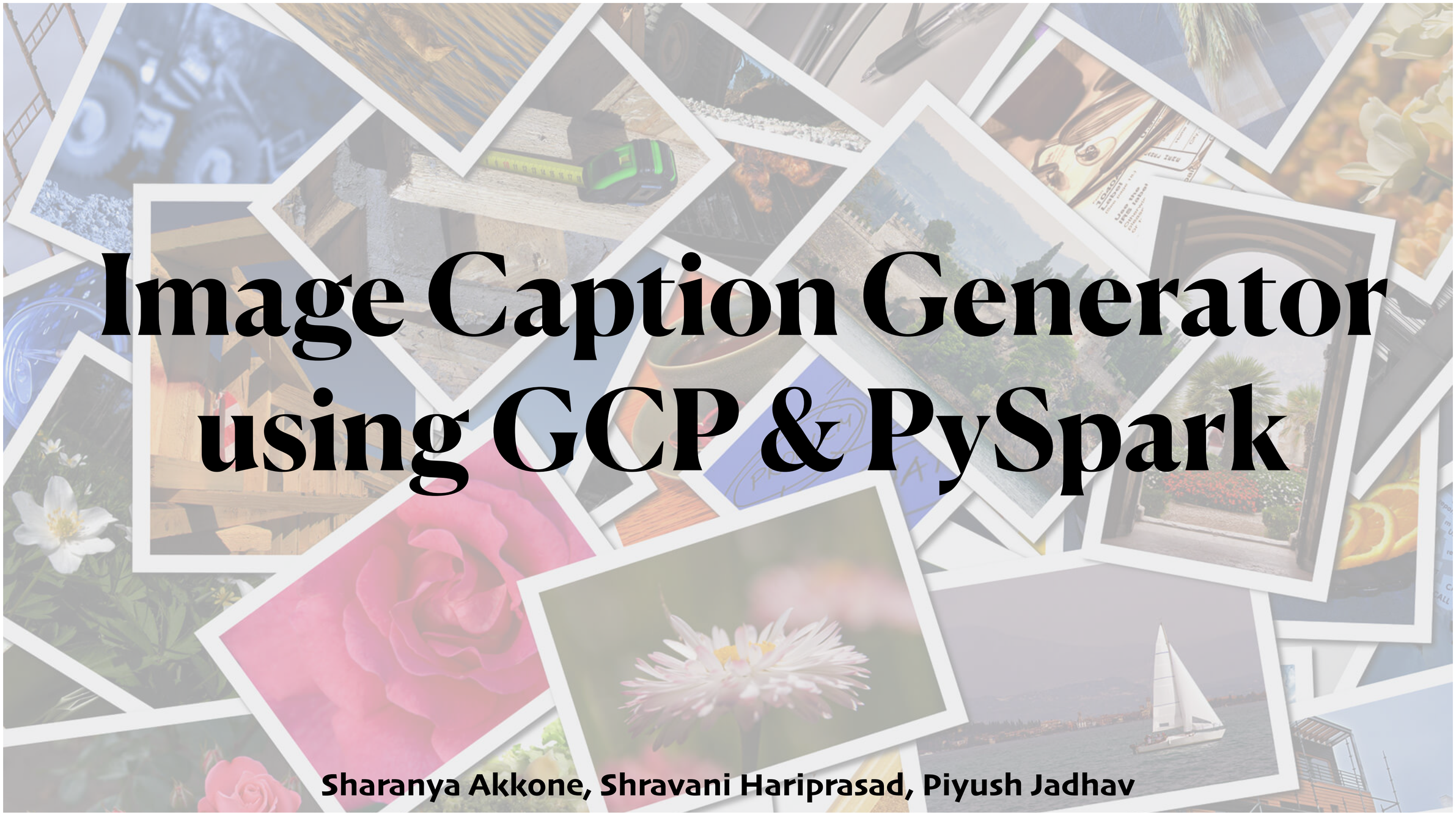


Image Caption Generator using GCP & PySpark

Sharanya Akkone, Shravani Hariprasad, Piyush Jadhav

Introduction

Goal

Develop a deep learning-based model that generates natural language descriptions that provide maximum context of input images.

Dataset

Microsoft Common Objects in Context (COCO) 2017 dataset with over 330,000 images and captions.

Model Architecture

Xception CNN as feature extractor and LSTM network as language model.

Objectives:

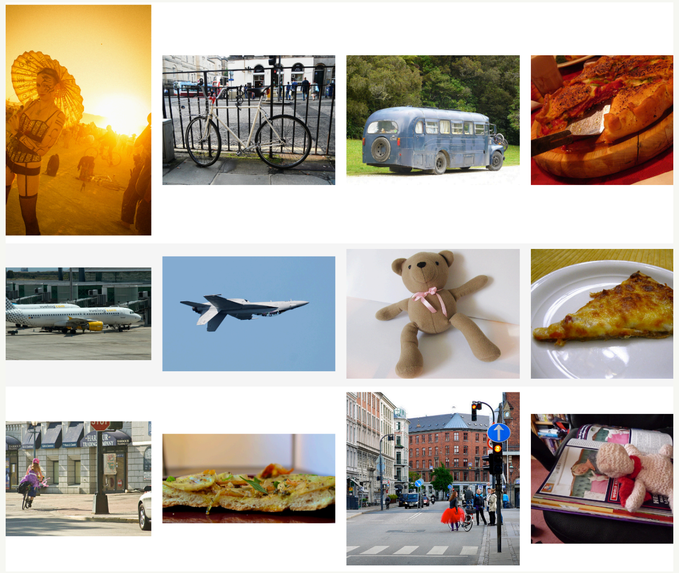
- Develop accurate and descriptive image caption generator model.
- Train model on COCO dataset and fine-tune Xception CNN model for better feature extraction.
- Evaluate model performance using standard metrics (BLEU, METEOR, ROUGE).
- Conduct exploratory data analysis (EDA) on COCO dataset to gain insights and identify potential data quality issues.
- Explore potential applications, including assistance for visually impaired, improving image search engines, and generating social media post descriptions.

Big Data and its role

- **Project** trained on a large dataset of images and captions (Microsoft COCO dataset) that contains over **330,000 images**
- **cannot** be processed by **traditional** methods.
- **Analysis**, **storage**, and **processing**
- Improve the **accuracy** and **diversity** of the generated **captions.**
- Enables us to **process** and analyze **large datasets**, which can lead to more **accurate** and **robust models** in various fields of study.

DATA

Glimpse of the dataset



```
An erotically dressed woman standing on a beach holding an umbrella.
A woman in burlesque clothing holding an umbrella
A tattled girl stands in front of a crowded gathering.
A woman standing outside with an umbrella  over her head.
A woman standing in a sunset with an umbrella in lingerie.
```

Challenges and opportunities our project faced in relation to big data

- Significant **computational resources** to train and test the deep learning models faced challenges in
- Dealing with issues such as **overfitting** and the need for **regularization** techniques to improve model performance.
- Challenges in **detailed context** of some images
- Presented opportunities for the project to **extract valuable insights** and knowledge from large datasets that were previously too large to analyze.
- Demonstrated the **potential** of big data to enable advancements in the field of **image caption generation**, with implications for various industries and society at large.

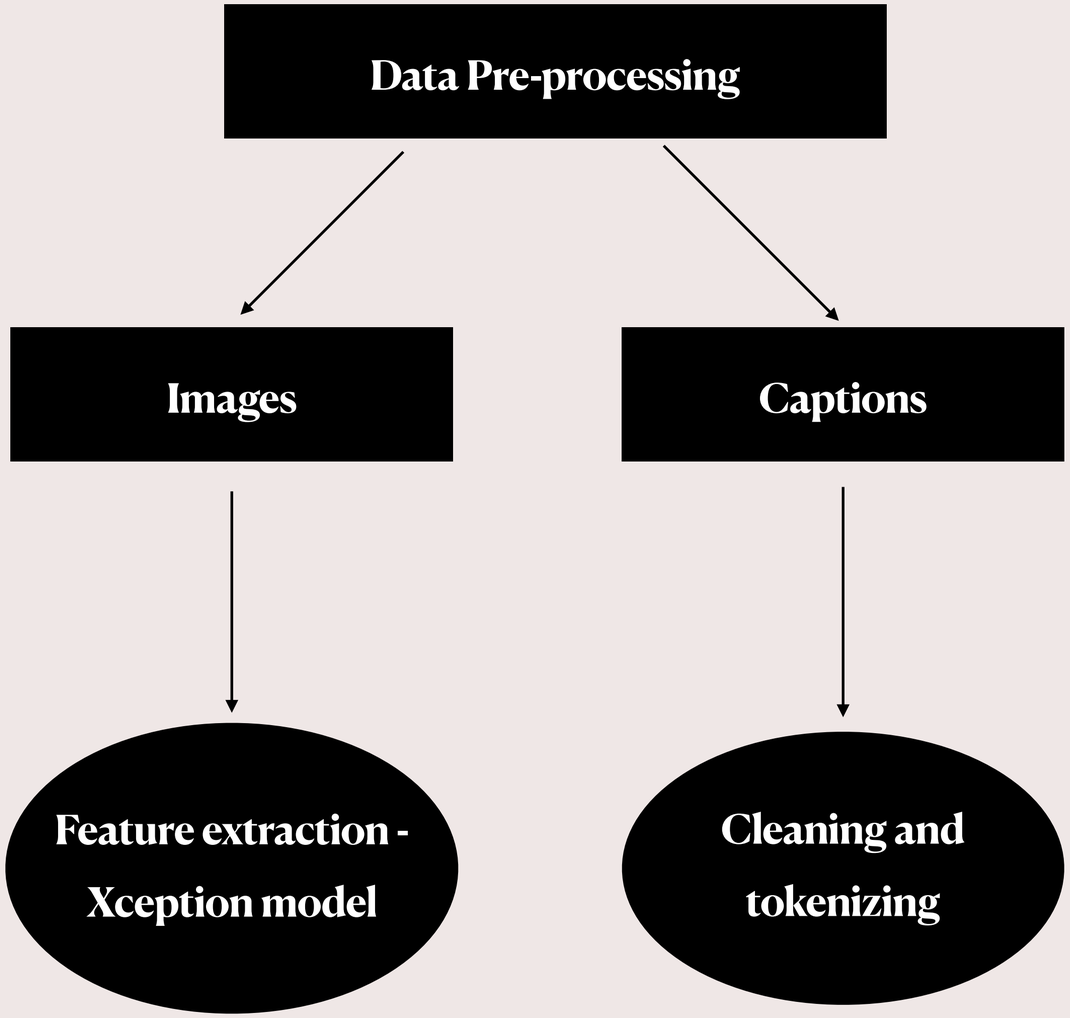
The ***Microsoft COCO 2017*** dataset was used as the primary data source, with over ***330,000*** images and ***5 captions*** for each image

Dataset was sourced from: https://cocodataset.org/#download

resizing, cropping, tokenizing, and converting to numerical vectors

split into training, validation, and test sets for training and evaluation

pre-trained Xception convolutional neural network- visual features from the images.

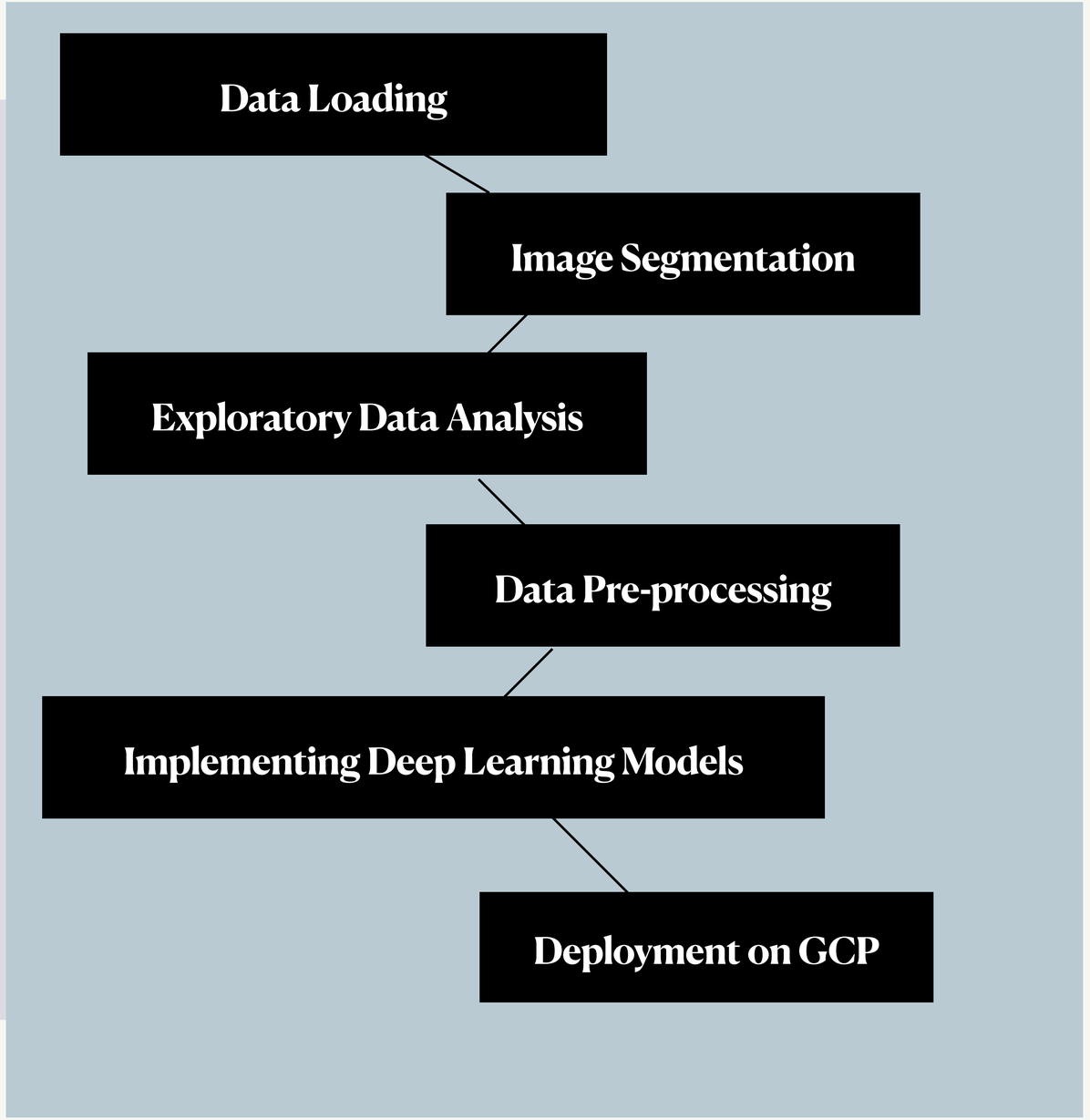

Data Pre-processing
Images
Captions
Feature extraction - Xception model
Cleaning and tokenizing

For the better good

1. e-commerce, social media, and healthcare, assist in product recommendation, content curation, and medical diagnosis.
2. potential of large-scale image datasets and deep learning models for natural language generation tasks.
3. importance of ethical considerations in data-driven technologies, such as bias in dataset annotations and model outputs, and the need for transparency and accountability in AI applications.
4. enhance accessibility for visually impaired individuals, enabling them to more fully participate in online and offline environments.
5. The project's approach to combining convolutional neural networks and recurrent neural networks for image captioning can serve as a basis for future research in the field of deep learning.
6. potential to contribute to the development of more advanced natural language generation techniques, such as chatbots and voice assistants

Methodology & Approach

Steps:


Data Loading
Image Segmentation
Exploratory Data Analysis
Data Pre-processing
Implementing Deep Learning Models
Deployment on GCP

• Data Loading

To load the COCO dataset efficiently, **PySpark** was used on a **Dataproc cluster**

coco function from the pycocotools.coco package.

PySpark's **parallelize** method was used to distribute the data across multiple nodes, which enabled faster processing of the data.

PySpark's **RDD (Resilient Distributed Dataset)** was used to extract information about the categories and subcategories present in the dataset

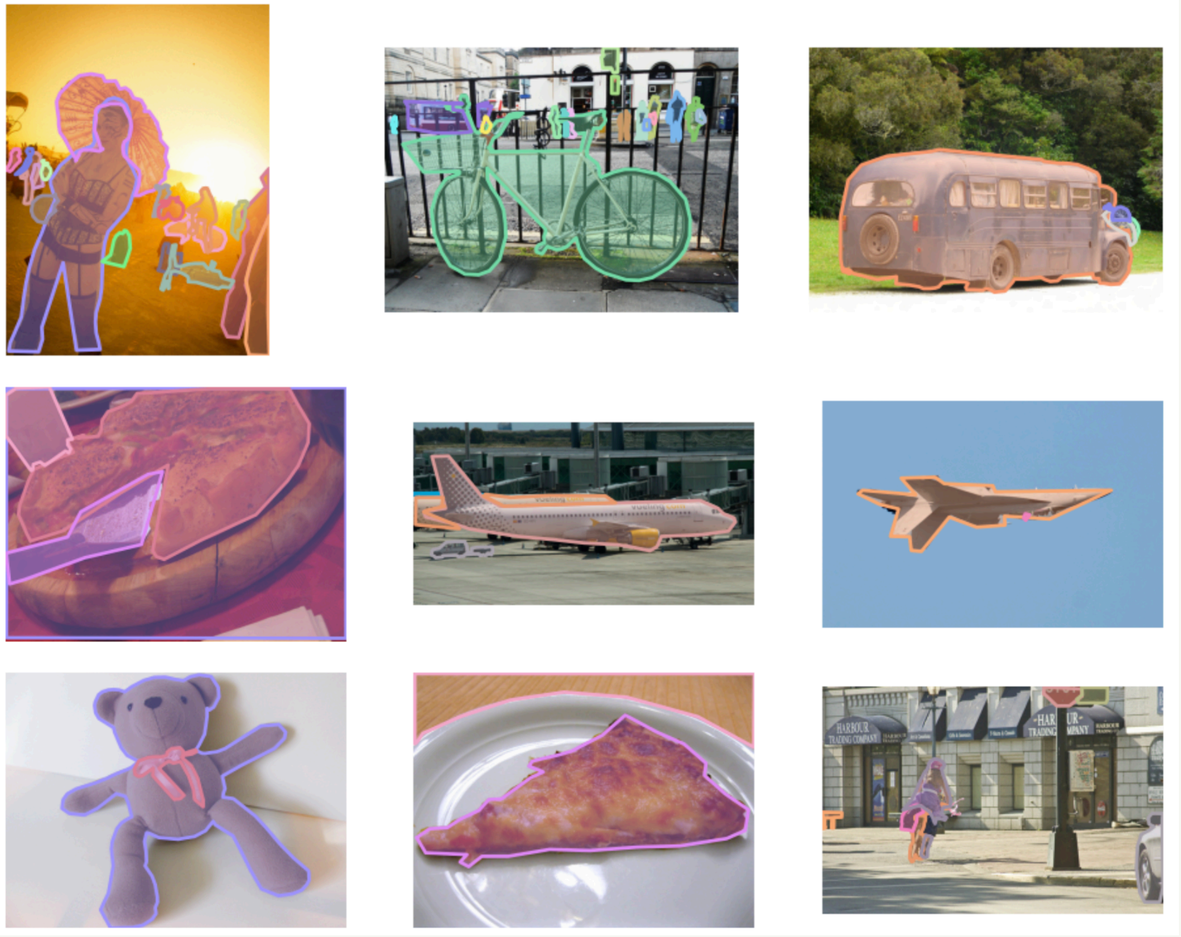
• Image Segmentation

showAnns() method is used in the code to display the segmentation **masks** for the annotated objects within an image

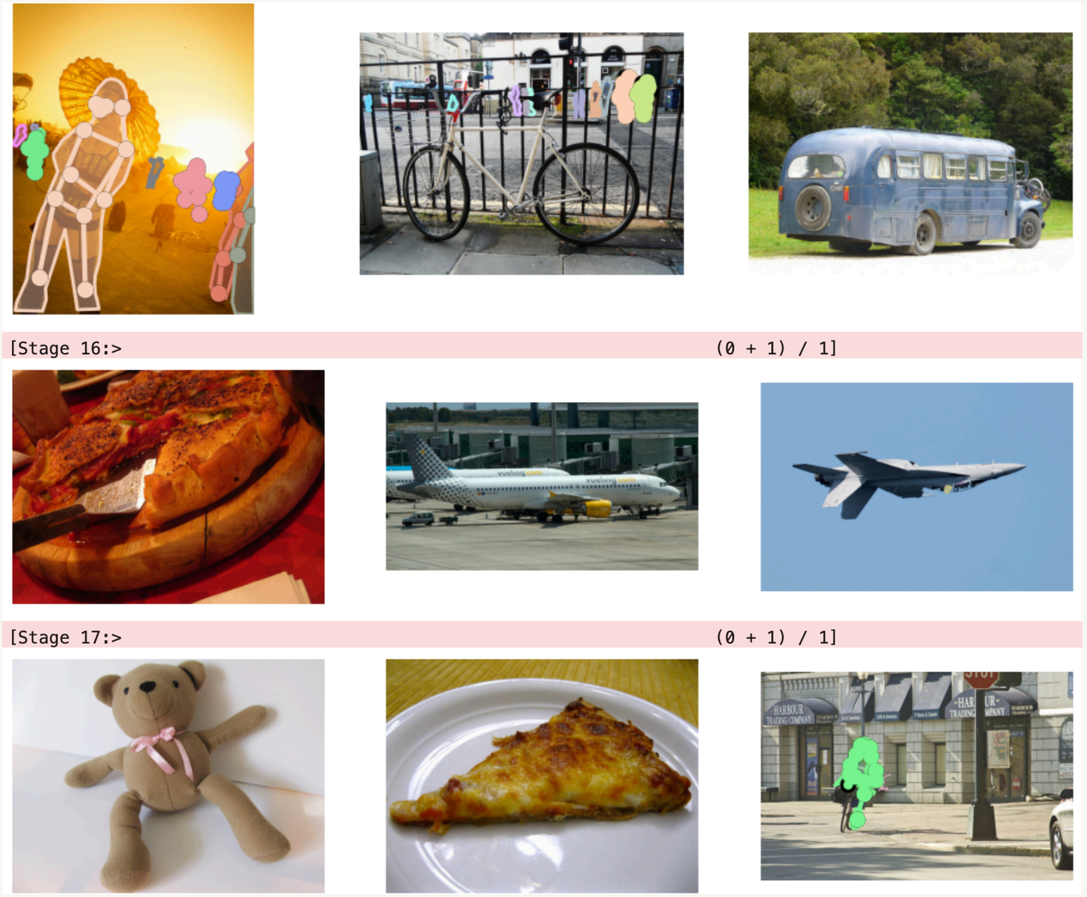
keypoint segmentation is another method of image segmentation that was used in the code to identify people in the images by locating specific body parts such as the nose, eyes, and hands

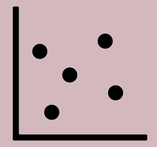
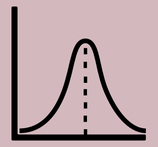
COCO API's **getAnnIds** method was used to obtain the relevant annotation IDs, and the loadAnns method was used to load the annotations from the COCO dataset, including the key point coordinates for each person in the image. Finally, the **showAnns** method was used to overlay the **key point** information onto the **original image.**

Plain Segmentation Masks



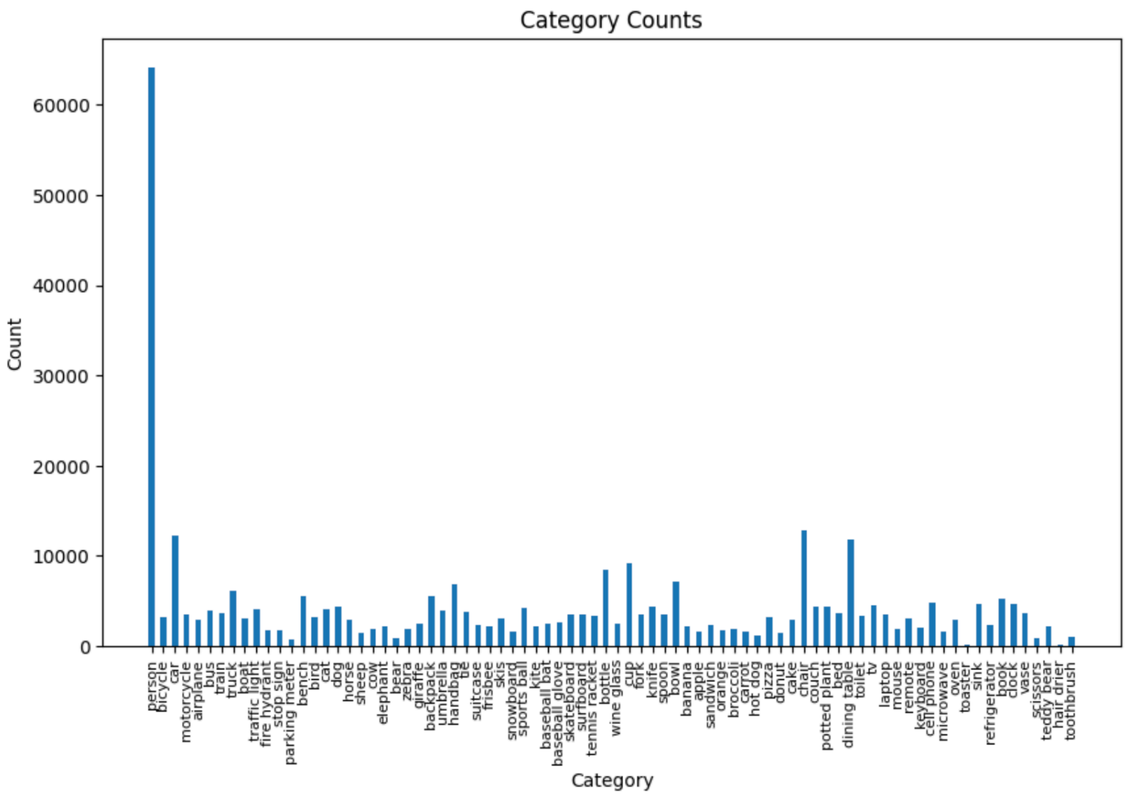
Keypoint Segmentation



Results from EDA

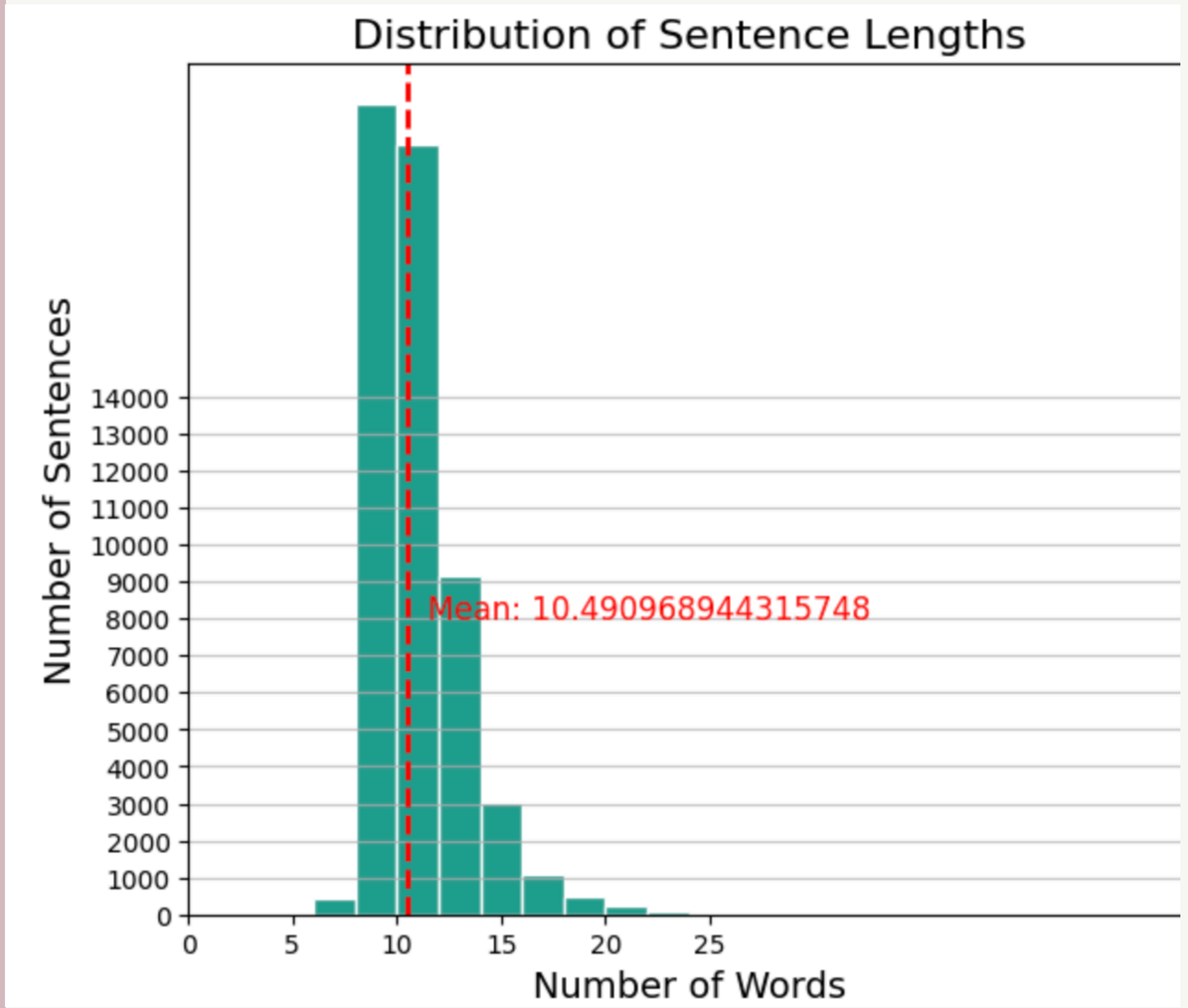
(Exploratory Data Analysis)

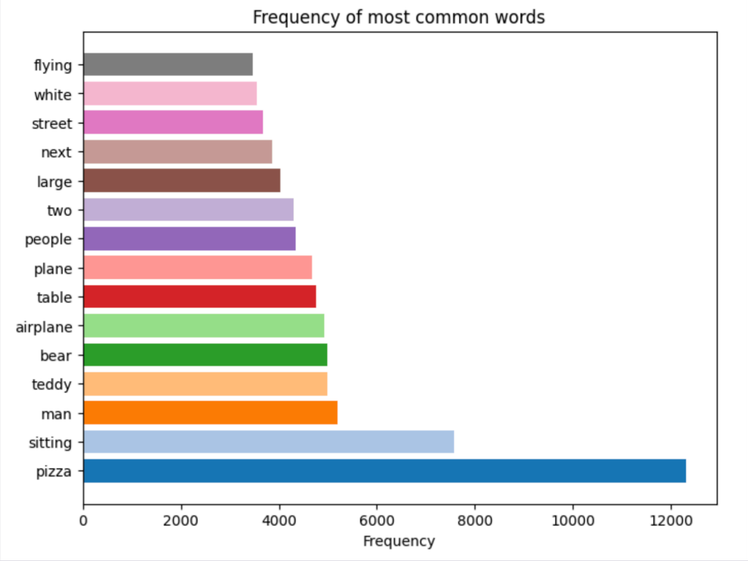



This bar graph represents the **frequency** of the category of the image

Person - most commonly detected category

On an average, there are **10 words** per sentence in a caption






```
[('pizza', 12306),
 ('sitting', 7570),
 ('man', 5204),
 ('teddy', 4988),
 ('bear', 4982),
 ('airplane', 4924),
 ('table', 4751),
 ('plane', 4672),
 ('people', 4352),
 ('two', 4304),
 ('large', 4036),
 ('next', 3856),
 ('street', 3676),
 ('white', 3543),
 ('flying', 3458)]
```

‘Pizza’ is the most common word in captions, followed by **‘sitting’** and **‘man’**

How is EDA contributing to our project?

- **distribution** and **characteristics** of the image and caption data ; informed decisions about data **preprocessing**, **model selection**, and **evaluation metrics**.
- identifying any **anomalies, outliers**, or **missing data** in the dataset ; improve in **quality** and **accuracy** of the image caption generator model.
- spark **new ideas** for feature engineering, model architecture, or evaluation metrics by providing insights into the **strengths** and **weaknesses** of the dataset.
- validates **assumptions** about the data, such as the relationship between image features and caption text, and can help in testing the **model's performance** on **different subsets** of the data.

• Data Pre-processing

Text data is preprocessed to **remove stop words** and common phrases by importing the **stopwords** module from **NLTK** package

image data is preprocessed by **resizing** and **normalizing** the images

appending the **start** and **end identifiers** to each caption which indicate the starting and ending points of a caption

important for the **LSTM** to understand the beginning and end of a **sequence**.

```
['<start> a jumbo jet plane connected to a boarding deck <end>', '<start> a large blue passenger plane
sits on the tarmac at the airport <end>', '<start> a blue commercial airplane parked at a jet way <end
>', '<start> a large airplane that is sitting out on the runway <end>', '<start> a blue plane at the ai
rport being offloaded <end>']
```

• Extraction of Feature Vectors

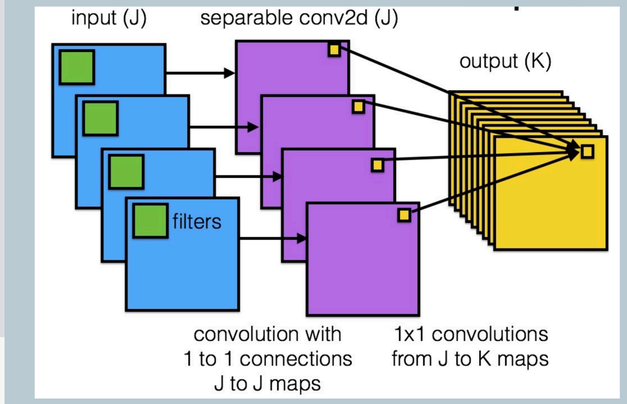
Transfer Learning, which involves using a pre-trained model (in our case the Xception model) to extract features

Convolutional Neural Network that is **71 layers deep** and has been trained on the **Imagenet dataset**

Remove the **classification** layer from the **Xception** model using **keras.applications.xception** module, which will give us the 2048 feature vector.

Download weights for each image and map image urls with their respective feature array after normalizing, and resizing to a standard size of **299x299** since the xception model takes 100 x 100 x 3 image size

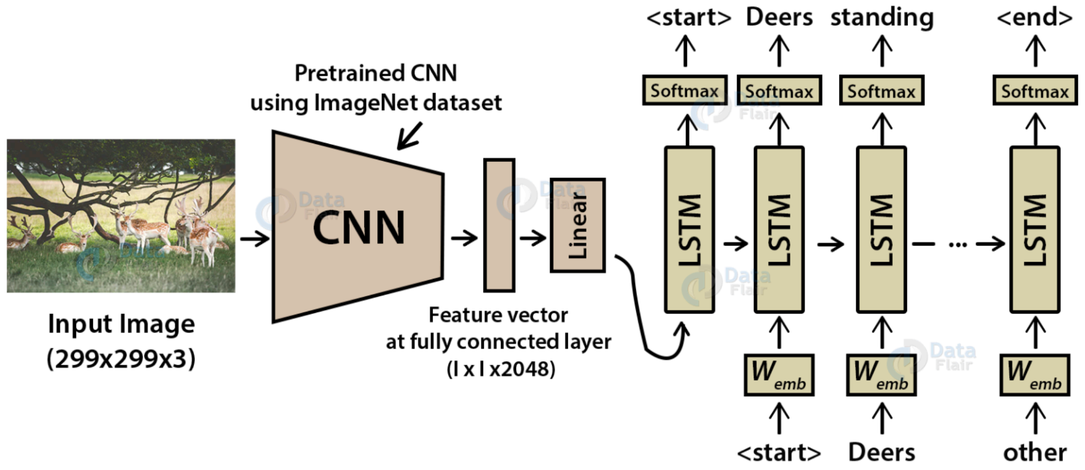
image features are then stored in a **spark dataframe** which is then stored in a RDD and **parallelized**.



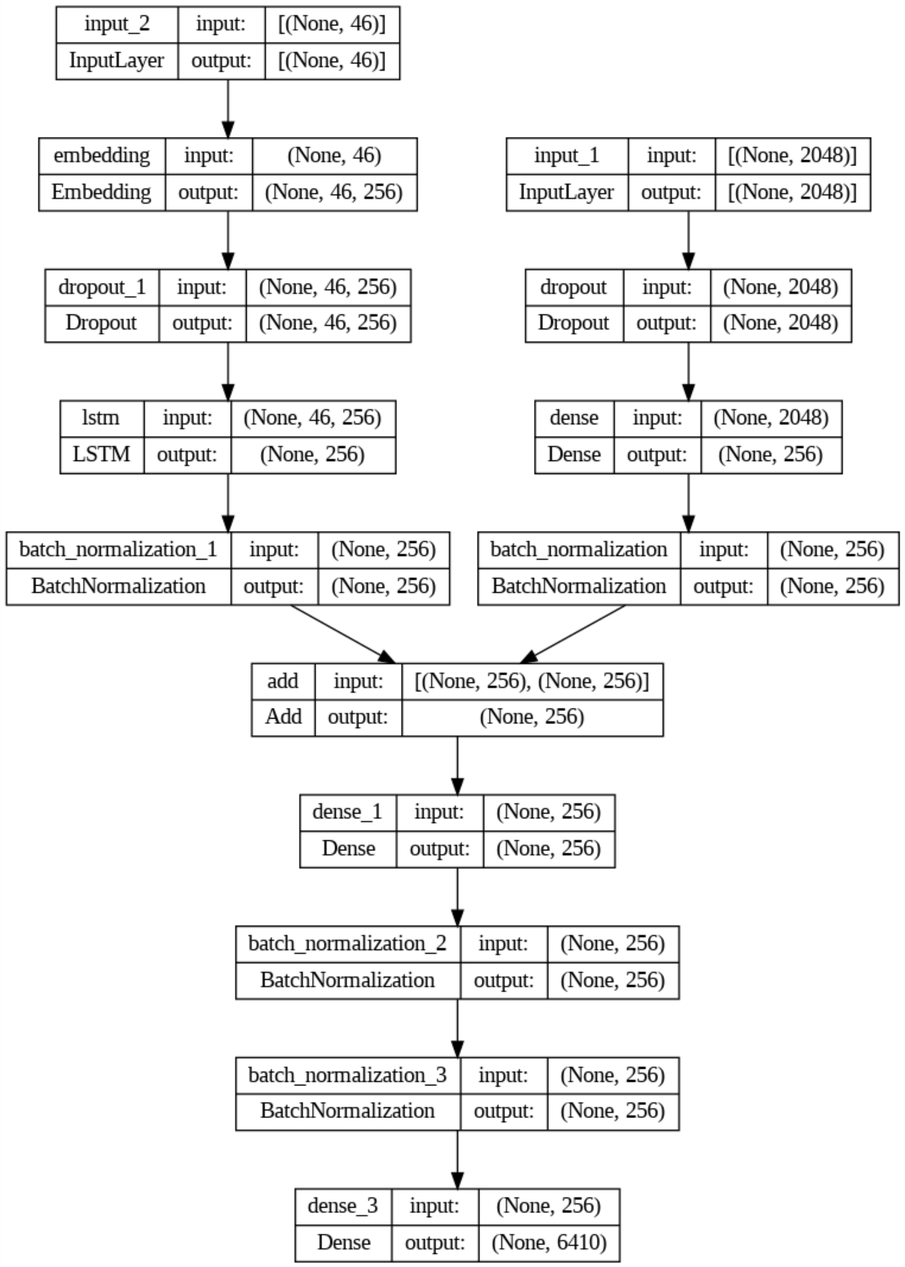

```
+--------------------+--------------------+
|          image_path|            features|
+--------------------+--------------------+
|http://images.coc...|[0.002998488, 0.0...|
|http://images.coc...|[0.0, 0.014579403...|
|http://images.coc...|[0.0, 0.19064964,...|
|http://images.coc...|[0.03379664, 0.02...|
|http://images.coc...|[0.2977563, 0.025...|
+--------------------+--------------------+
only showing top 5 rows
```

Implementation of Deep Learning Models

- **Xception model** is a deep CNN pre-trained on the ImageNet dataset with over 1 million images and 1000 classes.
- used as **CNN** to extract **features** from input images.
- **LSTM network** was responsible for **generating textual description** of the image based on features extracted by Xception model.
- Architecture is capable of handling both **spatial and temporal features** of an image.
- **Data Generator** was used to create **batches** of data to improve the speed of the process.
- The model was trained for **20 epochs** using Adam optimizer with a learning rate of **1e-3** and categorical cross-entropy loss function.



Model Architecture



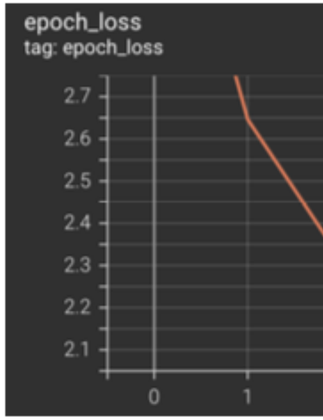
Results

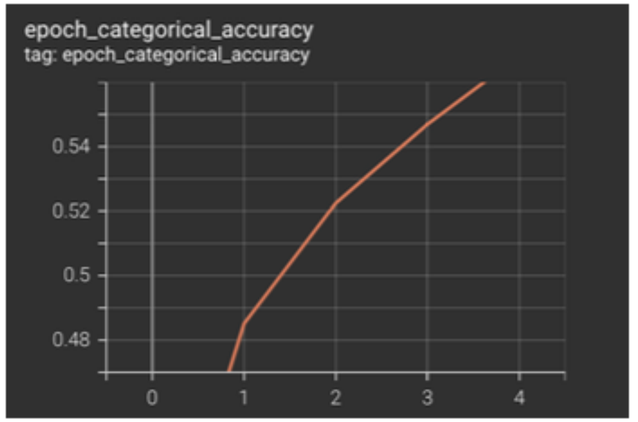
TensorBoard, a web-based visualization tool - **monitor** and **visualize** various metrics during the training process of a deep learning model

used to plot the **loss and accuracy** of our model over epochs, **performance**

loss metric in TensorBoard - overfitting or underfitting the training data, and adjust the hyperparameters accordingly - **1.213**

categorical accuracy metric - the percentage of correctly classified samples out of the total number of samples - **56%**





```
start a large passenger jet sitting on top of an airport tarmac end

<matplotlib.image.AxesImage at 0x7f75881abe90>
```


```
start a man is riding a bike on a sidewalk end

<matplotlib.image.AxesImage at 0x7f75368fed10>
```

BLEU Scores

degree of similarity between the predicted and actual captions is measured by the BLEU score, a popular assessment statistic for automatic image caption generation

using the NLTK library, it **compares** the **predicted** caption with the **actual** captions for that image to determine the BLEU score

results of the BLEU-1 and BLEU-2 tests are computed **independently**

BLEU-1(0.526724)

BLEU-2(0.352768)

higher scores indicating **better** performance.

Implications of Results on the project

- successfully demonstrated the **effectiveness** of using both convolutional neural networks (**CNNs**) and recurrent neural networks (**RNNs)** in a hybrid model, but also helped to capture the contextual information of the image.

- **evaluation** of the model using standard metrics such as **BLEU** with high scores, showed that the generated captions were comparable to human-generated captions in terms of accuracy and quality. This indicates that the model is highly effective at learning the underlying patterns in the data and generating accurate captions.

- the project also highlighted the **importance** of data preprocessing and feature extraction in deep learning models, and provided insights into best practices for model training and evaluation. This is important for improving the accuracy and generalizability of the model across different datasets and use cases.

- The project's findings also contribute to the development of more advanced natural language processing techniques, which can have broad applications in industries such as finance, law, and education.

Conclusion

- Image caption generator using deep learning techniques.
- The Xception convolutional neural network (CNN) was used as a feature extractor to obtain image features.
- The Long Short-Term Memory (LSTM) network was used as a language model to generate captions for images.
- The model was trained on the Microsoft Common Objects in Context (COCO) dataset.
- The performance of the model was evaluated using standard evaluation metrics such as BLEU, which was a high score.

Limitations

- Only explored the Faster CNN-LSTM architecture with the ResNet50 backbone network, This could potentially limit the scope and generalizability of the project
- Limited modules in PySpark, as well as the unavailability of the CNN-LSTM model in PySpark. As a result, the project had to rely on the Keras library to implement the model.
- Free trial period may be insufficient for long- term projects.
- The available resources in the free trial are limited, which may result in resource constraint issues when running large-scale deep learning models and incorporating large datasets may require to run the training in batches
- Limited language support

Future Scope

- Examine **different frameworks** or platforms with full subscription, such Google Cloud AI Platform or Amazon SageMaker, that are better suited for scaling object identification models.

- Investigate the impact of additional data augmentation techniques on model performance, such as **color jittering** or **Gaussian noise**

- Explore the potential of using **ensemble methods** for object detection, which involve combining the outputs of multiple models to improve accuracy.

Contributions and Impact

- E-commerce, social media, healthcare, and accessibility. For example, the image caption generator can assist visually impaired individuals in accessing online and offline environments, as well as enhance content curation and recommendation in e-commerce and social media platforms.

- Development of more advanced natural language processing techniques, which can have broad applications in industries such as finance, law, and education.

The End