

EVOASTRA Major Project

...

- **Team:** CP33 – Team B
- **Topic :** Image Caption Generation Using Deep Learning (CNN–LSTM with Attention)
- **Tools Used :** Python, Tensorflow, Keras, InceptionV3 (ImageNet Pretrained)
- **Mentor :** Aniket Manwatkar
- **Team Members :**
 1. Mohammad Sharjeel Yazdani [Team Lead]
 2. Samradnyi Kale [Co-Lead]
 3. Rishabh Tanpure [Co-Lead]
 4. Sakshi Giglani
 5. Sriya Sahu
 6. Monu Kumar Jha

Project Overview

- Image Captioning is an AI task that combines Computer Vision (CV) and Natural Language Processing (NLP).
- The goal is to automatically generate meaningful, human-like captions for input images.
- Deep learning models interpret image content and translate it into natural language.
- Developed a deep learning system capable of generating descriptive captions for images.
- **The model integrates:**
 - CNN Encoder (InceptionV3) for extracting visual features
 - LSTM Decoder + Attention Mechanism for generating natural language caption
- Dataset Used: MS COCO (large-scale dataset with images + labeled captions)

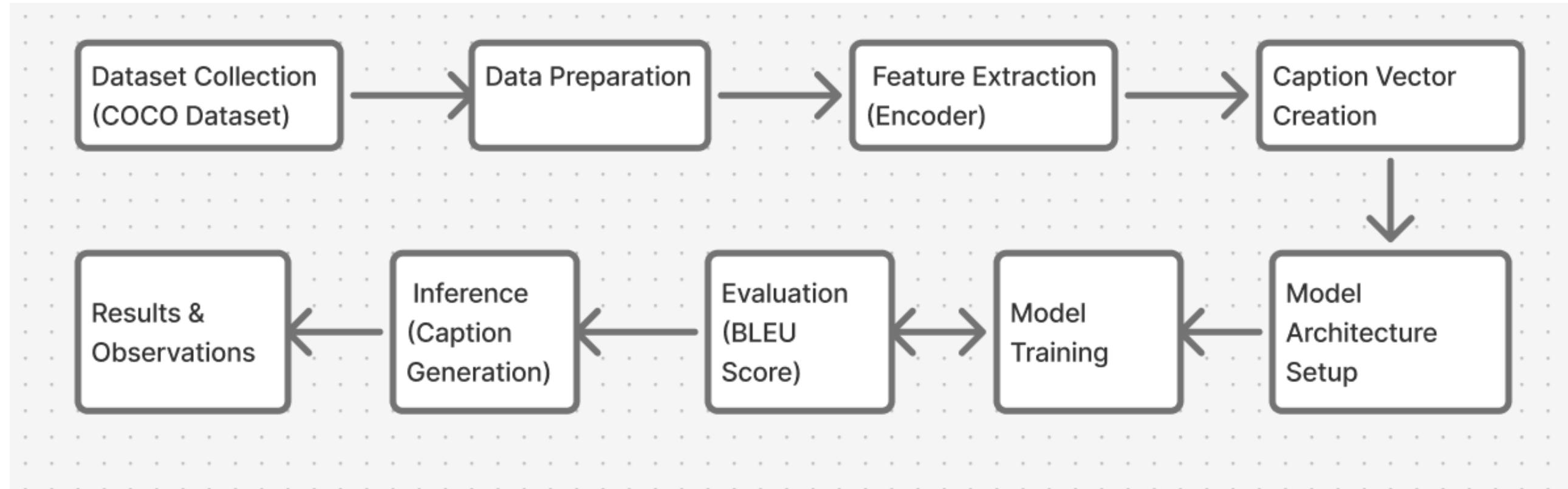
Project Objectives

- Build an AI model capable of generating human-like, context-aware captions.
- Utilize a pre-trained CNN (InceptionV3) to extract semantic features from images.
- Design and implement an Encoder–Decoder architecture with Attention.
- Evaluate model performance using BLEU score metrics.
- Provide a hands-on understanding of integrating Computer Vision + NLP techniques in a real-world application.
- **Workflow includes:**
Data Preparation → Feature Extraction → Caption Processing → Model Training → Evaluation → Fine-tuning

Implementation Workflow



K Keras



Projects Requirements

1. Data Preparation

- Access to MS COCO dataset (images + captions)
- Tools for image loading & resizing

2. Feature Extraction

3. Caption Processing

- word → index
- index → word

4. Model Training

5. Model Evaluation

- BLEU score computation script

6. Fine-Tuning Requirements

1. Hardware Requirements :

- GPU-enabled system (recommended: NVIDIA GPU)
- Minimum 8 GB RAM (16 GB preferred)
- 20–30 GB storage for dataset + model files

2. Software Requirements :

- Python 3.8+
- TensorFlow / Keras
- NLTK / Tokenizers
- VS Code

Data Preparation: MS COCO

...

- Large dataset with 80k+ training images and 5 captions per image.
- Includes objects, people, scenes, daily life activities.
- Datasets link: [2017 Train images \[118K/18GB\]](#)
[2017 Train/Val annotations \[241MB\]](#)

Image Preprocessing :

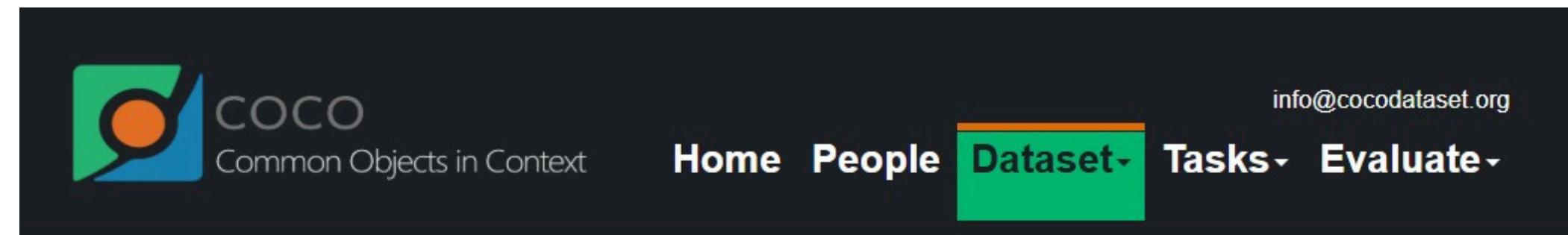
- Resize all input images to 299×299 pixels (compatible with InceptionV3)
- Normalize pixel values to the 0–1 range for stable training
- Convert images into arrays suitable for deep learning models

Caption Preprocessing:

- Tokenization using Keras Tokenizer
- Padding for uniform length
- Add <start> and <end> tokens
- Remove punctuation, convert to lowercase, and clean unnecessary characters

Data Source

...



Tools

COCO API

Images

[2014 Train images \[83K/13GB\]](#)
[2014 Val images \[41K/6GB\]](#)
[2014 Test images \[41K/6GB\]](#)
[2015 Test images \[81K/12GB\]](#)
[2017 Train images \[118K/18GB\]](#)
[2017 Val images \[5K/1GB\]](#)
[2017 Test images \[41K/6GB\]](#)
[2017 Unlabeled images \[123K/19GB\]](#)

Annotations

[2014 Train/Val annotations \[241MB\]](#)
[2014 Testing Image info \[1MB\]](#)
[2015 Testing Image info \[2MB\]](#)
[2017 Train/Val annotations \[241MB\]](#)
[2017 Stuff Train/Val annotations \[1.1GB\]](#)
[2017 Panoptic Train/Val annotations \[821MB\]](#)
[2017 Testing Image info \[1MB\]](#)
[2017 Unlabeled Image info \[4MB\]](#)

1. Overview

Which dataset splits should you download? Each year's images are associated with different tasks. Specifically:

2014 Train/Val
2014 Testing
2015 Testing

2017 Train/Val/Test

2017 Unlabeled

Detection 2015, Captioning 2015, Detection 2016, Keypoints 2016,
DensePose 2020
Captioning 2015
Detection 2015, Detection 2016, Keypoints 2016
Detection 2017, Keypoints 2017, Stuff 2017,
Detection 2018, Keypoints 2018, Stuff 2018, Panoptic 2018,
Detection 2019, Keypoints 2019, Stuff 2019, Panoptic 2019,
Detection 2020, Keypoints 2020, Panoptic 2020
[optional data for any competition]

Output: Saving the Extracted Features



The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** Major_Project_Caption_Generation118k.ipynb X, Untitled-1 9+, Major_Project_Caption_Generation_5k.ipynb
- Toolbar:** Generate, + Code, + Markdown, Run All, Restart, Clear All Outputs, Jupyter Variables, Outline, Python 3.12.6
- Section Header:** Cell 5 — Extract and save image features (run once)
- Description:** What this does: uses pre-trained InceptionV3 (without the final classification head) to convert each image into a fixed-length feature vector and saves it to disk. Run this cell once — it may take time depending on number of images.
- Code:**

```
# Cell 5 - extract pooled features and save as .npy files
base_model = InceptionV3(include_top=False, weights='imagenet', pooling='avg') # pooling='avg' -> one vector per image
base_model.trainable = False
print("Feature dimension:", base_model.output_shape[1])

for fname in tqdm(img2caps.keys(), desc="Extracting features"):
    image_id = os.path.splitext(fname)[0]
    out_path = os.path.join(FEATURES_DIR, image_id + '.npy')
    if os.path.exists(out_path):
        continue
    img_path = os.path.join(IMAGES_DIR, fname)
    if not os.path.exists(img_path):
        # skip if image file missing
        continue
    arr = load_and_preprocess_image(img_path)
    arr = np.expand_dims(arr, 0)
    feat = base_model.predict(arr, verbose=0)[0] # shape (feature_dim,)
    np.save(out_path, feat)
print("Feature extraction done. Saved in", FEATURES_DIR)
```
- Cell Status:** [57] ✓ 8.5s Python
- Output:** ... Feature dimension: 2048
Extracting features: 100%|██████████| 118287/118287 [00:06<00:00, 17266.89it/s]
Feature extraction done. Saved in ./features

Output: Cleaned, structured data ready for modeling



The screenshot shows a Jupyter Notebook interface with a single code cell. The code is written in Python and prints feature vectors for three randomly selected images. The output shows the image ID, feature vector shape, and the first 10 feature values for each image.

```
# === Cell 5.1 extracted feature vectors ===
import numpy as np
import random
import os

# Pick some random image IDs from train_ids
sample_ids = random.sample(train_ids, 3)

for img_id in sample_ids:
    feat_path = os.path.join(FEATURES_DIR, img_id + '.npy')
    if os.path.exists(feat_path):
        feat = np.load(feat_path)
        print(f"\nImage ID: {img_id}")
        print("Feature vector shape:", feat.shape)
        print("First 10 feature values:", feat[:10])
    else:
        print(f"Feature file missing for {img_id}")

[58]  ✓  0.0s  Python
```

...

```
Image ID: 000000315805
Feature vector shape: (2048,)
First 10 feature values: [0.1403391  0.58434105  0.26905352  0.3974986  0.625443   0.21929707
 1.0755721  0.31673926  0.01581189  0.31125978]

Image ID: 000000162358
Feature vector shape: (2048,)
First 10 feature values: [0.31865853  0.16279033  0.92625237  0.390826   0.48552263  0.2658648
 0.19599834  0.16347247  0.02084418  0.39156142]

Image ID: 000000304684
Feature vector shape: (2048,)
First 10 feature values: [0.20918494  0.15040675  0.2227888  0.02145237  0.0744825  0.64717346
 0.6128953  0.17526665  0.4107612  0.4435931 ]
```

x ⊗ 2 △ 19 Spaces: 4 { } Cell 12 of 32

Model Architecture : Encoder–Decoder with Attention

Encoder

- Uses InceptionV3 CNN to extract deep visual features from each image
- Converts the image into a 2048-dimensional feature embedding
- Provides high-level semantic representation for the decoder

Decoder (LSTM)

- An LSTM-based language generator produces the caption word-by-word
- Takes encoded image features + previous word as inputs
- Learns language patterns and sentence structure

Attention Mechanism

- Dynamically focuses on relevant regions of the image for each generated word
 - Allows the model to “look” at different image parts during captioning
 - Enhances context awareness and improves caption accuracy
-

Model Training

Training Strategy

- Model trained using the Adam Optimizer for stable and efficient convergence
- Dataset divided into 80% training and 20% validation to evaluate performance during training

Key Hyperparameters

- **Batch Size:** Controls number of samples processed per step
- **Learning Rate:** Determines how quickly model weights are updated
- **LSTM Units:** Defines the capacity of the decoder's language model

Monitoring During Training

- **Training & Validation Loss:** Indicates model learning progress
- **Accuracy Metrics:** Tracks how well predicted tokens match actual caption tokens
- **Caption Quality:** Evaluated through periodic sample predictions and BLEU score assessment

Epoch
screenshots



File Edit Selection View Go Run Terminal Help ← → Search

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===

Generate + Code + Markdown | ⚡ Interrupt ⚡ Restart Clear All Outputs Go To Jupyter Variables Outline ... Python 3.12.6

300m 46.4s

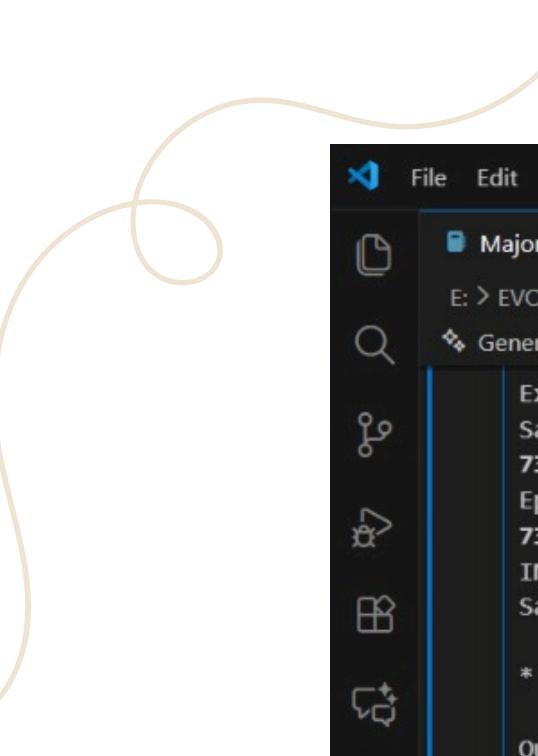
```
... FEATURE_DIM detected: 2048
Feature batch shape: (64, 2048)
Input seq batch shape: (64, 30)
Target batch shape: (64, 30)

Found checkpoint: saved_models_tf\epoch_5.keras (saved epoch 5)
Loaded checkpoint with compile=True (optimizer state restored).
Resuming training from the next epoch: epoch 6

===== DATASET SUMMARY =====
Images in training split: 94629
Captions per image: 5
Total caption samples: 473145
Batch size: 64
→ Steps per epoch: 7393
=====

Epoch 6/20
7393/7393 0s 514ms/step - loss: 2.8035INFO:tensorflow:Assets written to: saved_models_tf\epoch_6\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_6\assets
Saved artifact at 'saved_models_tf\epoch_6'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_6
Saved .keras checkpoint: saved_models_tf\epoch_6.keras
7393/7393 4457s 602ms/step - loss: 2.8035 - val_loss: 2.6241
Epoch 7/20
```



```
File Edit Selection View Go Run Terminal Help ⏴ ⏵ Search EVOASTRA INTERNSHIP > Major_Project [ Caption Generation ]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb # === CELL 10 (FINAL — with robust validation repeat) === Generate + Code + Markdown ⏷ Interrupt ⏹ Restart ⏸ Clear All Outputs ⏵ Go To Jupyter Variables ⏸ Outline Python 3.12.6 Exported TF artifact: saved_models_tf\epoch_6 Saved .keras checkpoint: saved_models_tf\epoch_6.keras 7393/7393 ━━━━━━ 4457s 602ms/step - loss: 2.8035 - val_loss: 2.6241 Epoch 7/20 7393/7393 ━━━━━━ 0s 435ms/step - loss: 2.7531INFO:tensorflow:Assets written to: saved_models_tf\epoch_7\assets INFO:tensorflow:Assets written to: saved_models_tf\epoch_7\assets Saved artifact at 'saved_models_tf\epoch_7'. The following endpoints are available:  
* Endpoint 'serve'  
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]  
Output Type:  
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)  
Captures:  
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)  
Exported TF artifact: saved_models_tf\epoch_7 Saved .keras checkpoint: saved_models_tf\epoch_7.keras 7393/7393 ━━━━━━ 3914s 529ms/step - loss: 2.7531 - val_loss: 2.5875 Epoch 8/20 7393/7393 ━━━━━━ 0s 478ms/step - loss: 2.7112INFO:tensorflow:Assets written to: saved_models_tf\epoch_8\assets INFO:tensorflow:Assets written to: saved_models_tf\epoch_8\assets Saved artifact at 'saved_models_tf\epoch_8'. The following endpoints are available:  
* Endpoint 'serve'  
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]  
Output Type:  
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)  
Captures:  
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)  
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

Ln 48, Col 28 Spaces: 4 CRLF Cell 26 of 38

File Edit Selection View Go Run Terminal Help ⏪ ⏩ Search ⏷ ⏵ ⏶ ⏸ ⏹ ⏵ ⏴ ⏶ ⏵ ⏵ ⏵

Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===

Generate + Code + Markdown | Interrupt ⚡ Restart ⏺ Clear All Outputs ⏺ Go To ⏺ Jupyter Variables ⏺ Outline ⏺ Python 3.12.6

```
Exported TF artifact: saved_models_tf\epoch_7
Saved .keras checkpoint: saved_models_tf\epoch_7.keras
7393/7393 3914s 529ms/step - loss: 2.7531 - val_loss: 2.5875
Epoch 8/20
7393/7393 0s 478ms/step - loss: 2.7112INFO:tensorflow:Assets written to: saved_models_tf\epoch_8\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_8\assets
Saved artifact at 'saved_models_tf\epoch_8'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_8
Saved .keras checkpoint: saved_models_tf\epoch_8.keras
7393/7393 4386s 593ms/step - loss: 2.7112 - val_loss: 2.5598
Epoch 9/20
7393/7393 0s 490ms/step - loss: 2.6770INFO:tensorflow:Assets written to: saved_models_tf\epoch_9\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_9\assets
Saved artifact at 'saved_models_tf\epoch_9'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

x ⑧ 0 △ 11 Ln 48, Col 28 Spaces: 4 Spaces: 4 CRLF ⏺ Cell 26 of 38

File Edit Selection View Go Run Terminal Help ← → Search

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===

Generate + Code + Markdown | ⚡ Interrupt ⚡ Restart ⚡ Clear All Outputs ⚡ Go To ⚡ Jupyter Variables ⚡ Outline ... Python 3.12.6

```
* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_8
Saved .keras checkpoint: saved_models_tf\epoch_8.keras
7393/7393 4386s 593ms/step - loss: 2.7112 - val_loss: 2.5598
Epoch 9/20
7393/7393 0s 490ms/step - loss: 2.6770INFO:tensorflow:Assets written to: saved_models_tf\epoch_9\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_9\assets
Saved artifact at 'saved_models_tf\epoch_9'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_9
Saved .keras checkpoint: saved_models_tf\epoch_9.keras
7393/7393 4047s 547ms/step - loss: 2.6770 - val_loss: 2.5369
Epoch 10/20
4169/7393 17:49 332ms/step - loss: 2.6498
```

```

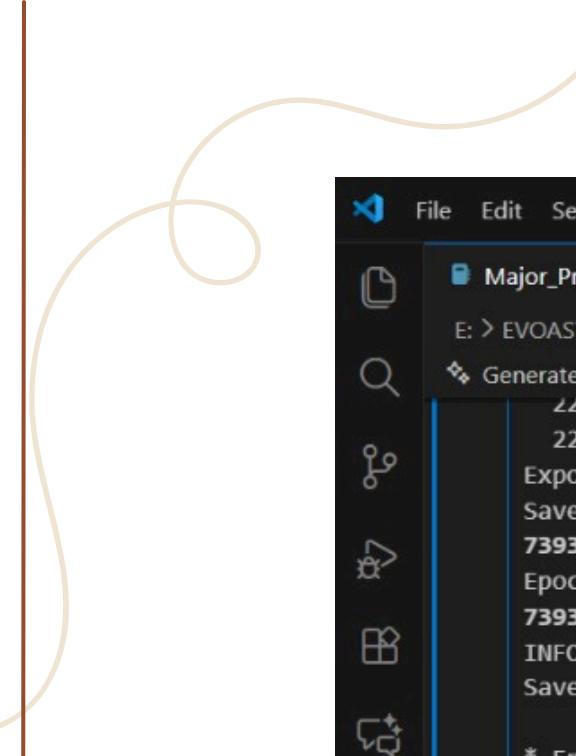
File Edit Selection View Go Run Terminal Help ⏪ ⏫ Search ⏷ ⏸ ⏹ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ Python 3.12.6
Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb
E: > EVOASTRA INTERNSHIP > Major_Project [ Caption Generation ]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===
Generate + Code + Markdown | ⏹ Interrupt ⏹ Restart ⏹ Clear All Outputs ⏹ Go To ⏹ Jupyter Variables ⏹ Outline ...
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_9
Saved .keras checkpoint: saved_models_tf\epoch_9.keras
7393/7393 ━━━━━━━━━━ 4047s 547ms/step - loss: 2.6770 - val_loss: 2.5369
Epoch 10/20
7393/7393 ━━━━━━━━ 0s 389ms/step - loss: 2.6489INFO:tensorflow:Assets written to: saved_models_tf\epoch_10\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_10\assets
Saved artifact at 'saved_models_tf\epoch_10'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_10
Saved .keras checkpoint: saved_models_tf\epoch_10.keras
7393/7393 ━━━━━━━━ 3444s 466ms/step - loss: 2.6489 - val_loss: 2.5199
Epoch 11/20
7393/7393 ━━━━━━━━ 0s 376ms/step - loss: 2.6236INFO:tensorflow:Assets written to: saved_models_tf\epoch_11\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_11\assets
Saved artifact at 'saved_models_tf\epoch_11'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)

```

Spaces: 4 CRLF ⏹ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ ⏸ Cell 26 of 38



```
File Edit Selection View Go Run Terminal Help ⏪ ⏫ Search ⏴ ⏵ ⏷ ⏸ ⏹ ⏺ ⏻ ⏻ ⏻ ⏻ ⏻ ⏻ ⏻ Python 3.12.6
Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb
E: > EVOASTRA INTERNSHIP > Major_Project [ Caption Generation ]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # == CELL 10 (FINAL — with robust validation repeat) ===
Generate + Code + Markdown | ⚡ Interrupt ⚡ Restart ⚡ Clear All Outputs ⚡ Go To ⚡ Jupyter Variables ⚡ Outline ...
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_10
Saved .keras checkpoint: saved_models_tf\epoch_10.keras
7393/7393 3444s 466ms/step - loss: 2.6489 - val_loss: 2.5199
Epoch 11/20
7393/7393 0s 376ms/step - loss: 2.6236INFO:tensorflow:Assets written to: saved_models_tf\epoch_11\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_11\assets
Saved artifact at 'saved_models_tf\epoch_11'. The following endpoints are available:
* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_11
Saved .keras checkpoint: saved_models_tf\epoch_11.keras
7393/7393 3390s 459ms/step - loss: 2.6236 - val_loss: 2.5019
Epoch 12/20
7393/7393 0s 374ms/step - loss: 2.6019INFO:tensorflow:Assets written to: saved_models_tf\epoch_12\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_12\assets
Saved artifact at 'saved_models_tf\epoch_12'. The following endpoints are available:
* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

Spaces: 4 CRLF { } Cell 26 of 38

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===

Generate + Code + Markdown | ⚡ Interrupt ⚡ Restart ⚡ Clear All Outputs ⚡ Go To | Jupyter Variables | Outline ...

Python 3.12.6

```
* Endpoint 'serve'
args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_11
Saved .keras checkpoint: saved_models_tf\epoch_11.keras
7393/7393 ━━━━━━━━ 3390s 459ms/step - loss: 2.6236 - val_loss: 2.5019
Epoch 12/20
7393/7393 ━━━━━━━━ 0s 374ms/step - loss: 2.6019INFO:tensorflow:Assets written to: saved_models_tf\epoch_12\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_12\assets
Saved artifact at 'saved_models_tf\epoch_12'. The following endpoints are available:

* Endpoint 'serve'
args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_12
Saved .keras checkpoint: saved_models_tf\epoch_12.keras
7393/7393 ━━━━━━━━ 3372s 456ms/step - loss: 2.6019 - val_loss: 2.4874
Epoch 13/20
5265/7393 ━━━━━━━━ 13:44 387ms/step - loss: 2.5820
```

Spaces: 4 CRLF { Cell 26 of 38

File Edit Selection View Go Run Terminal Help ← → Search

Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # Cell 14 - save tokenizer

Generate + Code + Markdown | Run All Restart Clear All Outputs Jupyter Variables Outline ... Python 3.12.6

```
2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_12
Saved .keras checkpoint: saved_models_tf\epoch_12.keras
7393/7393 3372s 456ms/step - loss: 2.6019 - val_loss: 2.4874
Epoch 13/20
7393/7393 0s 446ms/step - loss: 2.5819INFO:tensorflow:Assets written to: saved_models_tf\epoch_13\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_13\assets
Saved artifact at 'saved_models_tf\epoch_13'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_13
Saved .keras checkpoint: saved_models_tf\epoch_13.keras
7393/7393 4351s 589ms/step - loss: 2.5819 - val_loss: 2.4744
Epoch 14/20
7393/7393 0s 421ms/step - loss: 2.5642INFO:tensorflow:Assets written to: saved_models_tf\epoch_14\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_14\assets
Saved artifact at 'saved_models_tf\epoch_14'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

Ln 120, Col 2 (4921 selected) Spaces: 4 Spaces: 4 CRLF Cell 38 of 38

File Edit Selection View Go Run Terminal Help ⏪ ⏫ Search ⏮ ⏱ ⏳ ⏰ ⏰ ⏰ ⏰ ⏰ ⏰ ⏰ ⏰

Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # Cell 14 - save tokenizer

Generate + Code + Markdown | Run All ⏵ Restart ⏵ Clear All Outputs | Jupyter Variables ⏵ Outline ... Python 3.12.6

```
Exported TF artifact: saved_models_tf\epoch_13
Saved .keras checkpoint: saved_models_tf\epoch_13.keras
7393/7393 4351s 589ms/step - loss: 2.5819 - val_loss: 2.4744
Epoch 14/20
7393/7393 0s 421ms/step - loss: 2.5642INFO:tensorflow:Assets written to: saved_models_tf\epoch_14\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_14\assets
Saved artifact at 'saved_models_tf\epoch_14'. The following endpoints are available:

* Endpoint 'serve'
args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_14
Saved .keras checkpoint: saved_models_tf\epoch_14.keras
7393/7393 3765s 509ms/step - loss: 2.5642 - val_loss: 2.4660
Epoch 15/20
7393/7393 0s 397ms/step - loss: 2.5490INFO:tensorflow:Assets written to: saved_models_tf\epoch_15\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_15\assets
Saved artifact at 'saved_models_tf\epoch_15'. The following endpoints are available:

* Endpoint 'serve'
args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

✖️ ⏷ 0 ▲ 11 Ln 120, Col 2 (4921 selected) Spaces: 4 Spaces: 4 CRLF ⏷ Cell 38 of 38 ⏷

File Edit Selection View Go Run Terminal Help ⏪ ⏩ Search ⏴ ⏵

Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # Cell 14 - save tokenizer

Generate + Code + Markdown | Run All Restart Clear All Outputs Jupyter Variables Outline ... Python 3.12.6

```
Exported TF artifact: saved_models_tf\epoch_14
Saved .keras checkpoint: saved_models_tf\epoch_14.keras
7393/7393 3765s 509ms/step - loss: 2.5642 - val_loss: 2.4660
Epoch 15/20
7393/7393 0s 397ms/step - loss: 2.5490INFO:tensorflow:Assets written to: saved_models_tf\epoch_15\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_15\assets
Saved artifact at 'saved_models_tf\epoch_15'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_15
Saved .keras checkpoint: saved_models_tf\epoch_15.keras
7393/7393 3575s 484ms/step - loss: 2.5490 - val_loss: 2.4561
Epoch 16/20
7393/7393 0s 372ms/step - loss: 2.5333INFO:tensorflow:Assets written to: saved_models_tf\epoch_16\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_16\assets
Saved artifact at 'saved_models_tf\epoch_16'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

Ln 120, Col 2 (4921 selected) Spaces: 4 Spaces: 4 CRLF Cell 38 of 38

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ Search Q Search
Major_Project_Caption_Generation118k.ipynb × Major_Project_Caption_Generation_5k.ipynb
E: > EVOASTRA INTERNSHIP > Major_Project [ Caption Generation ]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===
Generate + Code + Markdown | Run All ⌍ Restart ⌍ Clear All Outputs | Jupyter Variables ⌍ Outline ...
Python 3.12.6
Exported TF artifact: saved_models_tf\epoch_16
Saved .keras checkpoint: saved_models_tf\epoch_16.keras
7393/7393 ━━━━━━ 3396s 459ms/step - loss: 2.5333 - val_loss: 2.4461
Epoch 17/20
7393/7393 ━━━━ 0s 374ms/step - loss: 2.5202INFO:tensorflow:Assets written to: saved_models_tf\epoch_17\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_17\assets
Saved artifact at 'saved_models_tf\epoch_17'. The following endpoints are available:
* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_17
Saved .keras checkpoint: saved_models_tf\epoch_17.keras
7393/7393 ━━━━━━ 3409s 461ms/step - loss: 2.5202 - val_loss: 2.4395
Epoch 18/20
7393/7393 ━━━━ 0s 351ms/step - loss: 2.5079INFO:tensorflow:Assets written to: saved_models_tf\epoch_18\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_18\assets
Saved artifact at 'saved_models_tf\epoch_18'. The following endpoints are available:
* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
Ln 120, Col 2 (4921 selected) Spaces: 4 CRLF ⌋ Cell 26 of 38
```

File Edit Selection View Go Run Terminal Help ← → Search

Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # == CELL 10 (FINAL — with robust validation repeat) ==

Generate + Code + Markdown | Run All Restart Clear All Outputs Jupyter Variables Outline ... Python 3.12.6

```
Saved .keras checkpoint: saved_models_tf\epoch_17.keras
7393/7393 3409s 461ms/step - loss: 2.5202 - val_loss: 2.4395
Epoch 18/20
7393/7393 0s 351ms/step - loss: 2.5079INFO:tensorflow:Assets written to: saved_models_tf\epoch_18\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_18\assets
Saved artifact at 'saved_models_tf\epoch_18'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_18
Saved .keras checkpoint: saved_models_tf\epoch_18.keras
7393/7393 3251s 440ms/step - loss: 2.5079 - val_loss: 2.4341
Epoch 19/20
7393/7393 0s 363ms/step - loss: 2.4964INFO:tensorflow:Assets written to: saved_models_tf\epoch_19\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_19\assets
Saved artifact at 'saved_models_tf\epoch_19'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

< × 0 △ 11 Ln 120, Col 2 (4921 selected) Spaces: 4 CRLF Cell 26 of 38

File Edit Selection View Go Run Terminal Help ← → Search

Major_Project_Caption_Generation118k.ipynb X Major_Project_Caption_Generation_5k.ipynb

E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb > # === CELL 10 (FINAL — with robust validation repeat) ===

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline ... Python 3.12.6

```

Saved artifact at 'saved_models_tf\epoch_19'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_19
Saved .keras checkpoint: saved_models_tf\epoch_19.keras
7393/7393 3339s 452ms/step - loss: 2.4964 - val_loss: 2.4267
Epoch 20/20
7393/7393 0s 353ms/step - loss: 2.4862INFO:tensorflow:Assets written to: saved_models_tf\epoch_20\assets
INFO:tensorflow:Assets written to: saved_models_tf\epoch_20\assets
Saved artifact at 'saved_models_tf\epoch_20'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): List[TensorSpec(shape=(None, 2048), dtype=tf.float32, name='image_features'), TensorSpec(shape=(None, 30), dtype=tf.float32, name='input_seq')]
Output Type:
  TensorSpec(shape=(None, 30, 5000), dtype=tf.float32, name=None)
Captures:
  2298205755280: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754896: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205755856: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298240760528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205754128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  2298205756240: TensorSpec(shape=(), dtype=tf.resource, name=None)
Exported TF artifact: saved_models_tf\epoch_20
Saved .keras checkpoint: saved_models_tf\epoch_20.keras
7393/7393 5907s 799ms/step - loss: 2.4862 - val_loss: 2.4224
Training finished and model saved to: ./caption_model.keras

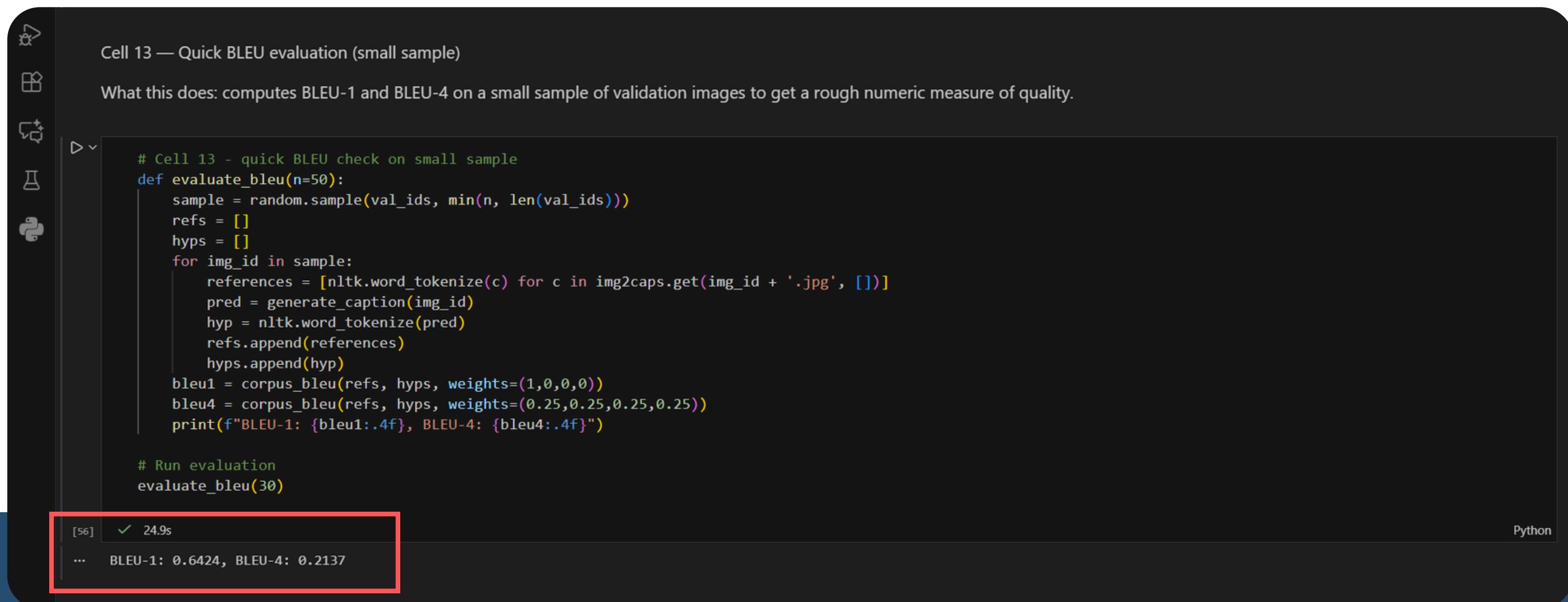
```

Ln 120, Col 2 (4921 selected) Spaces: 4 CRLF { } Cell 26 of 38

Model Evaluation

• • •

- BLEU Score used as the primary metric to measure similarity between predicted captions and ground-truth captions
- Evaluates n-gram overlap to assess caption fluency and accuracy
- Model tested on unseen images to verify generalization capability



Cell 13 — Quick BLEU evaluation (small sample)

What this does: computes BLEU-1 and BLEU-4 on a small sample of validation images to get a rough numeric measure of quality.

```
# Cell 13 - quick BLEU check on small sample
def evaluate_bleu(n=50):
    sample = random.sample(val_ids, min(n, len(val_ids)))
    refs = []
    hyps = []
    for img_id in sample:
        references = [nltk.word_tokenize(c) for c in img2caps.get(img_id + '.jpg', [])]
        pred = generate_caption(img_id)
        hyp = nltk.word_tokenize(pred)
        refs.append(references)
        hyps.append(hyp)
    bleu1 = corpus_bleu(refs, hyps, weights=(1,0,0,0))
    bleu4 = corpus_bleu(refs, hyps, weights=(0.25,0.25,0.25,0.25))
    print(f"BLEU-1: {bleu1:.4f}, BLEU-4: {bleu4:.4f}")

# Run evaluation
evaluate_bleu(30)
```

[56] ✓ 24.9s

... BLEU-1: 0.6424, BLEU-4: 0.2137

Python

Output : Caption generated

The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar displays three open notebooks: 'Major_Project_Caption_Generation118k.ipynb' (active), 'Untitled-1 9+', and 'Major_Project_Caption_Generation_5k.ipynb'. The status bar indicates the path: 'E: > EVOASTRA INTERNSHIP > Major_Project [Caption Generation]_16_11_2025 > Major_Project_Caption_Generation_118k > Major_Project_Caption_Generation118k.ipynb' and '# Cell 12 - fixed greedy inference (uses full history)'. The Python version is listed as 'Python 3.12.6'. The left sidebar contains icons for search, generate, code, markdown, run all, restart, clear outputs, jupyter variables, outline, and help. The main area shows Python code for generating captions from image IDs. A red box highlights the output cell, which displays the image ID, the generated caption ('a herd of sheep grazing on a lush green field'), and the reference captions. The output cell has a status of '[55] ✓ 1.2s'.

```
# we care about the last real timestep = len(in_tokens) - 1
step = len(in_tokens) - 1
token_probs = preds[0, step] # shape (vocab_size,)
next_id = int(np.argmax(token_probs)) # greedy

# 4) stopping conditions
if next_id == 0: # pad
    break
if end_id is not None and next_id == end_id:
    break

# 5) append token
in_tokens.append(next_id)

# 6) convert ids → words, skipping <start>/<end>/0
words = []
for idx in in_tokens[1:]: # skip <start>
    if idx == 0:
        break
    if end_id is not None and idx == end_id:
        break
    w = index_word.get(idx, '')
    if w in ('<start>', '<end>', ''):
        continue
    words.append(w)

return ' '.join(words)

# Quick test on a random validation image
if len(val_ids) > 0:
    test_img = random.choice(val_ids)
    print("Image ID:", test_img)
    print("Generated caption:", generate_caption(test_img))
    print("Reference captions:", img2caps.get(test_img + '.jpg', [])[0:3])
else:
    print("No validation images available for test.")

[55] ✓ 1.2s
```

...
Image ID: 000000061675
Generated caption: a herd of sheep grazing on a lush green field
Reference captions: ['some white and black sheep are on some grass hills and trees', 'some sheep standing together while surrounded by some tall grass', 'a herd of sheep standing on a

Challenges & Solutions



Challenge	Issue	Solution
Handling Large and Complex Datasets	MS COCO dataset is large and diverse, making preprocessing time-consuming	Implemented batch-wise preprocessing and used a GPU-accelerated pipeline for faster feature extraction
Generating Accurate Captions for Complex Scenes	Model struggles with cluttered backgrounds and multiple objects	Integrated an Attention Mechanism to help the model focus on the most relevant image regions
Limited Vocabulary for Rare Objects	Rare or unusual objects were poorly recognized due to limited training samples	Expanded vocabulary through tokenization strategies, and improved training with data augmentation
Avoiding Overfitting	Long training on a complex dataset can lead to overfitting	Used dropout layers, early stopping, and learning rate scheduling to improve generalization

Results & Observations

Key Results

- The model successfully generates relevant and context-aware captions for most images
- The Attention Mechanism significantly improves caption accuracy by focusing on important image regions
- Performance is stronger when trained on a diverse and well-balanced dataset, enabling the model to learn a wide range of objects and scenes

Observations:

- Captions are generally fluent, meaningful, and aligned with image content
- The encoder–decoder architecture effectively combines visual understanding with language generation

Limitations:

- Complex or cluttered scenes may result in shorter or less detailed captions
- Rare or unusual objects are more difficult for the model to identify due to limited representation in the training data

Conclusions

Project Summary

- Successfully developed an end-to-end Image Captioning System integrating Computer Vision and Natural Language Processing
- Implemented a complete workflow: data preparation, feature extraction, caption modeling, training, and evaluation

Key Learnings

- CNN-based Feature Extraction using pre-trained InceptionV3
- NLP Sequence Modeling for generating natural-language captions
- Encoder–Decoder Architecture with Attention for improved context understanding
- BLEU Score Evaluation to measure caption quality and model performance

THANK you