Image Caption Generation

Dataset

ArtEmis Dataset

ArtEmis (Art Emotions) is derived from the WikiArt collection of artworks, where each image is paired with human-written captions expressing emotions, interpretations, or descriptions of the artwork.

For this assignment, you will use a subset of the dataset for **image caption generation** — predicting a short, meaningful textual caption for each artwork.

- **Dataset Characteristics:** Multimodal (Image → Text)
- **Feature Type:** Pixel values (images), text (captions)
- **# Instances:** ~80,000 (subset 5–10k recommended as per your system permits)
- **Text Features:** Emotion-rich captions (short textual descriptions)
- **Target Variable:** Generated caption (sequence of words)

Objectives

- Understand and implement CNNs for image feature extraction without pre-training.
- Use LSTMs for sequence generation (caption generation).
- Explore and compare different text embedding strategies (TF-IDF and pre-trained embeddings).
- Combine CNN and LSTM architectures for a multimodal generation task.
- Use Transformer for image caption generation
- Practice clean experimental reporting, visualization, and interpretation.

Deliverables

Submit a **ZIP folder** containing:

1. **Jupyter Notebook (ArtEmis_Caption_Generation.ipynb)**
    - Contains all code, outputs, and visualizations.
    - Written as a clear report with headings and explanations along with loss curves.
2. **Trained Model Files** (.h5 or .pt)
3. **README File** with:
    - Setup and execution instructions
    - Dataset preprocessing steps
4. .py files used for training, pre-processing, evaluation.
5. Report

Tasks / Outline

1. Introduction

- Briefly describe the ArtEmis dataset and its focus on emotional art captions.
- Clearly define the goal: **Generate captions from image inputs** using a
    - CNN + LSTM model trained from scratch.
    - Transformer for image + text
- Mention that the task will compare text embeddings and sequence modeling variants.

---

2. Exploratory Data Analysis (EDA)

- Explore the dataset:
    - Number of samples, average caption length, vocabulary size

- Common words and bigrams in captions
- Visualize sample images with their ground-truth captions

- Identify caption diversity and patterns across artwork styles or emotions.

---

3. Preprocessing

Images

- Resize all images to a fixed shape (e.g., 128×128 or 224×224).
- Normalize pixel values to [0, 1].

Text

- Convert captions to lowercase, remove punctuation, tokenize.
- Build a vocabulary (limit vocabulary size to ~5,000–10,000).
- Add *start* (<start>) and *end* (<end>) tokens.
- Pad or truncate sequences to uniform length.

---

4. Text Representation

Implement and compare **three text embedding strategies**:

1. **TF-IDF Embeddings**
    - Compute TF-IDF vectors for words and use them as input to the LSTM.
    - Optionally reduce dimensionality (e.g., with PCA) to manage model size.
2. **Pre-Trained Word Embeddings**

    Choose **any two** from the following:

- **Word2Vec**
- **GloVe**
- **FastText**

Use them as **non-trainable embeddings** initialized from pre-trained weights.

For each embedding type:

- Report vocabulary size, embedding dimension, and token coverage.
- Compare generated captions qualitatively and quantitatively (BLEU, ROUGE, etc.).

---

5. Model 1 – CNN + LSTM Network

Architecture

- Use a custom CNN (from scratch) as the image encoder to extract features (avoid transfer learning) with multiple convolutional and pooling layers.
- Output a compact image feature vector (e.g., 256D).
- Build an LSTM or BiLSTM network to generate captions word by word.
- Inputs: Previous word embedding + image feature vector.
- Outputs: Next word in the sequence.
- Experiment with:
  - Different embedding dimensions
  - Hidden sizes (e.g., 128, 256)
  - Dropout layers

Training Objective

- Use **Categorical Cross-Entropy Loss** (mask padding tokens).

- Optimize using Adam or SGD optimizers.
- Track training and validation loss per epoch.

Evaluation

- Generate captions for test images.
- Compare generated vs. reference captions using metrics like BLEU, ROUGE, or CIDEr.
- Display a few sample images with their predicted captions.

---

6. Model 2 – Vision-Language Transformer

Architecture

- Replace both CNN and LSTM with Transformer-based components.
- Use:
    - A Vision Transformer (ViT) or custom Patch Transformer Encoder for images.
    - A Text Transformer Decoder for caption generation.
- The image is divided into patches, which are embedded and processed by a transformer encoder.
- The decoder attends to both previously generated tokens and image embeddings to produce captions.

Training Objective

- Use Categorical Cross-Entropy Loss to predict the next token in the sequence.
- Train end-to-end, allowing both visual and textual embeddings to be optimized together.

- Experiment with:
    - Sequence lengths
    - Number of transformer layers
    - Attention heads
    - Embedding dimensions

Evaluation

- Compare results with the CNN + LSTM baseline in terms of:
    - Caption fluency and relevance
    - BLEU, METEOR, or CIDEr scores
- Visualize attention maps to interpret how visual patches influence generated words.

---

6. Training and Evaluation

- Split the dataset into training, validation, and test sets.
- Use **categorical cross-entropy loss** and **Adam optimizer**.
- Apply **early stopping** to prevent overfitting.

Evaluation Metrics

- BLEU (Bilingual Evaluation Understudy)
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation)
- METEOR (optional)
- Qualitative evaluation: generated captions for sample images.

Comparisons

- TF-IDF vs. Word2Vec vs. GloVe/FastText embeddings
- LSTM vs. BiLSTM architectures
- Compare training loss curves, BLEU scores, and caption diversity between models.
- Discuss the effect of embedding choice on model performance.
- Reflect on how the Transformer handles long-range dependencies compared to LSTM.

---

7. Analysis & Discussion

- How does embedding choice affect caption fluency and accuracy?
- Which model setup generates the most diverse or coherent captions?
- Does the model learn emotional or stylistic cues from artworks?
- Discuss failure cases (e.g., repetitive or incomplete captions).

---

8. Conclusion

Summarize:

- Key insights from model performance
- Effect of embedding choice and architecture design
- Challenges in training from scratch
- Ideas for improving generation quality

---

9. References

- ArtEmis Dataset: Achlioptas et al., *CVPR 2021*
- Pre-trained embeddings: Word2Vec, GloVe, FastText documentation

- BLEU/ROUGE evaluation metrics references
- Any external tutorials or documentation used

Important Notes

- Libraries are allowed
- Code and explanations must be original — plagiarism or AI-generated work is not permitted.
- Use comments and markdowns to explain every decision.
- Visualize generated captions clearly.
- Proper citations are mandatory.

Evaluation

During evaluation, a **new subset of unseen artwork images** will be released.

Your task will be to:

- Generate captions for the new images using your trained model.
- Display top-3 generated captions per image.