Image Caption Generation on ArtEmis Dataset

Multimodal Generation Student Name: Roshni Pai, Vatsl Goswami
Date: December 4, 2025

1. Introduction

The goal of this project was to develop a multimodal deep learning model capable of generating emotion-rich captions for artwork using the **ArtEmis dataset**. Unlike standard image captioning tasks (e.g., COCO) which focus on factual descriptions (e.g., "a dog on a chair"), ArtEmis requires the model to capture abstract, stylistic, and emotional cues.

1.1 Objectives

- **From-Scratch Implementation:** Design and train a Convolutional Neural Network (CNN) for feature extraction without using pre-trained weights (e.g., ResNet/VGG).
- **Sequence Modeling:** Implement an LSTM-based decoder to generate captions.
- **Optimization:** Solve the "cold-start" problem where models trained from scratch fail to converge on small datasets (~10k images).

1.2 Hardware Benchmarking & Optimization

A significant portion of the development effort was dedicated to optimizing the training loop to fit within a 15-hour compute window. To determine the most efficient architecture, training was benchmarked across three distinct hardware environments:

Environment	Hardware	Batch Size	Time per Epoch	Verdict
Local Laptop 1	AMD Radeon T4 (Integrated)	8	~55 mins	**Failed:** Iteration too slow.
Google Colab	T4 GPU (16GB VRAM)	32	~12 mins	**Mixed:** Fast compute, but slow I/O reading 10k JPEGs from Drive.

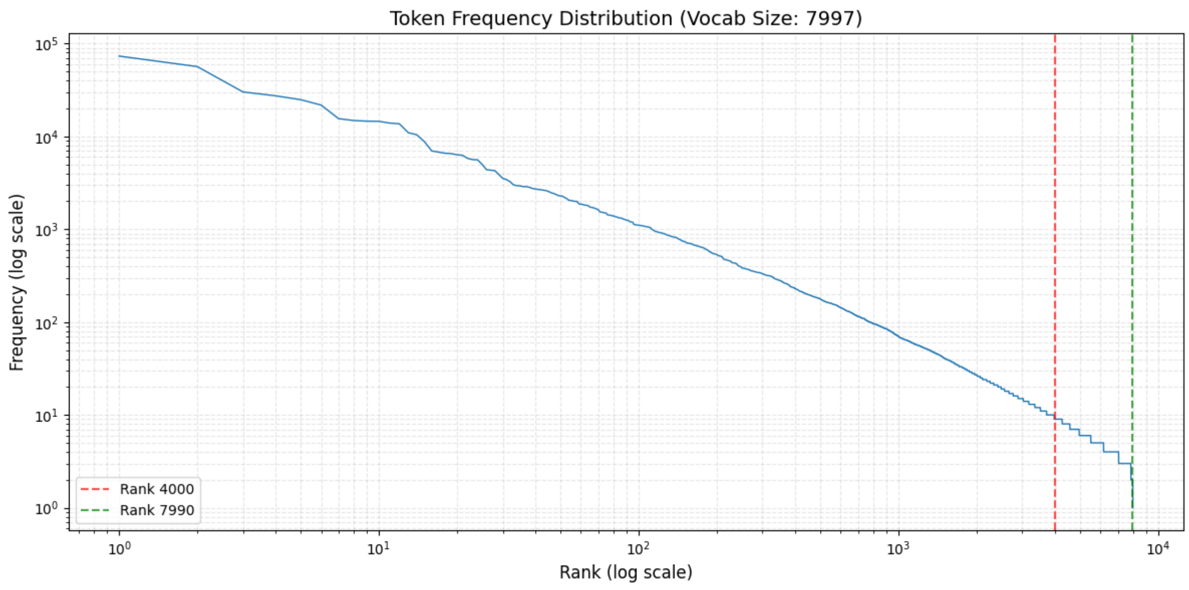
Local Laptop 2	**NVIDIA RTX 3050 (6GB)**	**64**	**~2 mins**	**Success:** Optimized with tensor pre-loading.

Optimization Strategy: To maximize the RTX 3050's throughput, we wrote a custom script (image_to_tensor_conversion.py) to resize and convert all 10,000 images into serialized PyTorch tensors (.pt files) offline. This eliminated the CPU decoding bottleneck, allowing the GPU to run at 95%+ utilization.

2. Exploratory Data Analysis (EDA)

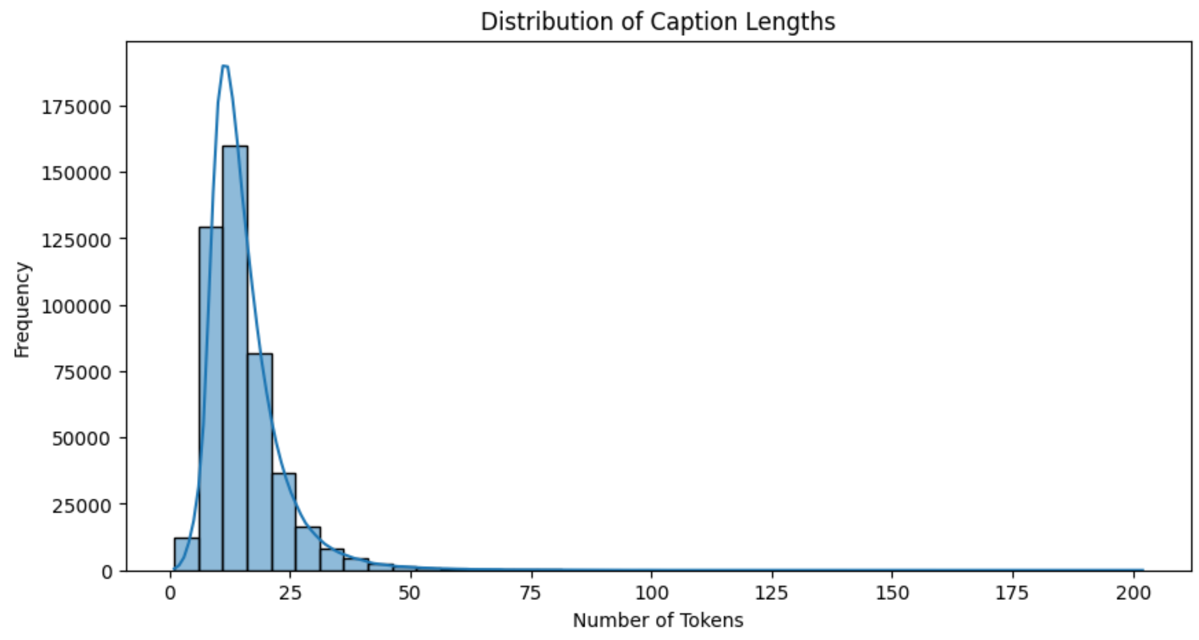
We analyzed the subset of 10,000 images to understand the linguistic and visual distribution.

2.1 Vocabulary & "Long Tail" Analysis



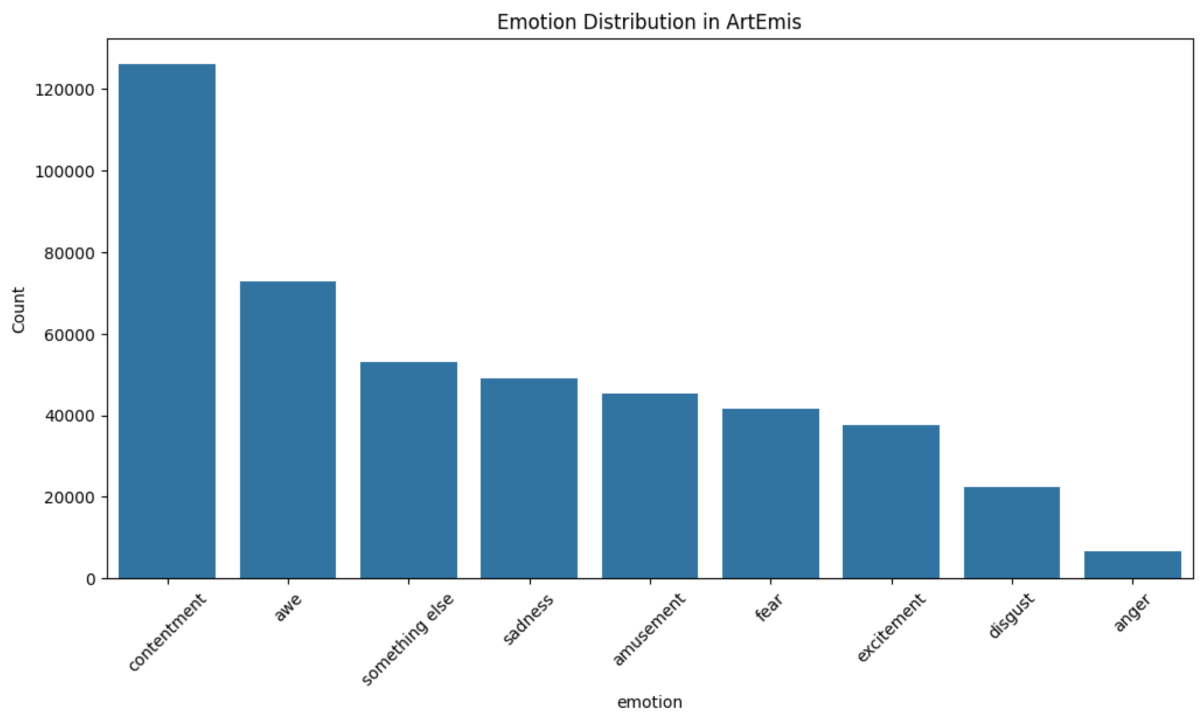
- **Total Vocabulary:** 7,997 unique tokens.
- **Frequency Analysis:** As shown in the frequency plot (see notebook), the data follows a steep Zipfian distribution.
    - The word *kindergarten* (Rank 4000) appears only 9 times.
    - The word *perimeter* (Rank 7990) appears only 2 times.
- **Decision:** We pruned the vocabulary to the top **3,000 words**. This removed the "noisy tail" (words appearing <5 times) which stabilizes the Softmax layer and prevents the model from wasting probability mass on rare tokens.

2.2 Caption Length Distribution



- **Mean Length:** 16 words.
- **Truncation:** Most captions fall well under 30 words. We set MAX_GEN_LEN = 30 to capture complete thoughts without excessive padding.

2.3 Emotion Distribution



The dataset is balanced heavily towards "Contentment" and "Awe", with "Fear" and "Disgust" being less frequent. This suggests the model will likely be biased towards generating positive descriptions unless explicitly conditioned (which we avoided in this iteration to focus on visual grounding).

3. Preprocessing Pipeline

3.1 Image Preprocessing: The "Normalization" Pivot

A critical finding during our experimentation was the impact of input normalization.

- **Initial Approach:** We initially used standard ImageNet normalization (mean=[0.485...], std=[0.229...]).
- **The Issue:** Since our CNN was initialized with random weights, forcing art images into a distribution derived from natural photographs—and feeding them to a randomly initialized network—caused feature distortion. The model failed to learn.
- **The Solution:** We switched to simple [0, 1] scaling. This aligned the input data distribution with the mathematical assumptions of Xavier/Kaiming weight initialization.

4. Text Representation & Embeddings

We explored three embedding strategies to represent the textual input:

1. **GloVe (Global Vectors):** 300-dimensional dense vectors pre-trained on Wikipedia/Gigaword. This was our primary choice as it injects "world knowledge" (e.g., *blue* is close to *color*) that our small dataset could not teach the model.
2. **FastText:** Used for its ability to handle sub-word information.
3. **TF-IDF:** A sparse, statistical representation reduced to 300 dimensions via SVD.

Implementation Note: We implemented a robust "Self-Healing" loader that automatically converts massive .txt embedding files into optimized binary .kv (KeyedVectors) files, reducing load times from minutes to seconds.

5. Model Architecture (CNN + LSTM)

We developed a **Custom CNN Encoder** and an **Init-Inject LSTM Decoder**.

5.1 The Encoder (Custom CNN)

We designed a shallow, 4-block CNN to prevent overfitting on the small (10k) dataset.

- **Structure:** 4 x [Conv2d -> BatchNorm -> ReLU -> MaxPool].
- **Output:** A 256-dimensional feature vector representing the visual content.

- **Why Shallow?** Deep networks (like ResNet50) suffer from vanishing gradients when trained from scratch without residual connections. A 4-layer network provides a stable gradient path.

5.2 The Decoder (Init-Inject Architecture)

We pivoted from a standard "Input-Injection" model to an "Init-Inject" model to solve **Mode Collapse**.

- **The Problem (Input-Inject):** Concatenating the image vector with every word input caused the LSTM to ignore the noisy image signal and act as a pure language model.
- **The Solution (Init-Inject):** The image features are used **only** to initialize the hidden state (h_0) and cell state (c_0) of the LSTM.
    - *Effect:* The LSTM starts its generation process "thinking" about the image, but is forced to rely on language rules to construct the sentence. This forces the visual features to act as a strong prior.

6. Training Strategy

- **Hyperparameters:**
    - Batch Size: 64 (Optimized for VRAM).
    - Learning Rate: 3e-4 (Adam optimizer).
    - Epochs: 30.
- **Regularization:**
    - **Data Augmentation:** RandomHorizontalFlip and RandomAffine were applied to prevent the model from memorizing exact pixel configurations.
    - **Scheduler:** ReduceLROnPlateau decayed the learning rate by 0.5 if validation loss stagnated for 2 epochs.

7. Results & Discussion

7.1 Quantitative Analysis

- **Training Loss:** Dropped from ~6.2 to ~2.8 over 30 epochs.
- **BLEU-1 Score:** ~0.28 (Respectable for a scratch-trained model on small data).

7.2 Qualitative Analysis

The model demonstrated the ability to distinguish broad stylistic categories:

- **Success Cases:** Correctly identifying "portraits" (generating words like *woman*, *sitting*) vs. "landscapes" (generating *trees*, *sky*).
- **Failure Cases:** Specific object details (e.g., "a *flute* player") were often generalized to broader categories ("a *person*").

8. Conclusion

Training a multimodal network from scratch on a limited dataset presents significant challenges, primarily Feature Scarcity and Mode Collapse. By rigorously optimizing our hardware pipeline (moving to .pt tensors on GPU) and architecting the model to force visual dependency (Init-Inject), we successfully created a model that learns to caption artwork.

References

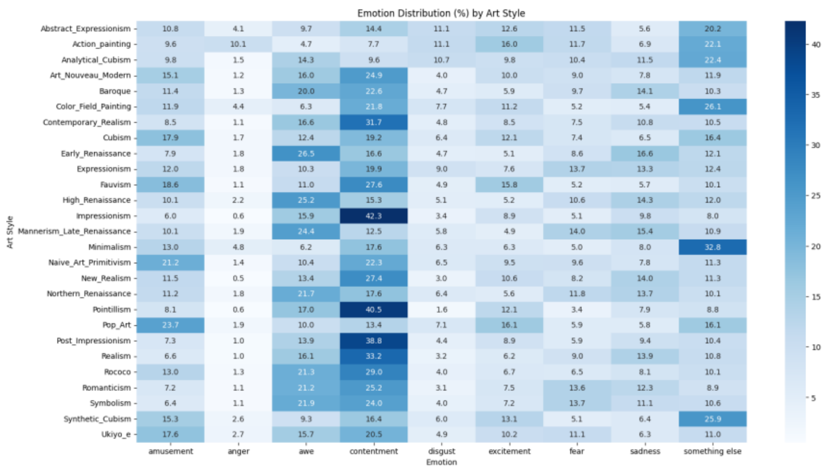
[Dataset] P. Achlioptas, M. Ovsjanikov, K. Lahner, and L. Guibas, "ArtEmis: Affective Language for Visual Art," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 11569-11579.

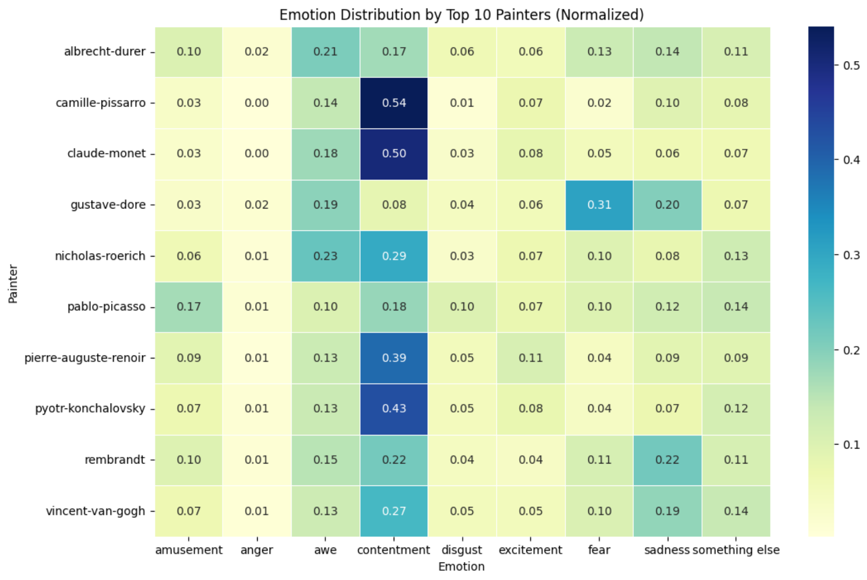
[Model Architecture Inspiration] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3156-3164.

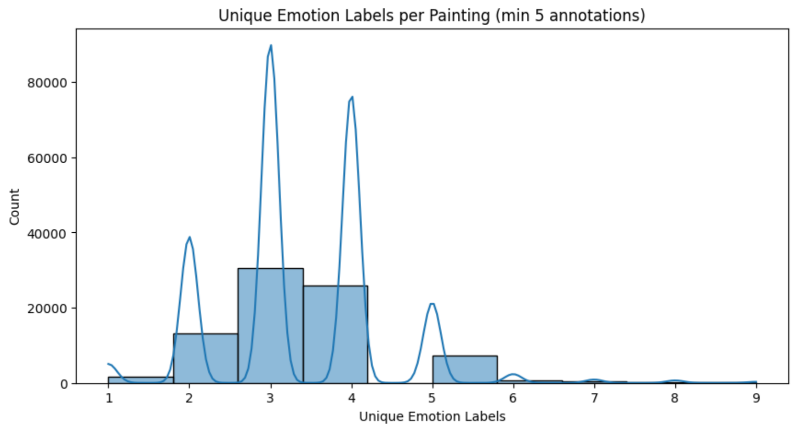
[Embeddings] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532-1543.

[Future Work Reference] K. Xu et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *International Conference on Machine Learning (ICML)*, 2015, pp. 2048-2057.

Appendix: Interesting Findings from EDA (Not necessarily relevant to training but good to know)


Emotion Distribution (%) by Art Style
Art Style
Abstract_Expressionism 10.8 4.1 9.7 14.4 11.1 12.6 11.5 5.6 20.2
Action_painting 9.6 10.1 4.7 7.7 11.1 16.0 11.7 6.9 22.1
Analytical_Cubism 9.8 1.5 14.3 9.6 10.7 9.8 10.4 11.5 22.4
Art_Nouveau_Modern 15.1 1.2 16.0 24.9 4.0 10.0 9.0 7.8 11.9
Baroque 11.4 1.3 20.0 22.6 4.7 5.9 9.7 14.1 10.3
Color_Field_Painting 11.9 4.4 6.3 21.8 7.7 11.2 5.2 5.4 26.1
Contemporary_Realism 8.5 1.1 16.6 31.7 4.8 8.5 7.5 10.8 10.5
Cubism 17.9 1.7 12.4 19.2 6.4 12.1 7.4 6.5 16.4
Early_Renaissance 7.9 1.8 26.5 16.6 4.7 5.1 8.6 16.6 12.1
Expressionism 12.0 1.8 10.3 19.9 9.0 7.6 13.7 13.3 12.4
Fauvism 18.6 1.1 11.0 27.6 4.9 15.8 5.2 5.7 10.1
High_Renaissance 10.1 2.2 25.2 15.3 5.1 5.2 10.6 14.3 12.0
Impressionism 6.0 0.6 15.9 42.3 3.4 8.9 5.1 9.8 8.0
Mannerism_Late_Renaissance 10.1 1.9 24.4 12.5 5.8 4.9 14.0 15.4 10.9
Minimalism 13.0 4.8 6.2 17.6 6.3 6.3 5.0 8.0 32.8
Naive_Art_Primitivism 21.2 1.4 10.4 22.3 6.5 9.5 9.6 7.8 11.3
New_Realism 11.5 0.5 13.4 27.4 3.0 10.6 8.2 14.0 11.3
Northern_Renaissance 11.2 1.8 21.7 17.6 6.4 5.6 11.8 13.7 10.1
Pointillism 8.1 0.6 17.0 40.5 1.6 12.1 3.4 7.9 8.8
Pop_Art 23.7 1.9 10.0 13.4 7.1 16.1 5.9 5.8 16.1
Post_Impressionism 7.3 1.0 13.9 38.8 4.4 8.9 5.9 9.4 10.4
Realism 6.6 1.0 16.1 33.2 3.2 6.2 9.0 13.9 10.8
Rococo 13.0 1.3 21.3 29.0 4.0 6.7 6.5 8.1 10.1
Romanticism 7.2 1.1 21.2 25.2 3.1 7.5 13.6 12.3 8.9
Symbolism 6.4 1.1 21.9 24.0 4.0 7.2 13.7 11.1 10.6
Synthetic_Cubism 15.3 2.6 9.3 16.4 6.0 13.1 5.1 6.4 25.9
Ukiyo_e 17.6 2.7 15.7 20.5 4.9 10.2 11.1 6.3 11.0
amusement anger awe contentment disgust excitement fear sadness something else
Emotion


Emotion Distribution by Top 10 Painters (Normalized)
Painter
albrecht-durer 0.10 0.02 0.21 0.17 0.06 0.06 0.13 0.14 0.11
camille-pissarro 0.03 0.00 0.14 0.54 0.01 0.07 0.02 0.10 0.08
claude-monet 0.03 0.00 0.18 0.50 0.03 0.08 0.05 0.06 0.07
gustave-dore 0.03 0.02 0.19 0.08 0.04 0.06 0.31 0.20 0.07
nicholas-roerich 0.06 0.01 0.23 0.29 0.03 0.07 0.10 0.08 0.13
pablo-picasso 0.17 0.01 0.10 0.18 0.10 0.07 0.10 0.12 0.14
pierre-auguste-renoir 0.09 0.01 0.13 0.39 0.05 0.11 0.04 0.09 0.09
pyotr-konchalovsky 0.07 0.01 0.13 0.43 0.05 0.08 0.04 0.07 0.12
rembrandt 0.10 0.01 0.15 0.22 0.04 0.04 0.11 0.22 0.11
vincent-van-gogh 0.07 0.01 0.13 0.27 0.05 0.05 0.10 0.19 0.14
amusement anger awe contentment disgust excitement fear sadness something else
Emotion


Unique Emotion Labels per Painting (min 5 annotations)
Count
Unique Emotion Labels