

AIML C24 - PGCP

Multimodal Media Retrieval and Captioning System

**Group - 6**

Lakshmi Narayana Mankala

Srinivasa Rao Chandakhanna

Venkat

Srikar Mukkavilli

**Project Mentor:**  Seshadri Mazumder

**Project Supervisor:**  Prof. Vineet Gandhi

May 24, 2025

Contents

[1. Introduction 3](#_Toc198991937)

[2. Phase 1: Retrieval (Image/Text Matching) 3](#_Toc198991938)

[Step-by-Step Process (Phase 1) 3](#_Toc198991939)

[Step 1: Data Acquisition & Preparation 3](#_Toc198991940)

[Step 2: Model Architecture Selection 4](#_Toc198991941)

[Step 3: Embedding Generation 4](#_Toc198991942)

[Step 4: Embedding Storage and Retrieval 4](#_Toc198991943)

[3. Phase 2: Captioning (Image to Text Generation) 4](#_Toc198991944)

[Step-by-Step Process (Phase 2) 4](#_Toc198991945)

[Step 5: Image Feature Extraction 4](#_Toc198991946)

[Step 6: Text Generation 4](#_Toc198991947)

[Step 7: Model Training & Experimentation 5](#_Toc198991948)

[Step 8: Evaluation 5](#_Toc198991949)

[4. Finalization and Deployment 5](#_Toc198991950)

[Step 9: Deployment Preparation 5](#_Toc198991951)

[5. Challenges 5](#_Toc198991952)

[6. Tools & Technologies 5](#_Toc198991953)

[7. Timeline 6](#_Toc198991954)

# 1. Introduction

This project proposes a multimodal deep learning system that performs cross-modal retrieval (matching images with text and vice versa) and automatic image captioning. The task is split into two phases:

1. Retrieval – Matching relevant text to images and vice versa using CLIP-based architectures.

2. Captioning – Automatically generating textual descriptions for input images.

This system has real-world applications in search engines, accessibility technologies, and content recommendation systems.

# 2. Phase 1: Retrieval (Image/Text Matching)

Objective: Develop a system capable of identifying and retrieving the most relevant image for a given text query and the most relevant text for a given image. The core idea is to understand the semantic similarity between visual and textual data using shared embedding space.

# Step-by-Step Process (Phase 1)

## Step 1: Data Acquisition & Preparation

- Dataset Collection: Curate a dataset comprising 8,000 and 30,000 click-bait images paired with their respective captions or associated text.

- Image-Text Pairing: Ensure proper one-to-one associations between each image and its corresponding textual context.

## Step 2: Model Architecture Selection

- CLIP Model Usage: Employ the CLIP (Contrastive Language-Image Pre-training) model for creating unified embeddings.

- CLIP Objective: Ensure semantically similar image-text pairs are closer in the shared embedding space.

## Step 3: Embedding Generation

- Pretrained Encoders: Use existing CLIP encoders to generate embeddings.

- Optional Fine-Tuning: If click-bait content differs significantly from CLIP’s pretraining, fine-tune using the collected dataset.

- Benchmark Validation: Validate the approach on public datasets like Flickr8k and Flickr30k.

## Step 4: Embedding Storage and Retrieval

- Efficient Storage: Save embeddings in a format conducive to fast querying.

- Similarity Querying: Build a querying system that computes cosine similarity to retrieve relevant matches.

- Future Accessibility: Design the system to allow seamless reuse in later stages.

# 3. Phase 2: Captioning (Image to Text Generation)

Objective: Generate natural language captions for input images by employing encoder-decoder architectures that translate visual content into text.

# Step-by-Step Process (Phase 2)

## Step 5: Image Feature Extraction

- Pretrained CNNs: Utilize ResNet-50 or VGG-16 to extract image features.

- Transformer Alternative: Optionally evaluate Vision Transformers (ViT) for improved representation.

## Step 6: Text Generation

- LSTM + Attention: Implement LSTM-based decoders with attention mechanisms to guide caption generation.

- Transformer Decoder: Explore the use of transformer-based architectures for enhanced performance.

- End-to-End Integration: Combine encoder and decoder modules to enable full image-to-text generation.

## Step 7: Model Training & Experimentation

- Training: Train on the same image-text pairs used in retrieval.

- Model Variations: Compare CNN + LSTM and ViT + Transformer decoder.

- Performance Tuning: Select the best model architecture via systematic evaluation.

## Step 8: Evaluation

- Quantitative Metrics: BLEU-1/2/3/4, METEOR, ROUGE, CIDEr.

- Comparison: Analyze metric scores to determine optimal model configuration.

# 4. Finalization and Deployment

## Step 9: Deployment Preparation

- Model Optimization: Refine the final model for deployment.

- Web-based Interface: Integrate into a usable application (e.g., via Streamlit or FastAPI).

- Practical Use Cases: Make the system demonstrable for real-world use and stakeholder presentation.

# 5. Challenges

- Domain-Specific Generalization

- Embedding Storage Efficiency

- Caption Diversity

- Model Overfitting

- Bias and Data Quality

# 6. Tools & Technologies

- Frameworks: PyTorch, TensorFlow, Hugging Face, Keras

- Monitoring: WandB, NLTK

- Deployment: FastAPI, Streamlit, Google Cloud, Heroku

- Models: CLIP, VGG-16, ResNet-50, LSTM, ViT, Transformers

# 

# 7. Timeline

Multimodal Media Retrieval and Captioning System