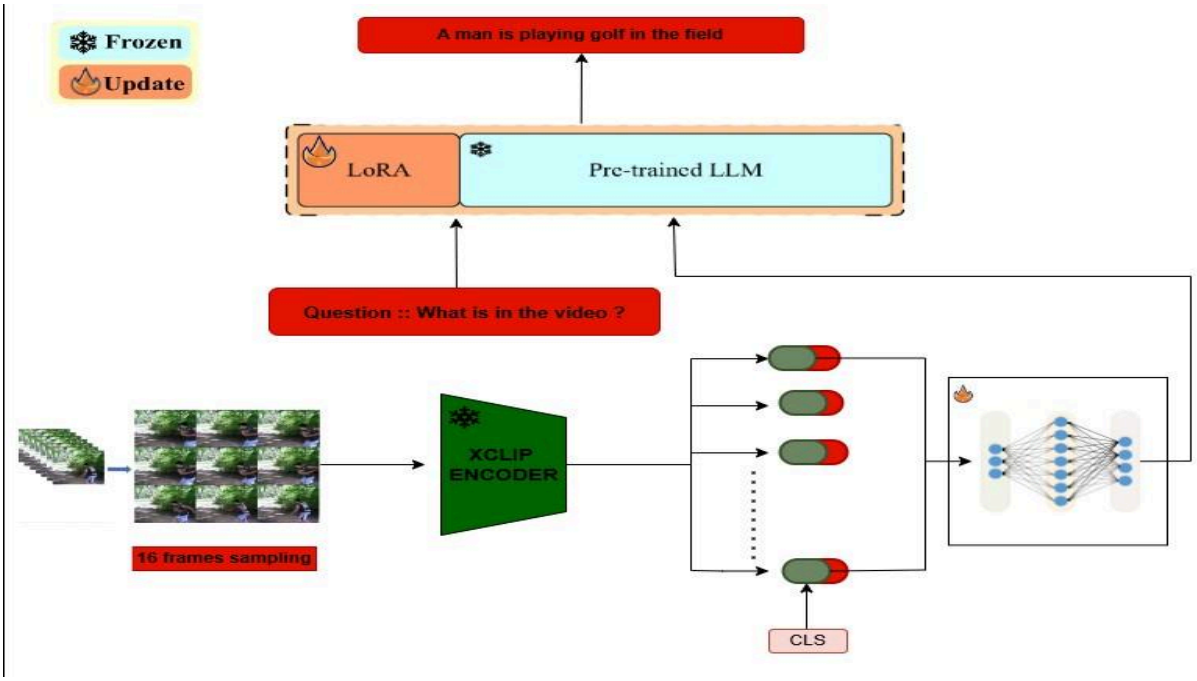
GROUP NO . 8

Approach to ML-Prep-A-thon Problem Statement for 13th Inter-IIT Tech Meet:

Overview:

For the task of processing GIFs and performing Visual Question-Answering on the processed GIFs, we considered the many architecture types mentioned in this Survey Paper on Multi-Modal Learning with Videos and LLMs. We finally settled on an architecture which involved fine-tuning with hybrid adapters which can be simplified to a Video-based Model which gives video embeddings, a connective adapter and an LLM which has insertive adapters to fine-tune it. The high-level architecture of the model is as follows:



Choice of each component:

● Video Embedder:

The video embedder chosen was XCLIP. We choose XCLIP due to the following reasons ::

1.Few Shot Learning

2.Good Model card on Hugging Face

3. Highly Interpretable model.

● Connective Adapter:

There were many options for connective adapters given in the previous paper which included MLP, Attention-based Adapters, Q-Former etc.

However, due to time and computation constraints, we opted for a simple Multi-Layer Perceptron which projected the video embeddings to the dimension of embeddings of LLMs

● LLM:

There were numerous models to pick from, but due to memory and computation constraints, we tried to go for models a bit on the lighter side. We experimented with Llama 3.2 1B model, Llama 3.2 3B model and GPT-2 Large model and decided to use Llama 3.2 3B as the final LLM due to its performance.

● Insertive Adapters:

We decided to choose LoRA as the insertive adapter combined with 4-bit quantization as numerous papers have experimentally shown that LoRA provides good results and is also memory efficient

Pipeline:

1) Albumentations GIF Processing and Frame Extraction:

The GIF is passed through a custom Albumentations processor written to make necessary changes to the GIF like resizing and other transforms, and then 16 frames are extracted from the GIF in uniform intervals . We didn't use the existing model processor as it was very slow and switching to Albumentation decreased our training time per epoch by more than **50%.**

2) Video Embedder:

The 16 frames are passed through the XCLIP model which gives 16 embeddings which represent the GIF. XCLIP in itself has a temporal encoder which captures the relationship between the frames .

3) LLM Embedder:

The tokenized question is passed through the LLM Embeddings to get embeddings for each token of the question

4) LLM:

The obtained GIF embeddings are concatenated to the start of the question embeddings and passed through the rest of the LLM. The

inference is made using some generation configuration with some decoding technique like greedy decoding or beam search decoding

5) Limited GPU :

We utilized Kaggle's platform for our project, which provided 30 hours of GPU runtime. However, due to limited storage capacity, we adopted a strategy where we downloaded 10,000 GIFs at a time, trained the model on these for one epoch, and then deleted the data to repeat the process for the next set of 10,000. Our estimated runtime for one epoch across 120,000 GIFs was approximately 15.6 hours. Due to time constraints, we were only able to complete a single epoch. Nevertheless, the model's loss had plateaued, and the results were satisfactory.

Possible Improvements:

- Better Connective Adapters like a Q-Former could have been utilized.
- LLM with more parameters could have been experimented with.
- Initially we were planning to use more than 16 frames by batching the frames of the videos into groups of 16 but due to time constraint we couldn’t apply masking to the padded frames and modify the XCLIP . So, we dropped this idea. We strongly believe that if we would have implemented this the results would have been far better .

Research and references:

- Video Understanding with Large Language Models: A Survey : A survey paper reviewing the various Multi-Modal video to language models.
- LLaMA: Open and Efficient Foundation Language Models : Paper introducing Llama models.\

- X-CLIP: End-to-End Multi-grained Contrastive Learning for Video-Text Retrieval