Medical Image Interpretation with Large Multimodal Models

Notebook for the CS_Morgan Lab at CLEF 2024

Mahmudul Hoque[1], Md Rakibul Hasan[1], Md. Ismail Siddiqi Emon[1], Fahmi Khalifa[2] and Md Mahmudur Rahman[1,*]

[1]Computer Science Department, Morgan State University, 1700 East Cold Spring Lane, Baltimore, Maryland 21251, USA
[2]Electrical and Computer Engineering Department, School of Engineering, Morgan State University, Baltimore MD 21251, USA

Abstract

This working note documents the participation of CS_Morgan in the ImageCLEFmedical 2024 Caption subtasks, focusing on Caption Prediction and Concept Detection challenges. The primary objectives included training, validating, and testing multimodal Artificial Intelligence (AI) models intended to automate the process of generating captions and identifying multi-concepts of radiology images. The dataset used is a subset of the Radiology Objects in COntext version 2 (ROCOv2) dataset and contains image-caption pairs and corresponding Unified Medical Language System (UMLS) concepts. To address the caption prediction challenge, different variants of the Large Language and Vision Assistant (LLaVA) models were experimented with, tailoring them for the medical domain. Additionally, a lightweight Large Multimodal Model (LMM), and MoonDream2, a small Vision Language Model (VLM), were explored. The former is the instruct variant of the Image-aware Decoder Enhanced à la Flamingo with Interleaved Cross-attentionS (IDEFICS) 9B obtained through quantization. Besides LMMs, conventional encoder-decoder models like Vision Generative Pre-trained Transformer 2 (visionGPT2) and Convolutional Neural Network-Transformer (CNN-Transformer) architectures were considered. Consequently, this enabled 10 submissions for the caption prediction task, with the first submission of LLaVA 1.6 on the Mistral 7B weights securing the 2nd position among the participants. This model was adapted using 40.1M parameters and achieved the best performance on the test data across the performance metrics of BERTScore (0.628059), ROUGE (0.250801), BLEU-1 (0.209298), BLEURT (0.317385), METEOR (0.092682), CIDEr (0.245029), and RefCLIPScore (0.815534). For the concept detection task, our single submission based on the ConvMixer architecture—a hybrid approach leveraging CNN and Transformer advantages—ranked 9th with an F1-score of 0.107645. Overall, the evaluations on the test data for the caption prediction task submissions suggest that LMMs, quantized LMMs, and small VLMs, when adapted and selectively fine-tuned using fewer parameters, have ample potential for understanding medical concepts present in images.

Keywords

Large Multimodal Models, Vision Language Models, Transformer, Large Language and Vision Assistant, Caption Prediction, Concept Detection, Medical Images, Low-Rank Adaptation, Quantization, Image-aware Decoder Enhanced à la Flamingo with Interleaved Cross-attentionS, Vision Generative Pre-trained Transformer 2.

1. Introduction

The tasks of automatic caption generation and multi-label prediction from medical images have become crucial for improving healthcare due to the growing availability of medical images from different modalities like X-radiation (X-ray), Computed Tomography (CT), Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI), and Ultrasound (US), as well as the significant advancements in the computing power of modern graphics processing units [1, 2, 3]. The increasing need for diagnostic radiology services and the lack of report writing expertise in many medical facilities highlight the need

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*Corresponding author.
mahoq1@morgan.edu (M. Hoque); mdhas1@morgan.edu (M. R. Hasan); mdemo1@morgan.edu (Md. I. S. Emon); fahmi.khalifa@morgan.edu (F. Khalifa); md.rahman@morgan.edu (M. M. Rahman)
https://github.com/HoqueMahmudul (M. Hoque); https://github.com/Hasan-MdRakibul (M. R. Hasan); https://github.com/ismailEmonFu (Md. I. S. Emon)
0009-0006-5532-4135 (M. Hoque); 0000-0002-6179-2238 (M. R. Hasan); 0000-0003-0595-229X (Md. I. S. Emon); 0000-0003-3318-2851 (F. Khalifa)

for automating the mentioned tasks. As a result, extensive applications of recently developed AI models have been found in these domains. As an active research area of AI, combining large language models (LLMs) with vision capabilities allows users to explore emergent abilities using multimodal data, which is being popularized as LMMs or VLMs [4]. For example, LLaVA [5], Flamingo [6], and Contrastive Language-Image Pretraining (CLIP) [7] have shown remarkable performance in various vision-text tasks. Consequently, there is also potential for applying LLMs in the biomedical imaging field [8]. These models are trained on extensive databases of human knowledge, demonstrating remarkable capabilities in offering valuable insights to physicians and healthcare professionals [9]. Utilizing knowledge from millions to billions of training examples, VLMs can help detect minor abnormalities in low-resolution radiology images that are difficult to spot with the naked eye [10]. Moreover, pre-trained LLMs like ChatGPT-4 [11] exhibit emergent abilities on tasks they were not specifically trained for (i.e., vision-language domain) [12]. Models like BiomedCLIP [13, 14], ChatDoctor [15], and GatorTron [16], which are pretrained on high-quality medical datasets, offer more useful applications for medical domain users. In this working note, various multimodal models were demonstrated that were initially pretrained on multimodal image-instruction pairs from diverse sources. This approach allows for attaining competitive results in this competition of analyzing medical images such as brain MRI, chest X-ray, PET, etc.

2. Objectives

For the ImageCLEFmedical Caption 2024 [17] challenge, CS_Morgan, participant in the competition, was tasked with developing solutions to automatically predict captions and identify multi-label concepts of radiology images from ROCOv2 [18] dataset. Considering the tasks, the objectives include the following:

- Concept Detection [19]: This task involved identifying and locating relevant concepts in the specified dataset. This formed the foundation for scene understanding and was essential for context-based image and information retrieval. The evaluation process was conducted using metrics like F1-score.
- Caption Prediction [19]: This task focused on predicting coherent captions for the entire image test dataset using the detected concepts and their interactions within the image. This task provided insights into the interplay of visual elements. Evaluation metrics used for this task consisted of BERTScore (as a primary approach), ROUGE (as a secondary approach), BLEU-1, BLEURT, METEOR, CIDEr, CLIPScore, RefCLIPScore, ClinicalBLEURT, and MedBERTScore.

3. Dataset

Dataset for both tasks included curated images from ROCOv2 [18], an updated version of the original ROCO [20] dataset. The medical images were collected from biomedical articles in the PMC OpenAccess and were accompanied by corresponding captions and concepts. The latter was also expressed using UMLS [21] terms. The training, validation, and test sets contained 70,108, 9,972, and 17,237 radiology images, respectively, with average dimensions of the images being 600×600. As a result, for the deep learning models implemented here, the images were resized to that average dimension, and the smaller images were padded to have a uniform distribution of image dimensions. Furthermore, the length of captions in words (without punctuations) or tokens for each image was 100 or fewer on average. Moreover, by analyzing both training and validation image-caption pairs, 42,121 unique words (without the punctuations) were found and used as the set of vocabulary in the models implemented. Additionally, there were 1,944 unique CUIs found in the concept list of the train and validation images, among which 1,934 were enlisted in the CUI mapping file.

4. Large Multimodal Models (LMMs)

LMMs as an extended variation of LLMs mark a major leap forward in AI by handling and comprehending various data types, including text, images, audio, and video [22, 23, 24]. By integrating and interpreting information from these diverse sources, LMMs achieve a holistic understanding of complex data [22, 23]. This capability allows them to perform sophisticated tasks, such as image captioning, visual question answering, and content recommendation, by leveraging the relationships between different data types [22, 23]. Figure 1 demonstrates theoretical architecture of LMMs.

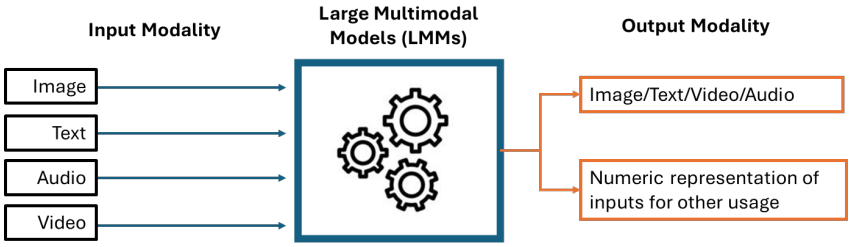

Figure 1: Theoretical architecture of large multimodal models.

4.1. Pre-training and Fine-tuning of LMMs

During pre-training, the model is initially trained on vast and diverse datasets, enabling it to learn general representations before being fine-tuned for specific tasks. This involves utilizing large-scale datasets that include various modalities [25, 26]. For instance, models like ViLBERT [27] have been pre-trained on extensive image-text pairs to increase their performance in downstream tasks like image captioning and visual question answering [25, 26].

Fine-tuning LMMs involves adjusting all pre-trained model parameters to enhance performance on specific tasks, such as image captioning. This process is computationally intensive and resource-demanding, especially for models with billions of parameters. Despite these challenges, the *full fine-tuning* technique remains popular due to its potential for achieving high accuracy. For instance, models like BLIP-2 [28] and InstructBLIP [29] have demonstrated enhancements in image captioning tasks through *full fine-tuning*, utilizing their extensive pre-training on large datasets to adapt to specific tasks. However, the substantial computational and memory requirements make *full fine-tuning* impractical for many applications, leading to the exploration of more efficient fine-tuning methods.

As a result, Parameter-Efficient Fine-Tuning (PEFT) [30, 31] presents a more efficient approach compared to *full fine-tuning* by modifying only a small portion of the model's parameters while leaving the majority unchanged. This strategy substantially decreases computational and memory demands, making it suitable for a variety of applications. In the domain of image captioning, PEFT techniques have proven effective with models such as mPLUG [32] and LLaVA [5]. Notably, approaches like Low-Rank Adaptation (LoRA) [33] have been particularly successful in fine-tuning. LoRA optimizes a matrix of updates to the pre-trained model weights rather than directly modifying them. This update matrix is decomposed into two smaller, lower-rank matrices, reducing the number of parameters that need updating while preserving the original weights [33, 34]. This allows different task-specific LoRAs to be easily swapped, effectively tailoring the pre-trained model for various applications. LoRA matches the performance of the *full fine-tuning* technique by updating a small number of additional weights, preventing catastrophic forgetting, and enabling better generalization with limited data [33, 34]. Figure 2 compares the approaches of LoRA and linear projection techniques.

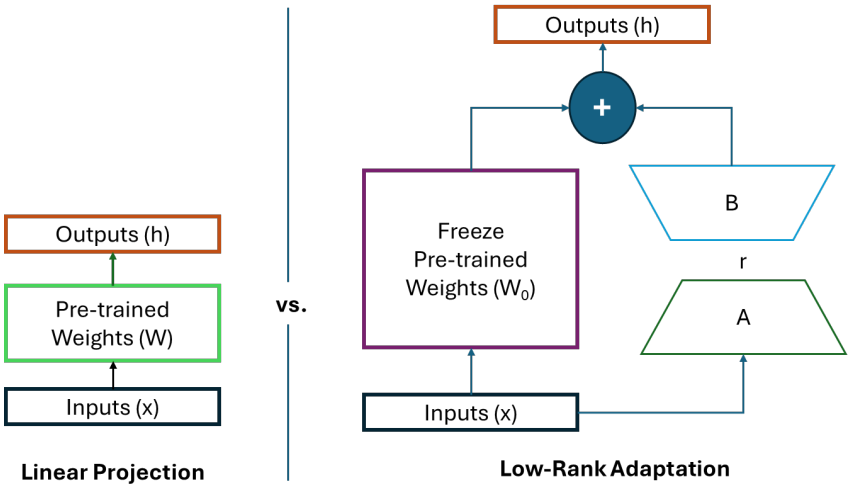

Figure 2: Comparison of approaches relevant to LoRA and linear projection techniques.

Figure 2 indicates that the LoRA approach involves two matrices, A and B. The matrix A is the first step in the adaptation process, projecting high-dimensional input features into a lower-dimensional latent space. Typically, its shape includes two values: rank and original dimension (e.g., 32 and 4096). The matrix B is the second component, mapping the lower-dimensional features back to the original high-dimensional space, effectively reversing the reduction performed by the matrix A and the shape becomes [4096, 32]. Both A and B matrices are trainable and updated during fine-tuning. LoRA focuses on specific weight matrices within the model, for example, the query, key, and value matrices in Transformer [35] architectures. However, traditional Transformers are hindered by their slow performance and high memory consumption, particularly with long sequences, due to the quadratic time and memory complexity of self-attention. Flash Attention [36] addresses these issues with an IO-aware exact attention algorithm that utilizes tiling to reduce the number of memory reads and writes between the GPU's high-bandwidth memory (HBM) and on-chip SRAM.

Visual instruction tuning [5] enhances LMMs by fine-tuning them with instructions that combine visual and textual data. This technique uses machine-generated instruction-following data to improve the model's zero-shot and few-shot performance on new tasks. For example, the LLaVA [37] model integrates a vision encoder with LLM for general-purpose visual and language understanding. The process involves generating detailed, context-aware language-image instructions using a language-only model like GPT-4 [11]. This data is then used to train the LMM, enabling it to perform tasks such as image captioning, visual question answering, and detailed image descriptions.

4.2. Large Language and Vision Assistant (LLaVA)

LLaVA [37, 38] stands as a comprehensive, end-to-end trained multimodal model that seamlessly merges a vision encoder and a LLM to facilitate broad-ranging visual and language comprehension (see Figure 3). The vision encoder is tasked with processing input images (X_v) and transforming them into a series of feature representations (Z_v). Situated above the vision encoder is the Projection (W), functioning as a vital conduit between the vision encoder and the language model. The projection matrix facilitates the conversion of feature representations (Z_v) from the vision encoder into a compatible format (H_v) for the language model. On the right side of the diagram, the Language Instruction input (X_q) represents the textual component that the model must comprehend and respond to in conjunction with the visual input. This input undergoes processing by the language model, generating its own set of feature representations (H_q). The Language Model (f_ϕ) (e.g., Vicuna 7B [39, 40] or Mistral 7B [41, 42] in this working note) ingests both the projected vision features (H_v) and the language features (H_q), seamlessly integrating them to produce a Language Response (X_a). The resulting output constitutes a coherent response incorporating elements from both visual and textual inputs. Figure 3 shows the basic

architecture of LLaVA and demonstrates its working principles.

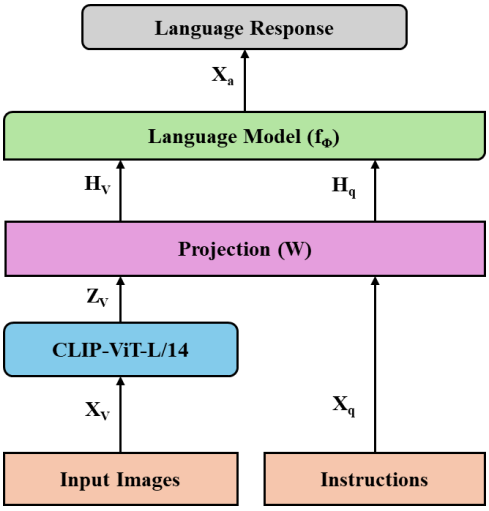

Figure 3: Architecture and working principle of LLaVA [37].

4.2.1. LLaVA-v.1.6-Vicuna-7B

The Vicuna 7B [39, 40] language model components include (see Figure 4): (a) Embedding Layer - Converts input tokens into dense vectors with an embedding dimension of 4,096, (b) Decoder Layers - Consists of 32 LLaMA-based Decoder Layer instances, where each layer includes a self-attention mechanism, a Multi-layer Perceptron (MLP) using Sigmoid Linear Unit (SiLU) activation, and Root Mean Square (RMS) normalization layers applied before and after the attention mechanisms, and (c) Final Normalization Layer - A RMS normalization layer applied to the final output of the decoder layers. The model supports input image resolutions of 672×672, 336×1344, and 1344×336, enhancing visual detail comprehension.

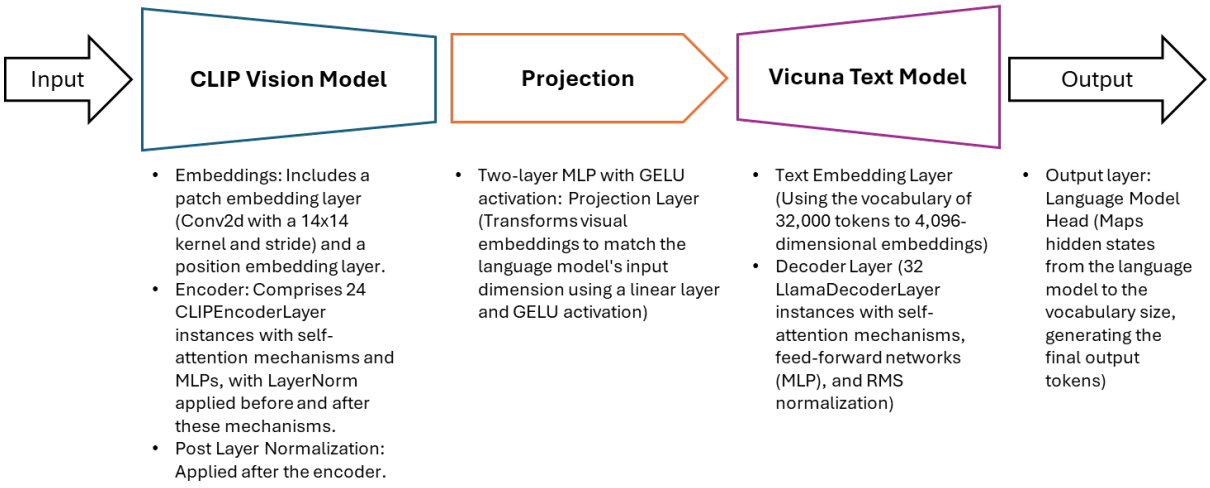

Figure 4: Major components and corresponding layers of LLaVA 1.6 Vicuna 7B model.

4.2.2. LLaVA-v.1.6-Mistral-7B

The LLaVA v.1.6 Mistral 7B [43] model integrates several key components for its functionality (see Figure 5). At its core is the vision encoder, utilizing a pre-trained CLIP ViT-L/14 [44] to extract visual

embeddings from high resolution images. This encoder processes visual input, converting it into a format compatible with the language model. The language model itself is based on the Mistral-7B architecture, which inherently incorporates advanced features like Sliding Window Attention and Grouped-Query Attention, enhancing its capability to manage long sequences and improve inference efficiency [41, 42]. Additionally, A two-layer MLP projection matrix is employed to map the visual embeddings from the vision encoder into the same embedding space as the language model, ensuring seamless integration of visual and textual information. The CLIP ViT-L/14 [44], a Vision Transformer(ViT) with 14 layers, is renowned for its ability to handle complex visual tasks, contributing to the model's overall performance.

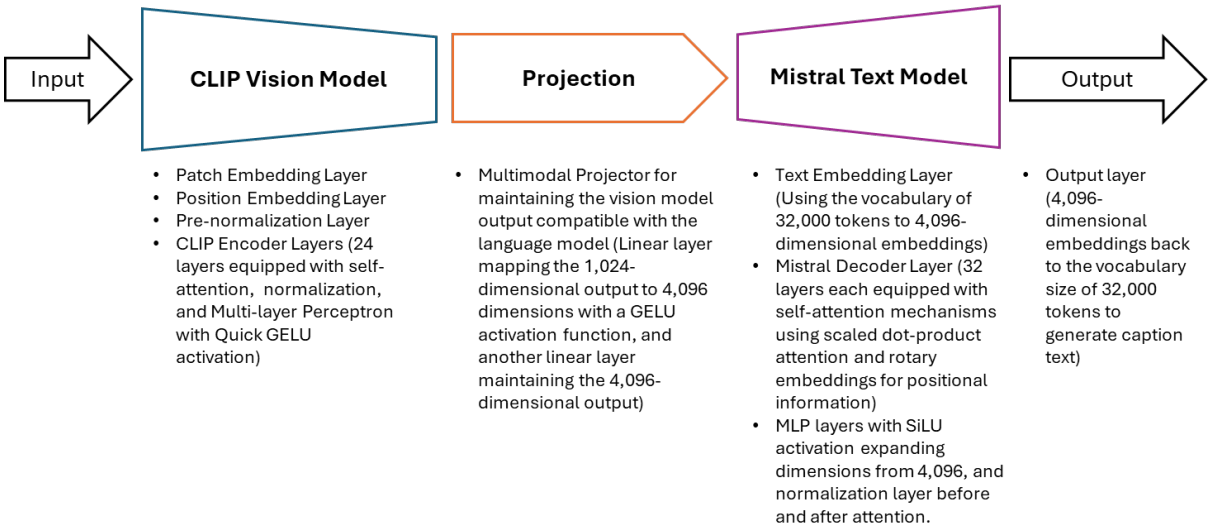

Figure 5: Important components and respective processing layers of LLaVA 1.6 Mistral 7B.

5. Caption Prediction Task

To address the caption prediction task, the CS_Morgan team fine-tuned several LMMs that were pre-trained using extensive standard datasets from the field of computer vision. These models were derived from well-known LLMs commonly utilized in Natural Language Processing (NLP). Ten submissions were made, and the technical details, methods, and approaches of these submissions are detailed in the following section. Moreover, the reproducible codes relevant to the following submissions can be found here [45].

Before any tasks are performed, the dataset is pre-processed to ensure that it is clean and correctly formatted. Beyond the initial image-text pre-processing described earlier, the training, validation, and testing datasets were structured for generating captions to meet the input requirements of the corresponding vision-language models. Furthermore, the dataset was managed using the Hugging Face Hub. Computational details can be found in Appendix A.

5.1. Submission 1: Selective fine-tuning of LLaVA-v.1.6-Mistral-7B

5.1.1. Model Description

For this submission, the pre-trained LLaVA 1.6 on Mistral 7B weights was loaded using Mistral-7B-Instruct-v.0.2 as the base LLM and flash attention was used to optimize attention mechanism computations. To enhance training stability, all float16 instances of the Vision Tower model were replaced with bfloat16. Additionally, prompts were set up by combining images and texts using the "mistral_instruct" conversation mode.

For efficient fine-tuning, LoRA was applied to specific layers, configuring it with a rank r = 16, an alpha (`lora_alpha`) of 32, and a dropout rate of 0.05. The query, key, and value projection layers in the self-attention mechanisms of the Mistral Decoder Layer, as well as the projection layers in the MLP, were specifically targeted. In the vision model, LoRA was applied to the linear projection layers within the self-attention mechanism (CLIP attention) of the encoder layers in the CLIP encoder. This resulted in 40,108,032 trainable parameters, about 0.527% of the model's total parameters. The LoRA components included `lora_A`, `lora_B`, and `lora_dropout` representing the low-rank projection to a smaller dimension, projection back to the original dimension, and a parameter to prevent overfitting, respectively.

5.1.2. Training Process

The training process involved setting up a Data Loader for the dataset, ensuring images and text inputs were properly loaded. Custom callbacks were defined for printing the best checkpoint and implementing early stopping. Key training parameters included a learning rate of 1e-4, bfloat16 precision, and the AdamW [46] optimizer. Each device processed batches of 4, with gradient accumulation steps of 8. Evaluations and saves were performed every 1,095 steps, with the training capped at 21,900 steps (10 epochs). Early stopping was set with a patience of 5 steps and a threshold of 0.01, monitoring evaluation loss (where lower values are better). Training was halted at 9,855 steps, and the best model, saved at 4,380 steps, was reloaded at the end. For evaluation, caption generation was configured with a temperature of 1.0, a beam width of 1, and a maximum of 512 new tokens. Figure 6 depicts the training and validation loss over the steps.

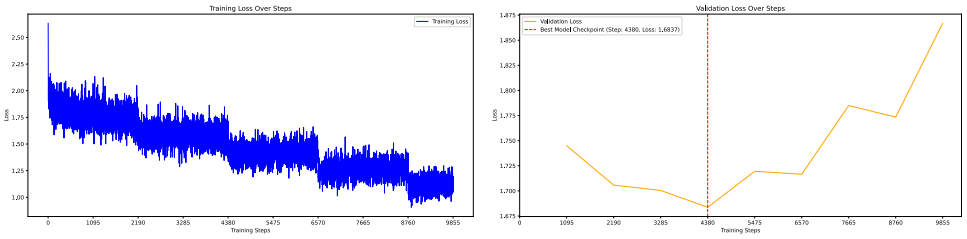

Figure 6: Training and validation loss of submission 1-LLaVA-v.1.6-Mistral-7B with LoRA for selective fine-tuning.

5.2. Submission 2: Additional fine-tuning of LLaVA-v.1.6-Mistral-7B Model

5.2.1. Model Description

The second submission was built upon the first one by fine-tuning a larger portion of the model using the same pattern. This included an expanded application of LoRA to improve utilization of the model's capacity for more accurate and robust predictions. The fine-tuning involved additional layers to enhance the learning and improve visual-textual alignment. Specifically, output projection layers such as `o_proj` in the Mistral Decoder Layer's self-attention mechanism and `out_proj` in the vision model were included to better capture complex relationships within the data, which is essential for tasks like image captioning. Targeting multimodal projector layers (`mm_projector.0` and `mm_projector.2`) enhanced the alignment of visual and textual representations, which is crucial for multimodal tasks. Despite the increased number of trainable parameters (98,467,840 compared to 40,108,032), this expansion represented only a small fraction (1.285%) of the total model parameters, maintaining parameter efficiency while improving learning capabilities. LoRA was configured with a rank r = 32, `lora_alpha` of 32, and a dropout rate of 0.05. Various layers were targeted in the Mistral Decoder Layers, including query projection (`q_proj`), key projection (`k_proj`), value projection (`v_proj`), and output projection (`o_proj`) in the self-attention mechanism, as well as gate projection (`gate_proj`), up projection (`up_proj`), and down projection (`down_proj`) in the MLP components.

In the CLIP Vision Model, LoRA was applied to the similar projection layers in the attention mechanism and fully connected layers (`fc1` and `fc2`) in the MLP. Additionally, the multimodal projector layers (`mm_projector.0` and `mm_projector.2`) were included to further enhance the model's capabilities. These modifications were applied to the LLaVA-v.1.6 model and its pre-trained checkpoints on the Mistral-7B.

5.2.2. Training Process

The training configuration included a learning rate of 1e-5, using the AdamW [46] optimizer, bfloat16 precision, and Flash Attention enabled. Each device handled a batch size of 4, with gradient accumulation steps set to 8. The model underwent training for a maximum of 8,760 steps (4 epochs), with checkpoints and evaluations performed every 548 steps. Early stopping parameters were defined with a patience of 4 and a threshold of 0.01, monitoring the evaluation loss to select the best model, with lower values being preferable. Training was halted early at 3,836 steps, and the model saved at this point was considered the best and subsequently loaded. For evaluation, specifically for generating captions, parameters were set with a temperature of 1.0, beam width of 1, and a maximum of 100 new tokens.

5.3. Submission 3: Hybrid fine-tuning of LLaVA-v.1.6-Mistral-7B

This submission was built on the previous one, maintaining the same general pattern but altering which layers were fine-tuned and the fine-tuning strategy itself. The fine-tuning strategy was multifaceted, employing LoRA to adapt key components, such as attention mechanism projections, MLP components, and multimodal projector layers. Additionally, the language model's head (`lm_head`) and the embedding tokens (`embed_tokens`) were explicitly set as trainable parameters to further enable these parts of the model to learn and adapt to the task. This hybrid approach leveraged the strengths of both LoRA adapters and traditional fine-tuning. Fine-tuning the `lm_head` allowed the model to better tailor its output generation to specific tasks or datasets, which was particularly important for generating appropriate language or captions from medical images. On the other hand, fine-tuning the `embed_tokens` layer helped the model learn better representations of input tokens, improving overall performance, especially when the input data distribution differs from the pre-training data.

In this configuration, LoRA was set with a rank $r = 32$, and the `lora_alpha` was calculated as $32 \times \sqrt{32}$ to stabilize training and enhance low-rank adaptation performance. This scaling factor normalized the learning rate for LoRA parameters based on rank, ensuring effective updates without causing gradient explosion or vanishing gradients. A dropout rate of 0.05 was applied to prevent overfitting and maintain generalization ability.

For layers explicitly set as trainable, the `lm_head` was a linear layer that mapped hidden states to the vocabulary space, generating the final output logits for each token. This layer was crucial for the model's text generation capability. The `embed_tokens` layer converted input token indices into dense vectors, providing initial representations of the input tokens essential for the model to process the input text. Both the `lm_head` and `embed_tokens` layers had their full weights fine-tuned, in addition to the LoRA adapters.

Overall, this hybrid fine-tuning approach combined LoRA fine-tuning for attention, MLP, and multimodal projection layers with full weight fine-tuning of the `lm_head` and `embed_tokens` layers. The total number of trainable parameters was 350,650,368 out of 7,654,729,728 total parameters, making up 4.581% of the parameters.

5.3.1. Training Process

The training arguments included a learning rate of 1e-5, the AdamW [46] optimizer, bfloat16 precision, Flash Attention, per-device batch sizes of 4, and gradient accumulation steps of 8. The model was trained for a maximum of 6,570 steps (3 epochs), with checkpoints and evaluations performed every 548 steps. Gradient checkpointing was enabled using a re-entrant approach to reduce memory usage. Early stopping was configured with a patience of 3 and a threshold of 0.01, monitoring evaluation loss

(with lower values being better). Early stopping was triggered at 3,836 steps, at which point the best model was saved and later loaded. For evaluation and caption generation, the parameters were set to a temperature of 1.0, `num_beams` of 1, and `max_new_tokens` of 100.

5.4. Submission 4: Selective Fine-tuning of LLaVA-v.1.6-Vicuna-7B

5.4.1. Model Description

For this submission, the pre-trained multimodal language model on checkpoints of Llava v.1.6 Vicuna 7B was loaded, which used the lmsys/vicuna-7b-v1.5 as its base LLM. The model preparation involved configuring LoRA with a rank (r) of 16, an `lora_alpha` of 32, and a dropout rate of 0.05. The target modules for LoRA were expanded to include the query (`q_proj`), key (`k_proj`), and value (`v_proj`) projections within the self-attention mechanism of the LLaMA Decoder Layer, as well as the gate (`gate_proj`), up (`up_proj`), and down (`down_proj`) projections in the MLP components of the same layer. Additionally, in the CLIP Vision Model's CLIP Encoder layers, the key (`k_proj`), value (`v_proj`), and query (`q_proj`) projections, along with the first (`fc1`) and second (`fc2`) fully connected layers of the CLIP MLP, were targeted. Furthermore, the multimodal projector layers (`mm_projector.0` and `mm_projector.2`) were included. This expanded application of LoRA resulted in 34,422,784 trainable parameters out of a total of 7,097,329,664 parameters, constituting approximately 0.485% of the model's parameters.

5.4.2. Training Process

The training process involved setting up a Data Loader for the dataset and inspecting batches to ensure correct loading of images and text inputs. Custom callbacks were created for printing the best checkpoint and enabling early stopping. The training used a learning rate of 1e-4, bfloat16 precision, Flash Attention, the AdamW optimizer, batch sizes of 4 per device, and gradient accumulation steps of 8, with evaluation and save steps every 548 steps. The model was trained for a maximum of 10,950 steps (5 epochs), with early stopping configured with a patience of 3 and a threshold of 0.01. The evaluation loss was monitored to select the best model, with lower values being preferable. Early stopping occurred at 4,932 steps, and the best model, saved at 4,384 steps, was loaded at the end. For generating captions during evaluation, parameters included a temperature of 1.0, `num_beams` set to 1, and a maximum of 512 new tokens.

5.5. Submission 5: Hybrid Fine-tuning of LLaVA-v.1.6-Vicuna-7B

For this submission, same approach of the third submission was followed. The only difference was the implementation of Vicuna LLM. The total number of trainable parameters is 346,718,208 out of 7,147,481,088 total parameters (4.851% of parameters). The training process employed was similar to the previous submission. The only difference was that the maximum token limit was set to 150 for this submission. The model was trained for a maximum of 10,950 steps (5 epochs), with early stopping configured with a patience of 5 and a threshold of 0.01. Early stopping occurred at 6,576 steps, and the best model, saved at 4,384 steps, was loaded for evaluation.

5.6. Submission 6: Selective Fine-tuning of LLaVA-v.1.5-7B

The LLaVA 1.5 7B shares a similar architecture with that of LLaVA-v.1.6 Vicuna-7B. LLaVA 1.5 checkpoints on 7B parameters were loaded, and the expanded use of LoRA resulted in 84,574,208 trainable parameters out of a total of 7,147,476,992, constituting approximately 1.183% of the model's parameters. Precision was adjusted from float16 to bfloat16 to enhance computational efficiency, and Flash attention was not enabled in this submission. Instead, LLaMA Scaled Dot-Product Attention (SDPA) was utilized in the 32 layers of the LLaMA Decoder Layer. LoRA was configured with a rank of 32, `lora_alpha` of 32, and a dropout rate of 0.05. Target modules for LoRA included various projections

in LLaMA Decoder Layer, MLP components, and attention mechanisms within CLIP Vision Model. The training process involved creating a Data Loader, defining custom callbacks for early stopping and checkpoint printing, and setting training arguments such as a learning rate of 1e-5, and AdamW optimizer. Training was conducted for a maximum of 8760 steps with early stopping triggered at 4,672 steps, saving the best model. For evaluation, parameters included temperature = 1.0, `num_beams` = 1, and `max_new_tokens` = 100.

5.7. Submission 7: Adaptation of MoonDream2

5.7.1. Model Description

Moondream2, a small vision language model designed for efficient operation on edge devices, was evaluated on the ImageCLEF 2024 dataset using pre-trained weights from Huggingface [47, 48]. These weights were initialized from Sigmoid Loss for Language-Image Pre-Training (SigLIP) and Phi-1.5 models. Phi-1.5 [49], developed by Microsoft Research, is a compact Transformer-based language model with 24 layers, 32 heads (each with a dimension of 64), rotary embeddings, a rotary dimension of 32, a context length of 2,048, and flash-attention. SigLIP [50], an enhancement of the CLIP model, replaces the softmax loss with a pairwise sigmoid loss, operating on image-text pairs without global normalization. SigLIP's architecture includes a ViT [51] backbone that processes image patches through a transformer encoder and a classification head with a MLP using Gaussian Error Linear Unit (GELU) activation for final predictions. Moreover, the pre-processing included resizing, type conversion, and normalization. This architecture effectively combined visual and textual processing for caption generation.

LoRA was configured with an alpha (`lora_alpha`) of 32, which adjusts the learning rate for low-rank matrices, and a rank (`lora_rank`) of 64 for the adaptation process. It was applied to specific linear layers in both the vision encoder and the text model. In the vision encoder, LoRA targeted the projection layers (`proj`), and fully connected layers (`fc1` and `fc2`) within the 27 ViTBlock components. Additionally, LoRA was applied to the `fc1` and `fc2` layers in the multimodal projection layer, a custom module integrated to adapt the projection layer for the purpose of the caption prediction challenge. In the language model, LoRA targeted the `Wqkv`, `out_proj`, `fc1`, and `fc2` layers within the 24 Phi Decoder Layer components. `Wqkv` in the Phi Decoder Layer represents the combined weights for the self-attention mechanism's linear projections (query, key, and value). With LoRA applied, the model had 74,422,272 trainable parameters, which was about 3.850% of the total parameters (1,931,904,880).

5.7.2. Training Process

The training process employed various key parameters and strategies to optimize the model's performance. The number of image tokens was set to 729, aligning with text tokens. Training spanned 10 epochs over 40,000 steps, using a batch size of 8 and gradient accumulation steps of 4, with evaluation after each epoch. An early stopping mechanism with a patience of 6 epochs and a minimum delta of 0.0001 monitored validation loss to prevent overfitting. Data loading and batching utilized PyTorch's DataLoader with custom collation for images and text tokens, pre-processed and padded for uniform sequence lengths. Gradient accumulation steps set to 4 simulated a larger batch size for better GPU memory management. The Adam8bit optimizer from the bitsandbytes library, with a dynamic learning rate adjusted via a cosine schedule, was used. Loss computation combined image and text embeddings, processed by the Phi language model. The training loop iterated over epochs and batches, updating parameters post-gradient accumulation and checking validation loss for early stopping. LoRA parameters were optimized with an initial learning rate of 3e-6, scaled by a factor of 4, balancing exploration and convergence. This approach, along with gradient accumulation, enhanced resource use and fine-tuning efficiency.

5.8. Submission 8: Selective Fine-tuning of 4-bit Quantized IDEFICS-9B-Instruct

5.8.1. Model Description

IDEFICS 9B Instruct [52] is an advanced multimodal model developed by Hugging Face for integrated image and text processing tasks. The model combines the vision model CLIP ViT-H/14 [53] and the language model LLaMA 7B [54], incorporating novel transformer blocks to connect these modalities. Trained on extensive datasets, including OBELICS, Wikipedia, LAION, and PMD, the IDEFICS 9B Instruct variant is fine-tuned on supervised and instruction datasets.

The lightweight IDEFICS 9B Instruct variant was explored using 4-bit quantization to reduce model size and computational requirements while maintaining performance. BitsandByters (BnB) quantization assigns 4-bit precision to the model using double quantization with the normalized floating-point format (NF4) and bfloat16 precision for computations, crucial for running large language models on smaller devices. For fine-tuning IDEFICS 9B Instruct on the ImageCLEF dataset, the checkpoint *HuggingFaceM4/idefics-9b-instruct* was specified to load the pre-trained model with 4-bit quantization using the BitsAndBytesConfig class.

LoRA was applied to the query projection (`q_proj`), key projection (`k_proj`), and value projection (`v_proj`) in both the ViT and decoder layers, as well as the perceiver attention and gated cross-attention layers. However, the output projection (`o_proj` and `out_proj`) in the decoder, gated cross-attention, and perceiver attention layers did not use LoRA but remained as standard Linear4bit layers. This selective application of LoRA allows for efficient fine-tuning by reducing the number of trainable parameters specifically within the attention mechanisms while leaving other projections, like the output projection layers, unmodified.

5.8.2. Training Process

Custom callbacks were defined for printing the best checkpoint and early stopping. The training arguments included a learning rate of 1e-4, the AdamW optimizer, batch sizes of 2 per device for training and evaluation, gradient accumulation steps of 8, and evaluation and save steps every 500 steps. The model was trained for a maximum of 8762 steps (2 epochs). Early stopping parameters were set with a patience of 6 and a threshold of 0.001. Evaluation loss was monitored to select the best model, with lower values being better. Early stopping was triggered at 8,000 steps, and the best model, saved at 8,000 steps, was loaded at the end of training.

5.9. Submission 9: VisionGPT2

5.9.1. Model Description

The Encoder-Decoder model was designed to take an image as input and generate a descriptive caption as output. In this model, the Encoder was a ViT [55, 51] that processed the input image and extracted meaningful features. These features were then fed into the Decoder, which is based on GPT-2 [56], a powerful language model that generates the corresponding textual caption. For fine-tuning the model, the Hugging Face Seq2SeqTrainer [57] was employed. This trainer, part of the Hugging Face transformers library, is specifically designed to handle sequence-to-sequence tasks, making it well-suited for this image captioning model. The fine-tuning process leverages the transformers library to adapt the pre-trained ViT and GPT-2 models.

5.9.2. Training Process

Initially, the pre-trained layers were frozen to focus on training the cross-attention layers. In subsequent epochs, GPT-2 was unfrozen and trained, and in the final few epochs, the ViT was also unfrozen. The Adam optimizer and the One Cycle Learning Rate (OneCycleLR) scheduler are used for optimization. Mixed-precision fp16 training was employed with autocast and GradScaler in PyTorch. The training

metrics are cross-entropy loss and perplexity, with both metrics aimed to be minimized. The best model was saved based on validation perplexity and was loaded during caption generation.

5.10. Submission 10: CNN-Transformer Fusion Model

The CNN-Transformer fusion model for this submission was built around three core models. First, the pre-trained ChexNet [58] (a DenseNet121 backbone based CNN model) was used to extract features from the input images. These features captured essential visual information and were then passed to the second component, a Transformer Encoder [59]. The Transformer-based encoder processed the extracted image features to generate a new, more informative representation of the inputs. Finally, the third component, a Transformer Decoder [59], took the output from the encoder along with the text data (sequences). The decoder used these inputs to learn and generate the corresponding image captions, completing the image-to-text translation process. The hyper-parameters for the model included an embedding dimension set to 512 and an initial learning rate of 0.0001. The encoder used a single attention head, while the decoder utilized two attention heads to process the information. For early stopping, the patience level was set to 5, meaning the training process halted if there was no improvement in validation loss after five epochs.

5.11. Performance Measurement Metrics for the Caption Prediction Task

The performance of all the submissions regarding the caption generation task were evaluated using the following metrics.

- BERTScore [60] evaluates text generation by computing the similarity between BERT embeddings of the candidate and reference sentences, capturing semantic meaning better than traditional metrics.
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [61] is a set of metrics for evaluating automatic summarization and machine translation by comparing overlap in n-grams, word sequences, and word pairs between the candidate and reference texts.
- BLEU (Bilingual Evaluation Understudy) [62] is a precision-based metric for evaluating machine translation quality by comparing n-grams of the candidate translation to those of the reference translation. BLEU-1 specifically considers unigram matches.
- BLEURT (Bilingual Evaluation Understudy with Representations from Transformers) [63] is a learned evaluation metric for natural language generation that uses pre-trained transformers fine-tuned on a variety of supervised and unsupervised signals to predict human judgment scores.
- METEOR (Metric for Evaluation of Translation with Explicit ORdering) [64] evaluates machine translation by considering precision, recall, stemming, synonymy, and alignment, aiming to improve correlation with human judgment.
- CIDEr (Consensus-based Image Description Evaluation) [65] is a metric for evaluating image captioning by comparing candidate captions to reference captions using TF-IDF weighting and n-gram similarity, ensuring relevance and importance of the words are considered.
- CLIPScore [66] is an evaluation metric that uses the CLIP model to compare image and text similarity. It measures the alignment between visual content and textual descriptions, providing a score based on their embedding similarity.
- RefCLIPScore [66] is an extension of CLIPScore that includes a reference-based evaluation, incorporating both the similarity of the generated text to a reference text and the similarity between the image and the generated text.
- ClinicalBLEURT [67] adapts BLEURT for clinical text generation, fine-tuning it on clinical datasets to better evaluate the quality and relevance of generated clinical text against reference clinical text.
- MedBERTScore [67] adapts BERTScore for the medical domain, using BERT embeddings specifically fine-tuned on medical texts to provide a more accurate evaluation of medical text generation tasks.

5.12. Results and Discussion on Caption Prediction Submissions

In this year's evaluation for the ImageCLEF task, BERTScore [19] was the primary metric used to assess the quality of the generated captions, with ROUGE [19] as the secondary metric.

Table 1 shows the results of submissions in terms of the primary metrics of performance. In addition to BERTScore and ROUGE, some other performance metrics were also adopted to assess submission results. These metrics are BLEU-1, BLEURT, METEOR, CIDEr, CLIPScore, RefCLIPScore, ClinicalBLEURT, and MedBERTScore. Table 2 shows the results of the additional performance metrics other than the BERTScore and ROUGE used for the caption prediction task. In both tables, the submissions are listed according to the BERTScore (highest to lowest).

Table 1
Submission Results for the Caption Prediction Task - Primary Scores

Submission ID	Model	# of Parameters Trained	% of Parameters Trained	BERTScore	ROUGE
1	LLaVA-v.1.6 Mistral-7B	40108032	0.527	0.628059	0.250801
4	LLaVA-v.1.6-Vicuna-7B	34422784	0.485	0.625402	0.245398
3	LLaVA-v.1.6-Mistral-7B	350650368	4.581	0.624988	0.243983
2	LLaVA-v.1.6-Mistral-7B	98467840	1.285	0.622964	0.238009
8	IDEFICS-9B-Instruct	21061632	0.235	0.621052	0.229319
6	LLaVA-v.1.5-7B	84574208	1.183	0.617342	0.217850
7	MoonDream2	74422272	3.852	0.616561	0.215981
5	LLaVA-v.1.6-Vicuna-7B	346718208	4.851	0.615692	0.223682
9	VisionGPT2	28366848	13.493	0.545773	0.118446
10	CNN-Transformer Fusion Model	9053056	99.084	0.414342	0.044218

Our results indicate that LMMs, when selectively fine-tuned with fewer parameters, can achieve high performance. Additionally, LMMs obtained through quantization and smaller VLMs can maintain competitive performance in medical image understanding and caption generation. From Tables 1 and 2, it is evident that four different submission outperformed the others in terms of the pre-specified performance measurement metrics. Submission 1 using the LLaVA-v1.6-Mistral-7B model with 40.1M fine-tuned parameters using the LoRA technique achieved the highest scores across several key metrics: BERTScore (0.628059), ROUGE (0.250801), BLEU-1 (0.209298), BLEURT (0.317385), METEOR (0.092682), CIDEr (0.245029), and RefCLIPScore (0.815534). Submission 3, also using the LLaVA-v.1.6-Mistral-7B model with hybrid LoRA fine-tuning approach (350.6M parameters) attained the highest CLIPScore of 0.824171, indicating an improved semantic match between the generated captions and the visual content of the medical images. Submission 10, the CNN-Transformer fusion approach (Pre-trained CheXNet as the encoder and Transformer as the decoder) performs better than other submissions in terms of the ClinicalBEURT score of 0.676905. Finally, the submission 8 which was IDEFICS-9B-Instruct quantized to 4-bit, excelled in capturing relevant biomedical concepts compared to other submissions, achieving the highest MedBERTScore of 0.657460034. Overall, the first submission can be claimed as the top performer because of the highest scores in the primary and secondary metrics. Figure 7 shows the comparison of the submissions in terms of the primary and secondary metrics. The significance of these submissions lies in their demonstration of advanced fine-tuning techniques and model performance optimization in the context of generative models. These findings highlight the evolving landscape of model fine-tuning strategies, advocating for resource-efficient methods that maintain or enhance performance. This is crucial for practical and scalable AI deployments across diverse medical applications.

Table 2
Submission Results for the Caption Prediction Task - Secondary Scores

Submission ID	BLEU-1	BLEURT	METEOR	CIDEr	CLIPScore	RefCLIPScore	ClinicalBLEURT	MedBERTScore
1	0.209298	0.317385	0.092682	0.245029	0.821262	0.815534	0.455942	0.632664
4	0.207555	0.316524	0.089231	0.224142	0.820785	0.814251	0.443495	0.631529
3	0.204902	0.315257	0.089844	0.219909	0.824171	0.814689	0.443766	0.630013
2	0.195061	0.309630	0.085367	0.203407	0.822694	0.812071	0.435760	0.629846
8	0.154041	0.296429	0.077370	0.191725	0.811816	0.807033	0.443966	0.657460
6	0.155887	0.297628	0.072998	0.170832	0.816577	0.806713	0.448721	0.626198
7	0.182720	0.305801	0.076002	0.161918	0.815116	0.807100	0.453388	0.624334
5	0.174058	0.300482	0.076997	0.172990	0.819258	0.807451	0.433863	0.624581
9	0.102386	0.244725	0.035134	0.028805	0.685257	0.692432	0.450116	0.556308
10	0.028921	0.261376	0.019912	0.003447	0.666524	0.669843	0.676905	0.406249

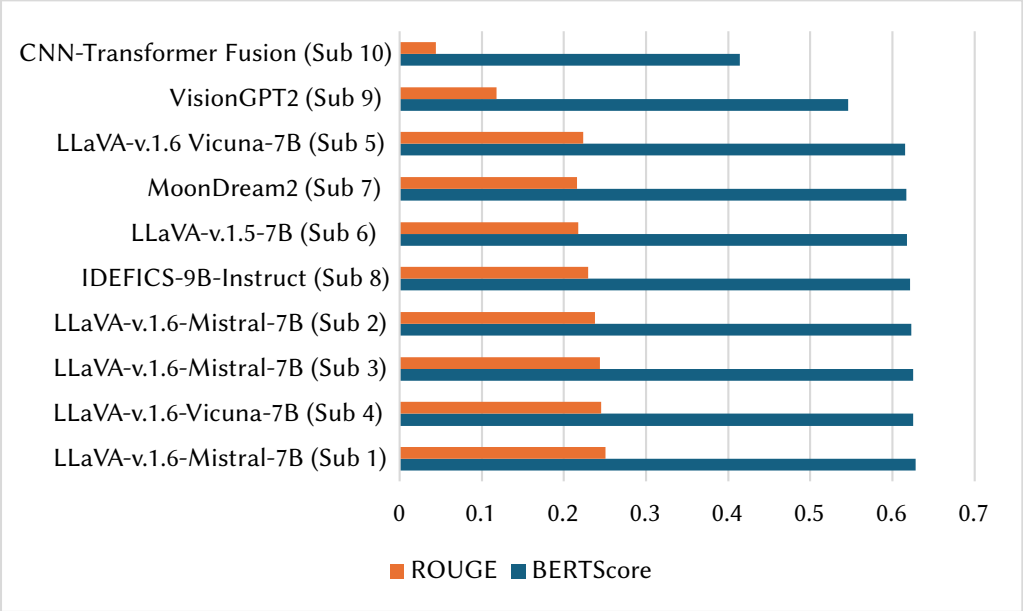

Figure 7: Comaprison of the submissions in terms of BERTScore and ROUGE values.

In addition to the above-mentioned submissions, Submission 4, utilizing the LLaVA v.1.6 Vicuna 7B with selective fine-tuning using LoRA (34.4M parameters), demonstrated well-balanced performance and closely followed Submission 1. Moreover, submissions 3 and 2, both based on the LLaVA v.1.6 Mistral 7B model but with different approaches for optimization, closely followed submission 4 in terms of BERTScore and ROUGE. However, the sixth submission of LLaVA v.1.5 7B, based on another variant of LLaVA, could not outperform the LLaVA v.1.6 variants except for LLaVA v.1.6 Vicuna 7B with hybrid fine-tuning using LoRA technique (Submission 5). The experiment with the MoonDream2 with 74.4M fine-tuned parameters in Submission 7 showed competitive performance on the test data relative to the larger models across multiple metrics. Submissions 9 and 10 were based on the pre-trained Transformer-based encoder-decoder models other than the LMMs. VisionGPT2 outperformed the conventional pre-trained CheXNet-Transformer or CNN-Transformer based model in every metric except ClinicalBLEURT. Table 3 shows the generated captions for a test image (ID: ImageCLEFmedical_Caption_2024_test_000016) corresponding to the submissions made for the caption prediction task.

Table 3
Model predicted captions corresponding to the submissions for the test image ID: ImageCLEFmedical_Caption_2024_test_000016 (Ground Truth Caption: *Complex pelvic fracture with widening and displacement of the pubis rami and widening of the S1joints bilaterally: arrows shows multiple fractures in the pelvis.*)

Submission ID	Model	Predicted Caption
1	LLaVA-v.1.6 Mistral-7B	*Anteroposterior radiograph of the pelvis showing a sacral fracture (yellow arrows) and a pubic fracture (yellow arrowhead).*
4	LLaVA-v.1.6-Vicuna-7B	*X-ray of the pelvis showing bilateral sacroiliitis (yellow arrows) and bilateral pubic symphysis (yellow arrowheads).*
3	LLaVA-v.1.6-Mistral-7B	*X-ray of the pelvis showing a large pelvic mass (arrows).*
2	LLaVA-v.1.6-Mistral-7B	*Plain radiograph of the pelvis showing a large pelvic mass (yellow arrows) with a large right-sided pelvic hematoma.*
8	IDEFICS-9B-Instruct	*X-ray of the pelvis showing the presence of a foreign body in the bladder (yellow arrow) and the presence of a foreign body in the rectum.*
6	LLaVA-v.1.5-7B	*X-ray of the pelvis showing the fracture of the right pubis.*
7	MoonDream2	*Anteroposterior radiograph of the pelvis showing a right-sided sacroiliitis.*
5	LLaVA-v.1.6-Vicuna-7B	*X-ray of the pelvis showing the fracture of the right ilium (yellow arrows).*
9	VisionGPT2	*CT scan of the chest. The CT scan showed a nodule in the right upper lobe.*
10	CNN-Transformer Fusion Model	*Bone defect detected in the axillary region.*

6. Concept Detection Task

This year CS_Morgan team submitted a single submission for the concept detection task. The submission involved the implementation of ConvMixer [68, 69] model which combined the CNN and Transformer architectures.

6.1. Model Description

ConvMixer [68, 69] closely resembles the MLP-Mixer [70] model, with key differences in its architecture. Instead of fully-connected layers, ConvMixer employs standard convolution layers. It uses batch normalization rather than layer normalization technique, which is typically used in ViT [51] and MLP-Mixers [70]. ConvMixer utilizes two types of convolution layers: depth-wise convolutions for mixing spatial locations of the images and point-wise convolutions, following the depth-wise convolutions, for mixing channel-wise information across the patches. Additionally, ConvMixer uses larger kernel sizes to achieve a larger receptive field. Figure 8 shows the corresponding architecture of the model.

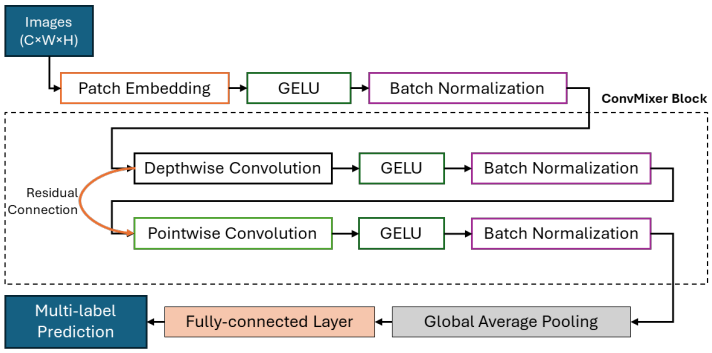

Figure 8: Key components and layers of the ConvMixer model used for the concept detection task submission.

6.2. Training and Result

The training process involved developing a ConvMixer model designed for classification or concept detection task with 1,944 unique CUIs. The model was built using TensorFlow and Keras, with key components including an initial rescaling layer, a patch extraction stem, and a series of ConvMixer blocks. The model utilized GELU activations and batch normalization for better performance. The architecture included a global average pooling layer followed by a dense output layer with a sigmoid activation function. Training was conducted over 200 epochs with a batch size of 8, a learning rate of 0.001, and a weight decay of 0.0001. The Adam optimizer was used for training, and the binary cross-entropy loss function was chosen for the multi-label classification task. Performance metrics such as accuracy, precision, recall, and Area Under the Curve (AUC) were tracked during training. However, on the F1-score was reported for the submission. A model checkpoint callback was implemented to save the best model based on validation accuracy. After training, the model was evaluated using the best checkpointed weights.

By implementing this model, the F1-score of 0.107645 was attained on the test data, placing it the ninth position for the concept detection task among the participants. This score indicates that the model's performance in terms of precision and recall is relatively low, as it represents the harmonic mean of precision and recall, providing a single metric that balances both. The score suggests that the model is struggling to correctly identify and classify the relevant instances among the 1,944 classes, leading to either a high number of false positives, false negatives, or both. This low score reflects room for improvement in the model's ability to accurately predict the target labels. For a test image (ID: ImageCLEFmedical_Caption_2024_test_000016), the predicted concepts or CUIs based on this ConvMixer model were C0030797, C0000726, and C1306645, whereas the ground truth concepts were C1306645, C0030797, and C0034014 (See Figure 9).

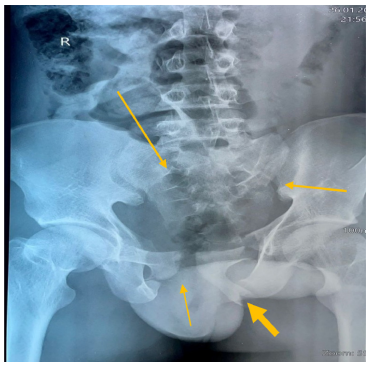

Figure 9: Ground Truth Concept CUIs: [C1306645, C0030797, C0034014] and **Predicted Concept CUIs:** [C0030797, C0000726, C1306645] of the test image ID: ImageCLEFmedical_Caption_2024_test_000016 (CC BY [Munihire et al. (2023)])

7. Conclusion

For the Caption Prediction task, submitted models included LLaVA v.1.6 with Mistral 7B and Vicuna 7B checkpoints, as well as the LLaVA v.1.5 7B model. Additionally, a 4-bit quantized instruct variant of the IDEFICS 9B model and MoonDream2, a compact VLM, were explored. Two fine-tuning strategies, selective and hybrid fine-tuning, were utilized. Furthermore, traditional encoder-decoder models like VisionGPT2 and CNN-Transformer architectures were also experimented with. Among these, the top-performing submission was the selective training of the LoRA projectors (40.1M parameters) on the LLaVA 1.6 model with Mistral 7B weights. For the Concept Detection subtask, a single model based on the ConvMixer architecture was submitted, which combines the strengths of CNNs and Transformers.

In future research, the primary aim will be to incorporate Explainable AI and reinforcement learning. Explainable AI will enhance model safety and reliability by identifying potential failures and undesirable actions in LMMs. Reinforcement learning, using context-aware reward modeling, will integrate detailed medical image concepts to improve content understanding and performance in multimodal tasks.

8. Acknowledgments

This work was supported by the National Science Foundation (NSF) grant (ID. 2131307) "CISE-MSI: DP: IIS: III: Deep Learning-Based Automated Concept and Caption Generation of Medical Images Towards Developing an Effective Decision Support."

References

[1] I. Allaouzi, M. Ben Ahmed, B. Benamrou, M. Ouardouz, Automatic caption generation for medical images, in: Proceedings of the 3rd International Conference on Smart City Applications, 2018, pp. 1–6.

[2] T. Pang, P. Li, L. Zhao, A survey on automatic generation of medical imaging reports based on deep learning, BioMedical Engineering OnLine 22 (2023) 48.

[3] R. Li, Z. Wang, L. Zhang, Image caption and medical report generation based on deep learning: a review and algorithm analysis, in: 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI), IEEE, 2021, pp. 373–379.

[4] M.-H. Van, P. Verma, X. Wu, On Large Visual Language Models for Medical Imaging Analysis: An Empirical Study, arXiv e-prints (2024) arXiv–2402.

[5] H. Liu, C. Li, Q. Wu, Y. J. Lee, Visual instruction tuning, Advances in neural information processing systems 36 (2024).

[6] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al., Flamingo: a visual language model for few-shot learning, Advances in neural information processing systems 35 (2022) 23716–23736.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.

[8] D. Tian, S. Jiang, L. Zhang, X. Lu, Y. Xu, The role of large language models in medical image processing: a narrative review, Quantitative Imaging in Medicine and Surgery 14 (2024) 1108.

[9] M. Hu, S. Pan, Y. Li, X. Yang, Advancing Medical Imaging with Language Models: A Journey from N-grams to ChatGPT, arXiv e-prints (2023) arXiv–2304.

[10] I. Hartsock, G. Rasool, Vision-language models for medical report generation and visual question answering: A review, 2024. arXiv:2403.02469.

[11] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, et al., Gpt-4 technical report, 2024. arXiv:2303.08774.

[12] D. Zhu, J. Chen, X. Shen, X. Li, M. Elhoseiny, Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. arXiv:2304.10592.

[13] BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs, author=Sheng Zhang and Yanbo Xu and Naoto Usuyama and Hanwen Xu and Jaspreet Bagga and Robert Tinn and Sam Preston and Rajesh Rao and Mu Wei and Naveen Valluri and Cliff Wong and Andrea Tupini and Yu Wang and Matt Mazzola and Swadheen Shukla and Lars Liden and Jianfeng Gao and Matthew P. Lungren and Tristan Naumann and Sheng Wang and Hoifung Poon, 2024. arXiv:2303.00915.

[14] S. Zhang, Y. Xu, N. Usuyama, J. Bagga, R. Tinn, S. Preston, R. Rao, M. Wei, N. Valluri, C. Wong, et al., Large-scale domain-specific pretraining for biomedical vision-language processing, arXiv e-prints (2023) arXiv–2303.

[15] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, Y. Zhang, Chatdoctor: A medical chat model fine-tuned on a Large Language Model Meta-AI (LLAMA) using medical domain knowledge, Cureus 15 (2023).

[16] X. Yang, A. Chen, N. PourNejatian, H. C. Shin, K. E. Smith, C. Parisien, C. Compas, C. Martin, A. B. Costa, M. G. Flores, et al., A large language model for electronic health records, NPJ digital medicine 5 (2022) 194.

[17] B. Ionescu, H. Müller, A. Drăgulinescu, J. Rückert, A. Ben Abacha, A. Garcıa Seco de Herrera, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, T. M. G. Pakull, H. Damm, B. Bracke, C. M. Friedrich, A. Andrei, Y. Prokopchuk, D. Karpenka, A. Radzhabov, V. Kovalev, C. Macaire, D. Schwab, B. Lecouteux, E. Esperança-Rodier, W. Yim, Y. Fu, Z. Sun, M. Yetisgen, F. Xia, S. A. Hicks, M. A. Riegler, V. Thambawita, A. Storås, P. Halvorsen, M. Heinrich, J. Kiesel, M. Potthast, B. Stein, Overview of ImageCLEF 2024: Multimedia retrieval in medical applications, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the 15th International Conference of the CLEF Association (CLEF 2024), Springer Lecture Notes in Computer Science LNCS, Grenoble, France, 2024.

[18] J. Rückert, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, C. S. Schmidt, S. Koitka, O. Pelka, A. B. Abacha, A. G. S. de Herrera, H. Müller, P. A. Horn, F. Nensa, C. M. Friedrich, ROCOv2: Radiology Objects in COntext version 2, an updated multimodal image dataset, 2024. URL: https://arxiv.org/abs/2405.10004v1. arXiv:2405.10004.

[19] J. Rückert, A. Ben Abacha, A. G. Seco de Herrera, L. Bloch, R. Brüngel, A. Idrissi-Yaghir, H. Schäfer, B. Bracke, H. Damm, T. M. G. Pakull, C. S. Schmidt, H. Müller, C. M. Friedrich, Overview of ImageCLEFmedical 2024 – Caption Prediction and Concept Detection, in: CLEF2024 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Grenoble, France, 2024.

[20] O. Pelka, S. Koitka, J. Rückert, F. Nensa, C. M. Friedrich, Radiology Objects in Context (ROCO): a multimodal image dataset, in: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3, Springer, 2018, pp. 180–189.

[21] O. Bodenreider, The Unified Medical Language System (UMLS): integrating biomedical terminology, Nucleic acids research 32 (2004) D267–D270.

[22] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023. arXiv:2303.18223.

[23] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, ACM Transactions on Intelligent Systems and Technology 15 (2024) 1–45.

[24] M. A. Manzoor, S. Albarri, Z. Xian, Z. Meng, P. Nakov, S. Liang, Multimodality representation learning: A survey on evolution, pretraining and its applications, ACM Transactions on Multimedia Computing, Communications and Applications 20 (2023) 1–34.

[25] Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, Q. Zhang, M. Chowdhury, M. Zhang, Efficient large language models: A survey, Transactions on Machine Learning Research (2024). URL: https://openreview.net/forum?id=bsCCJHbO8A, survey Certification.

[26] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, G. Wang, Instruction tuning for large language models: A survey, 2024. arXiv:2308.10792.

[27] J. Lu, D. Batra, D. Parikh, S. Lee, ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, Advances in neural information processing systems 32 (2019).
[28] J. Li, D. Li, S. Savarese, S. Hoi, BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in: International conference on machine learning, PMLR, 2023, pp. 19730–19742.
[29] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, S. Hoi, InstructBLIP: Towards general-purpose vision-language models with instruction tuning, Advances in Neural Information Processing Systems 36 (2024).
[30] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, C. A. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, Advances in Neural Information Processing Systems 35 (2022) 1950–1965.
[31] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, N. Collier, On the effectiveness of parameter-efficient fine-tuning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 12799–12807.
[32] C. Li, H. Xu, J. Tian, W. Wang, M. Yan, B. Bi, J. Ye, H. Chen, G. Xu, Z. Cao, J. Zhang, S. Huang, F. Huang, J. Zhou, L. Si, mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections, 2022. arXiv:2205.12005.
[33] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, 2021. arXiv:2106.09685.
[34] Y. Zeng, K. Lee, The Expressive Power of Low-Rank Adaptation, 2024. arXiv:2310.17513.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
[36] NVIDIA, Flash Attention, https://docs.nvidia.com/nemo-framework/user-guide/latest/nemotoolkit/nlp/nemo_megatron/flash_attention.html, 2024. Accessed: 2024-05-28.
[37] H. Liu, C. Li, Y. Li, Y. J. Lee, Improved Baselines with Visual Instruction Tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 26296–26306.
[38] Hugging Face, Hugging Face Transformers Documentation: LLaVA, 2024. URL: https://huggingface.co/docs/transformers/model_doc/llava, hugging Face documentation.
[39] lmsys, Vicuna-7B-V1.3, https://huggingface.co/lmsys/vicuna-7b-v1.3, 2023. Hugging Face model hub.
[40] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, Advances in Neural Information Processing Systems 36 (2024).
[41] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7B, arXiv preprint arXiv:2310.06825 (2023).
[42] M. AI, Mistral-7B-v0.1, https://huggingface.co/mistralai/Mistral-7B-v0.1, 2024. Hugging Face model hub.
[43] Liuhaotian, Llava v1.6 mistral 7b, https://huggingface.co/liuhaotian/llava-v1.6-mistral-7b, 2024.
[44] OpenAI, Clip vit-l/14 model, https://huggingface.co/openai/clip-vit-large-patch14, 2021. Accessed: 2024-05-28.
[45] M. Hoque, M. R. Hasan, Medical image interpretation with large multimodal models, https://github.com/HoqueMahmudul/Medical-Image-Interpretation-with-Large-Multimodal-Models, 2023. Accessed: 2024-06-19.
[46] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. arXiv:1711.05101.
[47] vikhyatk, Moondream2, https://huggingface.co/vikhyatk/moondream2, 2024.
[48] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations,

2020, pp. 38–45.
[49] Y. Li, S. Bubeck, R. Eldan, A. D. Giorno, S. Gunasekar, Y. T. Lee, Textbooks Are All You Need II: phi-1.5 technical report, 2023. arXiv:2309.05463.
[50] X. Zhai, B. Mustafa, A. Kolesnikov, L. Beyer, Sigmoid loss for language image pre-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11975–11986.
[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020. arXiv:2010.11929, https://huggingface.co/google/vit-base-patch16-224.
[52] H. Laurençon, L. Saulnier, L. Tronchon, S. Bekman, A. Singh, A. Lozhkov, T. Wang, S. Karamcheti, A. M. Rush, D. Kiela, M. Cord, V. Sanh, Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023. arXiv:2306.16527.
[53] LAION, CLIP ViT H-14 LAION2B S32B B79K, https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K, 2023. Accessed: 2024-06-19.
[54] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.
[55] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, P. Vajda, Visual transformers: Token-based image representation and processing for computer vision, 2020. arXiv:2006.03677.
[56] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners (2019).
[57] H. Face, transformers.seq2seqtrainer, https://huggingface.co/docs/transformers/main_classes/trainer#transformers.Seq2SeqTrainer, 2023. Accessed: 2024-05-28.
[58] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, et al., CheXNet: Radiologist-level pneumonia detection on chest x-rays with deep learning, arXiv preprint arXiv:1711.05225 (2017).
[59] F. Chollet, Image captioning, https://keras.io/examples/vision/image_captioning/, 2023. Accessed: 2024-05-28.
[60] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, 2020. arXiv:1904.09675.
[61] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.
[62] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
[63] T. Sellam, D. Das, A. P. Parikh, BLEURT: Learning Robust Metrics for Text Generation, 2020. arXiv:2004.04696.
[64] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005, pp. 65–72.
[65] R. Vedantam, C. L. Zitnick, D. Parikh, CIDEr: Consensus-based Image Description Evaluation, 2015. arXiv:1411.5726.
[66] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, Y. Choi, CLIPScore: A Reference-free Evaluation Metric for Image Captioning, 2022. arXiv:2104.08718.
[67] A. B. Abacha, W. wai Yim, G. Michalopoulos, T. Lin, An Investigation of Evaluation Metrics for Automated Medical Note Generation, 2023. arXiv:2305.17364.
[68] A. Trockman, J. Z. Kolter, Patches are all you need?, 2022. arXiv:2201.09792.
[69] Keras, ConvMixer example, https://keras.io/examples/vision/convmixer/, 2023. Accessed: 2024-05-28.
[70] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner,

D. Keysers, J. Uszkoreit, et al., Mlp-mixer: An all-mlp architecture for vision, Advances in neural information processing systems 34 (2021) 24261–24272.

A. Specifications of the Computational Environment

The specifications of the utilised computational resources and environment included two machines. The details are as follows.

- Machine 1
  - Machine Type: a2-highgpu-2g (Accelerator Optimized: 2 NVIDIA Tesla A100 GPUs, 24 vCPUs, 170GB RAM)
  - GPU: NVIDIA A100-40GB x 2
  - Booting Disk: 1000 GB SSD
  - Data Disk: 1000 GB SSD
  - Language: Python 3.12.x
- Machine 2
  - Machine Type: n1-highmem-16 (16 vCPUs, 104 GB RAM)
  - GPU: NVIDIA V100 x 2
  - Boot disk: 150 GB SSD
  - Data disk: 1000 GB SSD
  - Language: Python 3.12.x
  - Frameworks: PyTorch 2.x and Tensorflow 2.16.x

B. GitHub Repository

[45] provides the link to GitHub repository which is publicly available for accessing the reproducible codes relevant to the submissions made for this competition.