Multimodal Fusion: Advancing Medical Visual Question Answering

Anjali Mudgal[1†], Udbhav Kush[1*†], Aditya Kumar[1†], Amir Jafari[1]

[1]Department of Data Science, The George Washington University, Washington DC, USA.

*Corresponding author(s). E-mail(s): ukush4@gwu.edu;
Contributing authors: amudgal26@gwu.edu; aditya_kumar@gwu.edu; ajafari@gwu.edu;
†These authors contributed equally to this work.

Abstract

This paper explores the application of Visual Question Answering (VQA) technology, which combines computer vision and natural language processing, in the medical domain, specifically for analyzing radiology scans. VQA can facilitate medical decision-making and improve patient outcomes by accurately interpreting medical imaging, which requires specialized expertise and time. The paper proposes developing an advanced VQA system for medical datasets using the Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation (BLIP) architecture from Salesforce, leveraging deep learning and transfer learning techniques to handle the unique challenges of medical/radiology images. The paper discusses the underlying concepts, methodologies, and results of applying the BLIP architecture and fine-tuning approaches for Visual Question Answering in the medical domain, highlighting their effectiveness in addressing the complexities of VQA tasks for radiology scans. Inspired by the BLIP architecture from Salesforce, we propose a novel multimodal fusion approach for medical visual question answering and evaluating its promising potential.

Keywords: Medical Visual Question Answering (VQA), Medical image, Multimodal Transformer, Vision Language Model, Magnetic Resonance Imaging (MRI), Computed Tomography Scan (CT Scan)

1 Introduction

Recent years have seen remarkable progress in computer vision and natural language processing, enabling AI systems to tackle multimodal tasks requiring a joint understanding of visual and textual data. One such challenging task is visual question answering (VQA), where a system must accurately answer questions posed in natural language about the content of a given image. It involves understanding the content of the image and correlating it with the context of the question asked. Because we need to compare the semantics of information present in both modalities — the image and natural language question related to it — VQA entails a wide range of sub-problems in both Computer Vision (CV) and Natural Language Processing (NLP) (such as object detection and recognition, scene classification, counting, and so on). Thus, it is considered an AI-complete task [1].

Inspired by the success of VQA in general domains, there is growing interest in developing VQA capabilities specifically for the medical field - an area known as medical visual question answering (Med-VQA). In Med-VQA, the system receives a medical image, typically a radiology scan, along with a question in natural language, and it must provide a relevant answer by reasoning over the visual and textual inputs. While promising initial advances has been made, as highlighted by the introduction of the ImageCLEF[2] Med-VQA challenge in 2018, the Med-VQA field is still nascent and significant research is needed before such systems can be reliably deployed in real clinical settings. The medical field has undergone a data revolution, with electronic health records (EHRs) providing patients unprecedented access to their own medical data and imaging records. This increased accessibility enables patients to review their medical information independently, outside of formal consultations with healthcare professionals. However, this raises the need for intuitive methods that allow patients to understand and gain insights from their complex medical data without expert oversight. While patients could consult doctors, this is often impractical due to time and financial constraints. Alternatively, they may turn to general search engines or conversational AI, but risk receiving misleading or inaccurate information. To bridge this gap, Med-VQA systems could prove invaluable by allowing patients to pose natural language questions about their medical images and receive reliable answers in an accessible format.

A significant challenge faced is the limited availability of large-scale annotated Med-VQA datasets, which can impede effective learning of patterns by models. As a potential solution, pretraining and fine-tuning strategies play a crucial role in boosting performance by providing a strong initial starting point and domain-specific feature representation. Nonetheless, medical data exhibits unique characteristics that differentiate it from general domains. We hypothesize that pretraining and fine-tuning portions of a larger architecture specifically on medical data could yield performance benefits for Med-VQA by imbuing them with domain-specific knowledge more directly applicable to this specialized task.

In this research paper, we delve into the application of VQA technology to medical scans, focusing primarily on the BLIP architecture developed by Salesforce. We make the following contributions:

1. We combine the ImageCLEF dataset with the VQA-RAD[3] dataset which are part of a common database known as the MedPix database. We remove redundant images and retain unique scans and associated question-answer pairs. Further, we perform simple data augmentation that is appropriate for such scans to create a larger database for Med-VQA. Details of these datasets and augmentations are provided in later sections.
2. We utilize BLIP architecture from Salesforce for Med-VQA and perform answer generation as opposed to classification for Med-VQA. We further dissect this architecture, fine-tuning specific components to enhance performance.
3. Lastly, we propose a new transformer-based architecture, taking inspiration from BLIP, tailored specifically for medical image analysis. To do this, we created our own medical tokenizer based on the training data.

The rest of the paper is organized as follows: in section 2 relevant background is briefly reviewed; in section 3 details about the methodology are provided; in section 4 experimental results are presented. Finally, section 5 concludes this paper.

2 Background

2.1 Literature review of Visual Question Answering

Visual Question Answering (VQA) is a task that combines computer vision and natural language processing (NLP). In this task, a computer is shown an image and a question about that image in natural language. The goal is for the computer to correctly answer the question based on the information in the image [4][5][6][7]. There are different variations of VQA:

1. Binary (yes/no) VQA: The computer must answer the question with either "yes" or "no" [4][5].
2. Multiple-choice VQA: The computer is given a set of answers and must choose the correct one [4][6].
3. Fill-in-the-blank VQA: The computer is given a statement about the image with one or more blanks, and it must fill in the missing words to complete the statement correctly [7].

Joint embedding approaches, inspired by advancements in deep neural networks for both computer vision and natural language processing (CNNs and RNNs), aim to learn embeddings of images and sentences in a shared feature space. This enables feeding them into a classifier together for predicting answers ([8][9]). Attention mechanisms, building upon this concept ([9][10][6]), further refine the process by concentrating on specific parts of the input, drawing inspiration from successful implementations of image captioning ([11]). Multimodal models can be of various forms to capture information from the text and image modalities, along with some cross-modal interaction as well. In fusion models, the information from the text and image encoders are fused into a combined representation to perform the downstream task.

A typical fusion model for a VQA system involves the following steps:

1. Featurization of image and question: We need to extract features from the image and obtain the embeddings of the question after tokenization. The question can be featurized using simple embeddings (like GLoVe), Seq2Seq models (like LSTMs), or transformers. Similarly, the image features can be extracted using simple CNNs (convolutional neural networks), early layers of object detection or image classification models, or image transformers.
2. Feature fusion: Since VQA involves a comparison of the semantic information present in the image and the question, there is a need to jointly represent the features from both modalities. This is usually accomplished through a fusion layer that allows cross-modal interaction between image and text features to generate a fused multimodal representation.
3. Answer generation: Depending on the modeling of the VQA task, the correct answers could either be generated purely using natural language generation (for longish or descriptive answers) or using a simple classifier model (for one-word/phrase answers present in a fixed answer space).

The simple classifier approach for answer generation is inherently limited by its fixed answer space, restricting flexibility. Therefore, we explore decoder models that can generate free-form natural language answers, more suitable for complex medical reasoning. Additionally, we investigate late fusion techniques, as we believe separate bulky encoders can better capture rich representations from medical images and text before fusing them, compared to fusing from simple encoders. We hypothesize this late, powerful multimodal fusion can enhance Med-VQA performance.

2.2 BLIP Architecture

BLIP (Bootstrapping Language-Image Pre-training) [12] is a pre-training framework for unified vision-language understanding and generation, which achieves state-of-the-art results on a wide range of vision-language tasks.

Vision-language pre-training has emerged as an effective approach, where deep neural network models are pre-trained on large scale image-text datasets to improve performance on downstream vision-language tasks, such as image-text retrieval, image captioning, and visual question answering [13]. BLIP introduces a novel architecture called Multimodal mixture of Encoder-Decoder (MED) for effective vision-language pre-training.

To pre-train a unified vision-language model capable of both understanding and generation, BLIP introduces a multimodal mixture of encoder-decoder, a multi-task model that operates in three functionalities:

1. Unimodal encoders, which separately encode image and text. The model uses Vision Transformer ViT [14] which divides the input image into patches and encodes them as a sequence of embedding with addition to [CLS] token to represent the globe image feature. The text encoder is the same as BERT i.e., Mask Language Model [15] with a [CLS] token to append the beginning of the text input to summarize the sentence.
2. Image-grounded text encoder, which injects visual information by inserting a cross-attention layer between the self-attention layer and the feed forward network

for each transformer block of the text encoder. A task-specific [Encode] token is appended to the text, and the output embedding of [Encode] is used as the multimodal representation of the image-text pair [13].
3. Image-grounded text decoder, which replaces the bi-directional self-attention layers in the text encoder with causal self-attention layers. A special [Decode] token is used to signal the beginning of a sequence [13].

BLIP jointly optimizes three objectives during pre-training, with two understanding-based objectives (ITC, ITM) and one generation-based objective (LM):

1. Image-Text Contrastive Loss (ITC) activates the unimodal encoder. It aims to align the feature space of the visual transformer and the text transformer by encouraging positive image-text pairs to have similar representations in contrast to the negative pairs [13].
2. Image-Text Matching Loss (ITM) activates the image-grounded text encoder. ITM is a binary classification task, where the model is asked to predict whether an image-text pair is positive (matched) or negative (unmatched) given their multimodal feature [13].
3. Language Modeling Loss (LM) activates the image-grounded text decoder, which aims to generate textual descriptions conditioned on the images [13].

To perform efficient pre-training while leveraging multi-task learning as explained above:

1. The text encoder and text decoder share all parameters except for the self-attention layers. The reason is that the differences between the encoding and decoding tasks are best captured by the self-attention layers.
2. The encoder employs bi-directional self-attention to build representations for the current input tokens,
3. while the decoder employs causal self-attention to predict the next tokens.

For the task of Visual Question Answering (VQA), BLIP formulates it as an answer generation problem. During fine-tuning, the pre-trained model encodes an image-question pair into multimodal embeddings using the image-grounded text encoder, which are then passed to the answer decoder to generate the answer. The VQA model is fine-tuned using the language modeling (LM) loss with ground-truth answers as targets [12].

BLIP achieves state-of-the-art performance on the VQA task, outperforming models like ALBEF [16] and SimVLM [17], even when using significantly less training data. The success of BLIP is attributed to its effective MED architecture and joint pre-training approach, which enables seamless transfer learning to downstream tasks like VQA.

The BLIP model architecture consists of layers that process both visual and textual inputs in a unified manner.

- Visual Processing: Pass the visual features extracted from the images through a visual encoder network, which captures the visual information.

- Textual Processing: Process the encoded textual data through a textual encoder network, which understands the semantic meaning of the text.
- Fusion: Combine the visual and textual representations using fusion mechanisms such as attention mechanisms or multi-modal fusion layers. This step aims to create a joint representation that captures the correlation between the visual and textual modalities.

3 Methodology

3.1 Dataset

Our research capitalizes on a unified dataset derived from two closely related sources: ImageCLEF 2019 and VQA-RAD. Both ImageCLEF 2019 and VQA-RAD datasets serve as pillars of our investigation, offering a curated compilation of radiology images coupled with corresponding question-answer pairs. Despite their separate identities, these datasets share fundamental characteristics, focusing on radiological examinations and encompassing a spectrum of anatomical structures and medical scenarios. These datasets are characterized by their emphasis on four primary question types: Modality, Plane, Organ system, and Abnormality. Through this categorization, it offers a structured approach to querying medical images, enhancing interpretability, and facilitating targeted investigations.

Recognizing the intrinsic similarities between ImageCLEF2019 and VQA-RAD, we adopted a strategy of dataset consolidation to streamline our analysis. By merging the pertinent components of both datasets, we constructed a cohesive corpus tailored to our research objectives. This consolidation process facilitated a more coherent and comprehensive examination of medical image understanding and interpretation. Also, to ensure data integrity, we removed redundant images and question answer pairs from our combined dataset. After consolidating the datasets, our combined corpus comprises 14,590 question-answer pairs in the training set, 2,000 pairs in the validation set, and 500 pairs in the test set.

A sample from the ImageCLEF 2019 dataset:

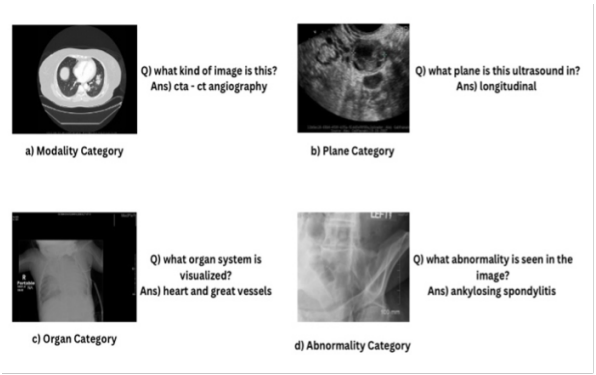

Fig. 1 Representative Image Samples from ImageCLEF

3.1.1 Augmentation

Due to the limited number of original medical scans available in our dataset, we employed image augmentation techniques to expand our dataset and enhance the quality of image embeddings. Image augmentation involves applying transformations to the original images to create new variations while preserving their semantic content.

For our augmentation process, we applied transformations such as blur, brightness adjustment, and noise addition to each original image. These transformations help to introduce variability in the dataset, which can improve the robustness and generalization ability of our machine learning models trained on the augmented data.

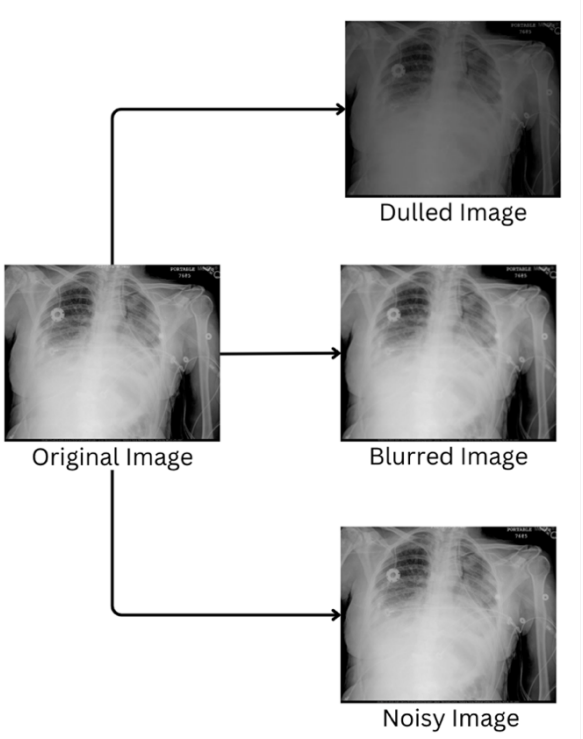

Fig. 2 Augmentation of the Images from the Train Dataset Sample

To expand our training set, we applied augmentation techniques to each of the 3,200 original images, resulting in the creation of three additional images per original image through blur, brightness adjustment, and noise addition. This process yielded an additional 9,600 images, effectively tripling the size of our training dataset. The final size of our training set is 24,190 question-answer pairs, validation set is 2,000 question-answer pairs and test set are 500 question-answer pairs.

3.2 BLIP Fine-Tuning

To start the analysis, we used the BLIP pre-trained model for question-answering from HuggingFace to run an evaluation loop or inference on the test data created while preprocessing the ImageCLEF and VQA-RAD datasets. The corresponding BLIP processor is utilized from HuggingFace on the image-question and answer pair. The BLIP processor wraps a BERT tokenizer and BLIP image processor into a single processor. The image processor performs preprocessing on our medical images according to the BLIP pre-training transformations and BERT tokenizer converts the questions and answers to corresponding token ids according to the vocabulary of this tokenizer. A batch size of 12 is used to run this inference. With this we benchmark the performance of BLIP pre-trained model, that has been trained on COCO dataset and Visual Genome datasets [18], on medical radiology scans. This serves as the baseline performance of BLIP pre-trained Visual Question Answering architecture for medical domain. The architectural flow of image-question in BLIP for VQA has been discussed in section 2.2. The performance metrics used for the Visual Question Answering tasks are BLEU score [19] and ROUGE score [20] which have been noted down in the results section below.

After this, we fine-tuned BLIP for question-answering on our medical dataset. All the model weights and biases are unfrozen to achieve this. We use the 24,190 question-answer pairs in the training set, and 2,000 pairs in the validation set to perform this fine-tuning as discussed in section 3.1. This fine-tuning was performed for a batch size of 12, and for 15 epochs. However, the model parameters seem to converge within the first 5 epochs when saved on the validation cross-entropy loss which is the Language Model loss discussed in section 2.2. The image-question and answer follow the same processor steps that have been described above. AdamW [21] is used as the optimizer with a learning rate of 4e-5 along with an exponential learning rate scheduler with a gamma value of 0.9. After the best model was saved, a similar inference or evaluation loop as above is run on the test data to retrieve the benchmark metrics for this fine-tuned BLIP model. The results and discussion are noted down below.

3.3 BLIP Vision Encoder – Incorporating Medical Domain Knowledge

We observed that the process of fine-tuning the entire BLIP architecture for Visual Question Answering in medical domain was a time-consuming process where each epoch took 1 hour to run on NVIDIA A10G Tensor Core GPUs. We attempted to dissect this architecture to see whether we can perform enhancements. We first dissected the vision model and observed the different components – a patch embedding convolution layer to divide images in patches and retain patch positional encoding and the transformer encoder to create image embeddings. We decided to customize the pre-trained BLIP vision model to our specific medical task and assess its impact. The size of the convolution layer was - Conv2d(3, 768, kernel size=(16, 16), stride=(16, 16)). We created a custom autoencoder architecture starting with the same dimension as this convolution patch embedding layer. A detailed representation of the custom

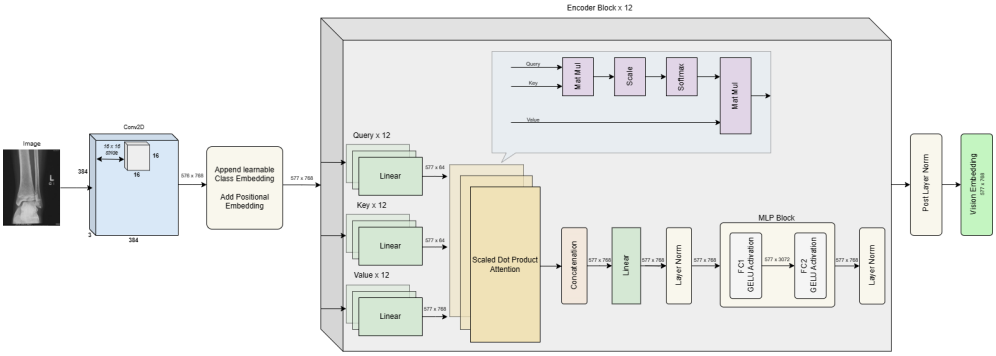

Fig. 3 BLIP Vision Encoder

autoencoder architecture is given in figure 8. The purpose of this autoencoder architecture is to learn a hidden representation of all the medical radiology images by minimizing the error of reconstruction. We hypothesized that after training the custom autoencoder, the first layer of this custom architecture, which is the same size as the convolution patch embedding layer of the vision model, will learn better weights and biases that are customized to medical images. We train our custom autoencoder architecture with complete medical images, for a batch size of 64 and 10 epochs. The optimizer and scheduler are utilized as described above. We use the mean square error loss between the reconstructed and original images passed as both the input and as target. The learned weights are retrieved from the first layer of the custom autoencoder and are plugged into the BLIP vision model. To assess the impact of performing this specialized fine-tuning on a component of the vision model, we run an inference or evaluation loop using BLIP model for question-answering initialized with the new convolution patch embedding weights. The results are discussed in the section below. As the next step, we fine-tune the entire BLIP model for question-answering again using the same specifications that are discussed above, with the new addition of convolution patch embedding layer in the vision model initialized with learned weights from the custom autoencoder. This is our hypothesized contribution to improve results of the fine-tuned BLIP model for question-answering on a relatively small medical dataset. We propose fine-tuning the patch embedding layer of vision model on medical images to learn customized weights. We then apply this learned weight to the base BLIP architecture for question-answering, freezing the parameters for convolution patch embedding layer of the vision model and finally fine-tuning the remaining BLIP architecture with the medical dataset. This fine-tuning has been performed for a batch size of 12 for 10 epochs.

3.4 Modified BLIP for Med-VQA- Proposed Architecture

Inspired by the BLIP architecture, which fuses text and vision embeddings using cross-attention, we propose a modified approach for Visual Question Answering in the medical domain. Our architecture leverages the pre-trained BLIP vision encoder

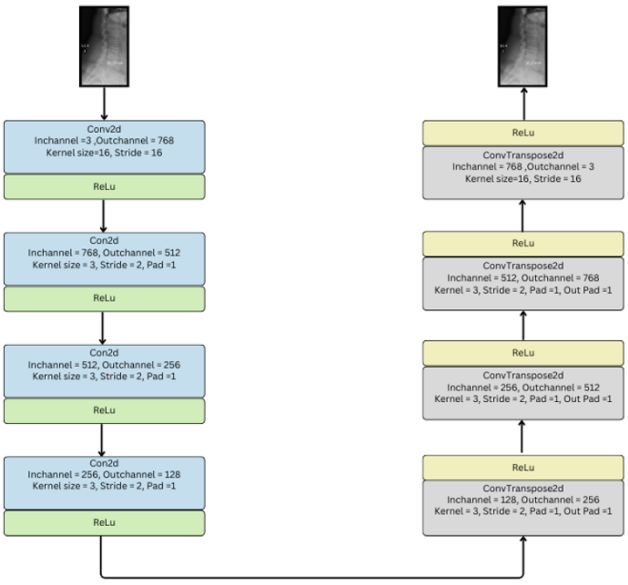

Fig. 4 Autoencoder Model to Finetune the Vision Model of BLIP VQA

to generate image embeddings and a custom transformer encoder architecture is pre-trained on our medical dataset to encode question embeddings.

These embeddings are concatenated and down-sampled using a linear layer, forming a rich fused representation. The resulting embedding serves as input to the decoder, enabling the generation of accurate answers based on the combined visual and textual information. We evaluate our approach and discuss the results, highlighting the potential of this fusion technique for enhancing Visual Question Answering performance in the medical domain.

3.4.1 Image Embedding

The BLIP (Bootstrapping Language-Image Pre-training) vision model processes images through a series of steps. First, the BLIP vision processor is used to preprocess all the images, resizing them to a fixed size of 368 x 368 pixels. The preprocessed images are then passed through a convolutional layer with a kernel size of 16 and a stride of 16, resulting in a vision embedding of size 576.

The size of the output is calculated using the convolutional formula:

$$n_{\text{out}} = \left\lfloor \frac{n_{\text{in}} - K + 2p}{S} \right\rfloor + 1 \tag{1}$$

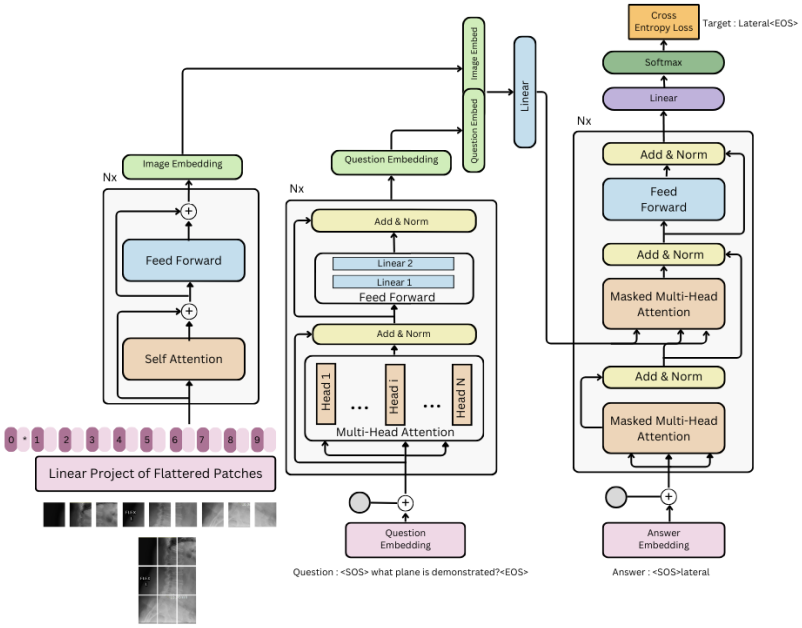

Fig. 5 Proposed Architecture

where n_{out} is the number of output channels, n_{in} is the number of input channels which is 3 in our case as we are dealing with an RGB image, K is the kernel size (16 in our case), p is the padding (0 in our case), and S is the stride (16 in our case).

To capture global information, an extra embedding is added at the start of the vision embedding, and positional embeddings are also incorporated. After adding the extra embedding and positional embeddings, the size of the embedding becomes 577. The BLIP vision model uses a dmodel (dimensionality of the model) of 768, projecting each patch in the vision embedding to this dimension. The final output of the BLIP vision model has a shape of (Batch_size, 577, 768), representing the encoded visual features of the input images. These rich visual embeddings capture both local patch-level information and global context.

3.4.2 Question and answer Embedding

Tokenization

To enhance the richness of the medical question embeddings, a custom vocabulary and tokenizer are constructed using the ImageCLEF medical dataset. This process involves combining both the questions and answers from the training data set. A Byte Pair Encoding (BPE) tokenizer is employed to create a specialized medical vocabulary through sub-word tokenization. BPE algorithmically identifies and combines frequently occurring sub-word units, enabling the effective handling of domain-specific terms and expressions prevalent in medical terminology. Additionally, special tokens

such as PAD (padding), SOS (start of sentence), EOS (end of sentence), UNK (unknown), MASK (masked token), and SEP (separator) are incorporated into the vocabulary. Despite the limited size of the medical dataset, this approach facilitates a more comprehensive understanding of medical language by capturing the nuances and intricacies specific to the domain. By leveraging this custom vocabulary and tokenization technique, the model is equipped to process and represent medical questions with greater precision and contextual understanding.

Encoding

The Transformer[22] Question Encoder plays a vital role in the proposed model for medical question answering. Its purpose is to process the input question and generate a dense vector representation that captures the semantic information of the question. To ensure compatibility with the visual embeddings, the question sequence length is fixed at 577, and the dimensionality of the model (d_{model}) is set to 768.

The encoding process begins by adding special tokens to the question. The SOS (start of sentence) token is added to the beginning of the question, and the EOS (end of sentence) token is appended to the end. These special tokens help the model identify the boundaries of the question and provide a clear starting and ending point for the encoding process.

Once the special tokens are added, the question tokens are passed through an embedding layer called InputEmbeddings. This layer converts each token into a dense word embedding, transforming the discrete token representations into continuous vector representations. The word embeddings capture the semantic and syntactic information of the tokens, allowing the model to understand the meaning and relationships between the words in the question. To scale the embeddings as suggested in the original Transformer paper, the embeddings are multiplied by the square root of d_{model}.

Next, positional encodings are added to the word embeddings using the PositionalEncoding module. Positional encoding is used in the Transformer architecture to inject sequence-wise information. The position encoding layer employs a combination of sinusoidal and cosine functions to create a geometric sequence of values.

For even positions in the range $[0, d_{\text{model}}]$, a sinusoidal function is used:

$$PE(pos, 2i) = \sin(pos \times div\_term) \tag{2}$$

For odd positions, a cosine function is used:

$$PE(pos, 2i + 1) = \cos(pos \times div\_term) \tag{3}$$

Here, pos is the position index, i is an even integer in $[0, d_{\text{model}}]$, and div_term is calculated as:

$$div\_term = exp(-i * (log(10000)/d_{\text{model}})) \tag{4}$$

The div_term creates a geometric sequence that decreases exponentially as i increases, with the rate of decrease controlled by $log(10000)/d_{\text{model}}$.

The alternating sine and cosine functions create unique embeddings for each position, allowing the model to distinguish between even and odd positions. These

positional encoding values are added element-wise to the input embeddings, providing the Transformer with positional information to effectively learn and attend to the relative positions of tokens within the sequence.

After adding the positional encodings, the question embeddings are passed through the Transformer encoder. The Transformer encoder consists of multiple layers, each containing a multi-head self-attention mechanism and a feed-forward neural network.

- The input embeddings and positional embeddings are added together and passed as query (Q), key (K), and value (V) matrices into the Multi-Head Attention block.
- Each head (h=8 in this case) has its own learned weight matrices: W_{q}, W_{k}, and W_{v}. The Q, K, and V matrices are multiplied with their respective weight matrices to obtain Q', K', and V' for each head.
- The dimensionality of each head is reduced to $d_{\mathrm{k}} = d_{\mathrm{model}}/h$ ($768/8 = 96$ in this case), allowing each head to focus on a specific part of the embedding.
- For each head, the attention scores are computed using the scaled dot-product attention: $head(i) = Attention(QW_i^q, KW_i^k, VW_i^v)$

$$\mathrm{Attention}\,(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

- The attention scores are scaled by dividing them by the square root of d_{model} to prevent the dot products from growing too large.
- The softmax function is applied to the scaled attention scores to obtain the attention weights, which are then multiplied with the value matrix (V) to get the weighted values for each head.
- The outputs from all heads are concatenated together:

$$\mathrm{Multihead}\,(Q, K, V) = \mathrm{Concat}(head_1, head_2, ..., head_{\mathrm{h}})W^0 \tag{6}$$

- The concatenated output is then multiplied with the W^0 matrix to obtain the final output of the Multi-Head Attention block. By using multiple heads, the Multi-Head Attention mechanism allows the model to attend to different aspects of the input embeddings simultaneously, capturing various relationships and dependencies within the sequence.

The output of the Transformer encoder is the final question embedding, which has a size of (Batch, 577, 768). This embedding size matches the dimensions of the visual embeddings, ensuring a consistent representation across both modalities. By aligning the question and visual embeddings in terms of sequence length and dimensionality, the model can effectively process and integrate the information from both sources in the subsequent stages. The consistent representation achieved by the Transformer Question Encoder facilitates seamless interaction between the question and visual information. It allows the model to attend to relevant parts of the question and visual features simultaneously, enabling it to reason and generate accurate answers based on the combined context. In summary, the Transformer Question Encoder is a critical component in the proposed model for medical question answering. It processes the input question, incorporates positional information, and generates a dense vector

representation that captures the semantic information of the question. By ensuring consistency with the visual embeddings, the question encoder enables effective integration and interaction between the question and visual modalities, facilitating accurate and informed answering of medical questions.

Fusion

The concatenation of the image embedding and question embedding results in a fused embedding that combines information from both modalities. We denote the image embedding as E_{img} and the question embedding as E_{q}. The fused embedding E_{fused} can be represented as:

$$E_{\text{fused}} = \text{Concat}\left(E_{\text{img}}, E_q\right) \tag{7}$$

where $Concat()$ represents the concatenation operation along the last dimension.

The shape of the fused embedding is (Batch, 577, 768*2), as it combines the image and question embeddings, each of shape (Batch, 577, 768). To ensure proper fusion and compatibility with the subsequent layers, a linear projection layer is applied to the fused embedding. This layer transforms the fused embedding into the desired shape of (Batch, 577, 768). We denote the linear projection layer as L_{proj}. The projected fused embedding E_{proj} can be expressed as:

$$E_{\text{proj}} = \text{L}_{\text{proj}}\left(E_{\text{fused}}\right) \tag{8}$$

where L_{proj} is a linear transformation that maps the fused embedding to the desired size.

The projected fused embedding E_{proj} is then passed to the decoder as the key (K) and value (V) pair for the attention mechanism. The query (Q) for the decoder is obtained from the input to the decoder after applying causal self-attention. In the decoder, the input is the answer sequence, and an SOS (start of sentence) token is appended at the start of the answer to predict the next token. The decoder uses causal attention, also known as masked self-attention, which allows each token to attend only to the previous tokens in the sequence. The attention mechanism in the decoder can be represented by the following equation:

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{9}$$

$$d_k \ = \frac{d_{model}}{number\ of\ heads} \tag{10}$$

$$numberofheads = 8 \tag{11}$$

where Q represents the query matrix derived from the decoder input after causal self-attention, with shape as (Batch, sequence length of answer, d_{model}), K represents the key matrix, which is the projected fused embedding E_{proj}, with shape (Batch, 577, d_{model})), V represents the value matrix, which is also the projected fused embedding E_{proj}, with size (Batch, 577, d_{model}). The sequence length of the answer is the maximum number of tokens in the answer, 200 in our case. The attention weights are

computed by taking the dot product of the query and key matrices, scaling it by square root of d_k, and applying a softmax function. The resulting attention weights are then used to compute a weighted sum of the value vectors, generating the attended output.

Decoder

By using the projected fused embedding as the key and value pair, the decoder attends to the relevant information from both the image and question embeddings to generate the answer sequence. The causal attention mask ensures that the decoder can only attend to the previous tokens in the answer sequence, enabling auto-regressive generation of the answer. To train the model, the widely adopted cross-entropy loss function is employed, which measures the dissimilarity between the predicted and actual probability distributions of the answer tokens. This loss function guides the model to minimize the discrepancy between its predictions and the ground truth answers. Furthermore, the BLEU score, a well-established metric in natural language processing, is utilized to evaluate the quality of the generated answers. BLEU assesses the similarity between the generated answers and the reference answers, considering factors such as n-gram precision and brevity penalty.

The proposed approach offers a novel method for generating rich embeddings that capture the essence of both the image and the question. By treating these embeddings separately and considering their relationship to the answer, the model can effectively leverage the information from both modalities to generate accurate and contextually relevant responses.

In conclusion, the proposed approach for Visual Question Answering in the medical domain introduces a novel fusion technique that combines image and question embeddings to generate accurate and contextually relevant answers. By leveraging the BLIP vision encoder for image embedding generation and pre-training a custom transformer encoder for question embedding encoding, the model captures rich visual and textual information. The concatenation and down sampling of these embeddings using a linear layer form a powerful fused representation that serves as input to the decoder. The decoder, equipped with causal attention, attends to the relevant information from the fused embedding to generate answer sequences. The incorporation of custom vocabulary through the BPE tokenizer further enhances the model's ability to handle domain-specific medical terminology. The evaluation using cross-entropy loss and BLEU score demonstrates the effectiveness of this approach in generating accurate answers. Overall, this innovative framework holds significant potential for improving the efficiency and performance of medical question-answering systems, ultimately benefiting healthcare professionals and patients by providing precise and reliable information.

4 Experiments and Results

In this section, we demonstrate BLIP's effectiveness on Visual Question Answering (VQA) on medical datasets such as X-rays, MRI scans, etc. by using a pre-trained base BLIP model and fine-tuning it in multiple ways to see its performance. The key

idea here is to achieve good results for a low fine-tuning time and computer resources. Detailed experimental setups are given in the next section.

4.1 Experimentation set-up

4.1.1 Backbone model

We use the BLIP Model as our base models. This model takes images and text as input and generates text as output. For display equations (with auto generated equation numbers) one can use the equation or align environments:

4.2 Metrics

Two primary metrics are used to assess the performance of fine-tuned BLIP models on VQA on medical dataset. The Bilingual Evaluation Understudy (BLEU) score [17] and the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score [18].

4.2.1 BLEU

The (BLEU) score is a metric used to evaluate the quality of machine-generated text by comparing it to reference texts. In our research paper, we calculate the average BLEU score to assess the performance of our Visual Question Answering model in generating accurate and relevant answers.

The average BLEU score provides an overall assessment of the model's performance in generating accurate answers. By incorporating the BLEU score as an evaluation metric, we can quantitatively measure the effectiveness of our Visual Question Answering model in generating answers that closely match the ground truth answers.

Based on the BLEU paper [17], the sentence-level BLEU score is calculated as follows:

$$\text{BLEU} = \text{BP} \times \exp\left(\sum_{n=1}^{N} w_n \times \log(p_n)\right) \tag{12}$$

where: BP is the brevity penalty, which is calculated as:

$$\text{BP} = \begin{cases} 1 & \text{if output length} \geq \text{reference length} \\ \exp(1 - \text{reference length/output length}) & \text{otherwise} \end{cases} \tag{13}$$

where,

N is the maximum n-gram order (typically 4)

w_{n} is the weight assigned to each n-gram precision (typically uniform weights, i.e., 1/N)

p_{n} is the modified n-gram precision, calculated as:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-\text{gram} \in C} \text{Count}_{\text{clip}}\,(n\text{−gram})}{\sum_{C' \in \{Candidates\}} \sum_{n-\text{gram}' \in C'} \text{Count}\,(n\text{−gram}')} \tag{14}$$

The sentence-level BLEU score is calculated by taking the geometric mean of the modified n-gram precisions and multiplying it by the brevity penalty factor. The

brevity penalty penalizes candidate translations that are shorter than the reference translations, while the modified n-gram precision accounts for both the adequacy and fluency of the translation.

4.2.2 ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used to evaluate the quality of summaries or machine-generated text by comparing them to reference summaries or gold-standard texts. ROUGE measures the overlap of n-grams, word sequences, and word pairs between the generated summary and the reference summaries.

ROUGE-L: It measures the longest common subsequence (LCS) between the generated summary and the reference summaries. The LCS is the longest sequence of words that appear in both the generated summary and the reference summaries, allowing for gaps ROUGE scores range from 0 to 1, with higher scores indicating better overlap between the generated summary and the reference summaries. ROUGE is widely used in the evaluation of summarization tasks, machine translation, and other natural language generation tasks.

4.3 Results

- **Base Blip** - This is the pre-trained BLIP model that is fetched from Hugging Face. The pre-training details are available in previous sections. This pre-trained model is used to run an inference on our dataset to determine a benchmark performance.
- **Fine-tuned BLIP - end-to-end** - This is the fine-tuned BLIP model on our dataset. This fine-tuning code is run for a batch size of 12, and for 15 epochs. However, the model parameters seem to converge within the first 5 epochs when saved on the validation cross-entropy loss. We observe a significant boost up in BLEU and ROUGE scores when the base BLIP model is fine-tuned on medical dataset.
- **Convolution Patch Embedding Fine-tuned + Base BLIP** - This is our first contribution in dissecting the BLIP model. As discussed previously, we have taken the first convolution patch embedding layer from the BLIP vision encoder and recreated a custom autoencoder architecture based on the dimensions of the patch embedding layer. We have fine-tuned this custom architecture on our complete medical image dataset, to achieve a better representation of this layer customized to medical application. The first layer of this fine-tuned layer matches the dimensions of the convolution patch embedding layer. The trained weights for this layer are retrieved from the custom autoencoder and the weights are plugged into the base BLIP model. This training has been performed for a batch size of 64 and 10 epochs. The training time is significantly faster. To assess the impact of performing this specialized fine-tuning, we run an inference of the base BLIP model initialized with these convolution patch embedding layer weights. We see a boost in the benchmark BLEU and ROUGE score highlighting the effectiveness of fine-tuning only a section of the BLIP architecture.

- **Convolution Patch Embedding Fine-tuned + Fine-tuned BLIP model** - This model version is an extension to the previous version. This is our hypothesized contribution to improve results of the fine-tuned BLIP model on a relatively small medical dataset. We have proposed fine-tuning the patch embedding layer of the vision encoder on medical images to get a good medical image representation, applying this learned weight representation to the base BLIP architecture, freezing the parameters for this specific layer and finally fine-tuning the remaining BLIP architecture with the medical dataset. We see a boost in our scores when we apply this form of training as opposed to just fine-tuning.
- **Proposed Architecture** - After experimenting with fine-tuning specific components, we explored a different approach for combining the vision embeddings with a question-answer transformer. Instead of using the standard methods, we utilized a technique to integrate the visual and question encodings, aiming to generate more accurate and contextually relevant answers. By leveraging our custom tokenization strategy, specifically designed for the question-answer text in the medical dataset, we observed a significant improvement in the model's performance. This combination of vision and question embeddings, coupled with our domain-specific tokenization, resulted in a substantial increase in the BLEU and ROUGE scores, reaching 0.41 and 0.44 respectively. These results highlight the effectiveness of our approach in enhancing the model's ability to generate precise and coherent answers in the medical domain, demonstrating the potential of this method for improving visual question-answering systems in specialized domains.

S no.	Model Version	BLEU score	ROUGE score
1.	Base BLIP	0.12	0.15
2.	Fine-tuned BLIP - end-to-end	0.37	0.40
3.	Convolution Patch Embedding Fine-tuned + Base BLIP	0.13	0.17
4.	Convolution Patch Embedding Fine-tuned + Fine-tuned BLIP model	0.38	0.42
5.	Proposed Architecture	0.41	0.44

Table 1 Results

5 Conclusion

This research paper explored the application of Visual Question Answering (VQA) technology in the medical domain, focusing on radiology scans. By leveraging and enhancing the BLIP architecture and proposing a multimodal transformer-based architecture, we demonstrated significant improvements in generating accurate answers to questions about medical images. However, the study has limitations, primarily the lack of large-scale annotated medical datasets. To address this, we employed data augmentation techniques and developed a custom medical tokenizer. Despite these efforts, further research, validation, and collaboration with medical experts are necessary to curate larger datasets and assess the models' performance in clinical settings.

In conclusion, this paper showcases the potential of VQA technology in revolutionizing medical radiology analysis. With continued advancements in dataset curation, architectural innovations, and extensive clinical validation, VQA systems hold great promise in empowering healthcare professionals and enhancing patient care. Future research should focus on addressing the limitations and conducting thorough clinical evaluations to facilitate the successful integration of VQA technology in real-world medical settings.

References

[1] Sahu, T.: Visual question answering with multimodal transformers. https://medium.com/data-science-at-microsoft/visual-question-answering-with-multimodal-transformers-d4f57950c867 (2022)

[2] Ben Abacha, A., Hasan, S.A., Datla, V.V., Liu, J., Demner-Fushman, D., Müller, H.: Vqa-med: Overview of the medical visual question answering task at imageclef 2019. In: Working Notes of CLEF 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org, Lugano, Switzerland (2019). https://ceur-ws.org/Vol-2380/paper_272.pdf

[3] Lau, J.J., Gayen, S., Demner, D., Ben Abacha, A.: Visual Question Answering in Radiology (VQA-RAD). OSF (2019). https://doi.org/10.17605/OSF.IO/89KPS . osf.io/89kps

[4] Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Batra, D., Parikh, D.: VQA: Visual Question Answering (2016)

[5] Zhang, P., Goyal, Y., Summers-Stay, D., Batra, D., Parikh, D.: Yin and yang: Balancing and answering binary visual questions. CoRR **abs/1511.05099** (2015) 1511.05099

[6] Zhu, Y., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7W: Grounded Question Answering in Images (2016)

[7] Yu, L., Park, E., Berg, A.C., Berg, T.L.: Visual Madlibs: Fill in the blank Image Generation and Question Answering (2015)

[8] Gao, H., Mao, J., Zhou, J., Huang, Z., Wang, L., Xu, W.: Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering (2015)

[9] Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to compose neural networks for question answering. In: Knight, K., Nenkova, A., Rambow, O. (eds.) Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1545–1554. Association for Computational Linguistics, San Diego, California (2016). https://doi.org/10.18653/v1/N16-1181 . https://aclanthology.org/N16-1181

[10] Chen, K., Wang, J., Chen, L.-C., Gao, H., Xu, W., Nevatia, R.: ABC-CNN: An Attention Based Convolutional Neural Network for Visual Question Answering (2016)

[11] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (2016)

[12] Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation (2022)

[13] Li, J.: BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. https://blog.salesforceairesearch.com/blip-bootstrapping-language-image-pretraining/ (2022)

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2021)

[15] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019)

[16] Li, J., Selvaraju, R.R., Gotmare, A.D., Joty, S., Xiong, C., Hoi, S.: Align before Fuse: Vision and Language Representation Learning with Momentum Distillation (2021)

[17] Wang, Z., Yu, J., Yu, A.W., Dai, Z., Tsvetkov, Y., Cao, Y.: SimVLM: Simple Visual Language Model Pretraining with Weak Supervision (2022)

[18] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D.A., Bernstein, M.S., Li, F.-F.: Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations (2016)

[19] Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, pp. 311–318. Association for Computational Linguistics, USA (2002). https://doi.org/10.3115/1073083.1073135 . https://doi.org/10.3115/1073083.1073135

[20] Lin, C.-Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (2004). https://aclanthology.org/W04-1013

[21] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (2019)

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need (2023)