DH602 Project Report
X-Ray Image Captioning using Actor-Critic Type Architectures

Abhishek Anand
200260002@iitb.ac.in

Rhishabh Suneeth
200260040@iitb.ac.in

Uppala Mukesh
200260056@iitb.ac.in

Ayushh
210070017@iitb.ac.in

Mohak Vyas
210040098@iitb.ac.in

Swadhin Dash
210020142@iitb.ac.in

Shristi Shrivastava
21d070069@iitb.ac.in

Abstract. In this project, we look into the possibility of employing actor-critic frameworks inspired by reinforcement learning for medical X-Ray image captioning. The initial model is inspired by an actor dual critic model (cite) used for remote sensing. We first adapt their code for use on the IU chest X-Ray dataset, followed by subsequent experiments where we replace the LSTM in the actor with a transformer model. We give our results for both variants, followed by a conclusion and future direction of work.

Keywords: X-Ray Image captioning · Actor-critic architectures

1 Introduction

The task of generating descriptive text for images, known as Image Captioning, sits at the crossroads of computer vision and natural language processing (NLP), to connect visual and textual modes by overcoming the semantic gap. Creating detailed image descriptions may come easily to humans, but it has proven difficult for machines until the emergence of deep learning techniques. The advancement of deep learning, especially with the introduction of CNNs and RNNs, has transformed image captioning. Early models for image captioning used CNNs to identify important characteristics in images and RNNs to transform these characteristics into meaningful sentences. This change in thinking was influenced by the comparison to machine translation, in which a system translates text back and forth using an encoder-decoder structure. Yet, image captioning goes beyond simple translation; it necessitates a deep comprehension of visual semantics and how they are portrayed in text. Creating models that can understand significant connections between visual and textual components is crucial for capturing the essence of multimodal data. Reinforcement Learning (RL) is a promising method for sequential decision-making, allowing agents to discover environments and acquire effective decision-making strategies.

While previous work has been done on Reinforcement Learning based image captioning tasks, there was little to no work done on medical imaging. We introduce an innovative Actor Dual-Critical training setup tailored for Medical Image Captioning using a Swin Encoder for feature extraction. By utilizing RL principles, we train an actor-critic model to learn to be able to generate captions and rate the caption quality itself, enabling self-evaluation and better convergence of the model.

2 Implementation Details

Our suggested design implements an Actor-Critic architecture using Vision transformers for feature extraction. In Figure 1, the structure comprises numerous essential elements:

1. **Tokenizer**: The function of this module is to preprocess the input text. It tokenizes the text into words and eventually, each token gets a token embedding that is used for further calculations.

2. **Encoder**: The encoder uses a sequence of Swin Transformer blocks. These blocks utilize shifting window-based multi-headed self-attention mechanisms to understand how various components of the input sequence are interconnected. This process converts the input image into a detailed numerical depiction.

3. **Decoder**: The decoder enhances the encoded representation produced by the encoder. It includes several tiers of RNN units, like LSTM. These RNN units excel at processing data in a sequence, allowing the decoder to produce the output text incrementally.

4. **Reviewer**: The presence of a critic is essential for improving the quality of the generated content. This component is an LSTM-based model that evaluates the quality of generated captions. Throughout the training period, the decoder receives input from the critic to steer it towards generating text of better quality.

This design operates as a model that processes images and outputs sequence-to-sequence text. The configuration with an encoder-decoder aids in producing text, while the critic element helps to improve the overall quality of the result.

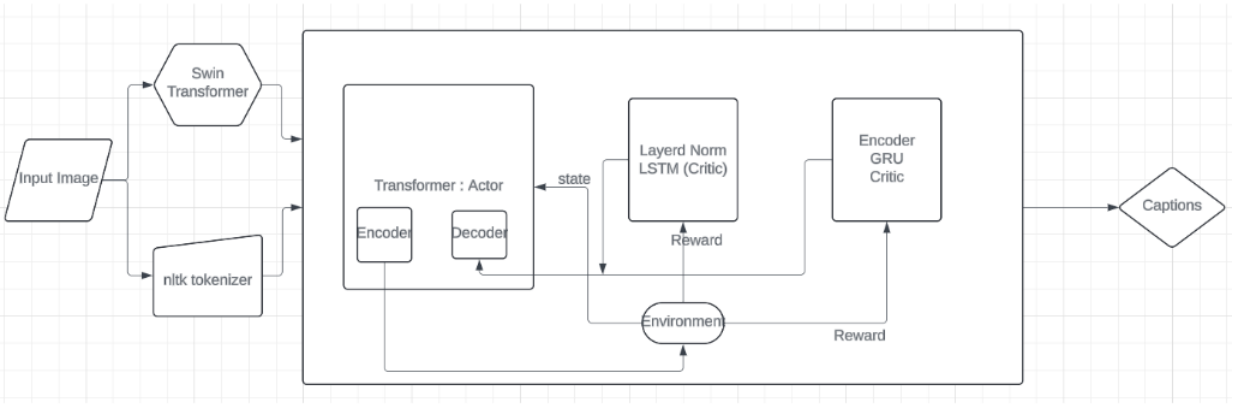

Figure 1: Transformer-based Actor-Critic Architecture

2.1 Swin Transformer Encoder

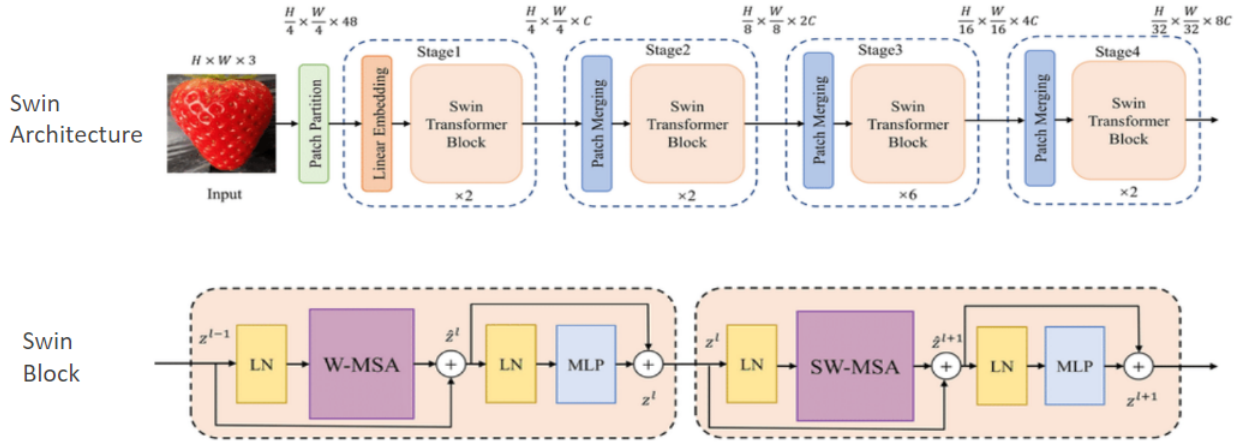

Figure 2: Architecture of the swin transformer Encoder

2.2 Actor

The actor provides a measure of confidence $q_\pi(a_t|s_t)$ to predict the next action according to the current state. For extraction of features, either an encoder CNN like AlexNet or the encoder Swin transformer is used. The extracted features are then fed as input to the GRU (Gated recurrent unit). The functionality of the actor is as follows:

$$
\begin{aligned}
f &= \text{Extractor}(I) \\
\phi_o &= f \\
o_t^g, h_t^g &= GRU(\phi_{t-1}, h_{t-1}^g) \\
o_t^l, h_t^l &= LSTM(o_t^g, h_{t-1}^g) \\
q_\pi(a_t|s_t) &= \psi(o_t^l) \\
\phi_t &= \xi(w_{t-1})
\end{aligned}
\tag{1}
$$

Here, Extractor refers to the image extractor part which is the Swin transformer encoder or AlexNet CNN. These are the features obtained from weights of the second last layer of the pretrained encoder part. o_t^g and o_t^l are outputs of the GRU and LSTM respectively. ψ transforms the output of the LSTM to a space where the dimension is equal to the vocabulary dimension. ξ denotes the embedding model to represent words in a common embedding space. The policy is denoted by $\pi(a_t|s_{t-1}$. This model is trained to optimise the objective:

$$
\min_\pi \sum_{t=0}^{T} \log(q_\pi(a_t|s_t)
\tag{2}
$$

2.2.1 Modified Actor

Now, we replace LSTM actor with a transformer actor. The modified actor is a typical transformer based captioning model. The functionality is as follows:

$$\begin{aligned} f &= \text{Extractor}(I) \\ o &= \text{Encoder-Decoder}(f) \end{aligned} \tag{3}$$

The Captioning Model is image feature extractor followed by a transformer based encoder-decoder block which can be used for captioning images. The encoder block consists of a MultiHeadAttention Block followed by an Linear layer and layerNormalisation layer. The decoder block consists of 2 MultiHeadAttention Blocks followed by an LRL (Linear-ReLU-Linear) block which is then followed by layerNormalisation layers. The detailed model structure is described by the figure below:

```
(encoder): TransformerEncoderBlock(
  (attention): MultiheadAttention(
    (out_proj): NonDynamicallyQuantizableLinear(in_features=8, out_features=8, bias=True)
  )
  (dense_proj): Linear(in_features=256, out_features=8, bias=True)
  (layernorm_1): LayerNorm((8,), eps=1e-05, elementwise_affine=True)
)
(decoder): TransformerDecoderBlock(
  (attention_1): MultiheadAttention(
    (out_proj): NonDynamicallyQuantizableLinear(in_features=8, out_features=8, bias=True)
  )
  (attention_2): MultiheadAttention(
    (out_proj): NonDynamicallyQuantizableLinear(in_features=8, out_features=8, bias=True)
  )
  (dense_proj): Sequential(
    (0): Linear(in_features=8, out_features=16, bias=True)
    (1): ReLU()
    (2): Linear(in_features=16, out_features=8, bias=True)
  )
  (layernorm_1): LayerNorm((8,), eps=1e-05, elementwise_affine=True)
  (layernorm_2): LayerNorm((8,), eps=1e-05, elementwise_affine=True)
  (layernorm_3): LayerNorm((8,), eps=1e-05, elementwise_affine=True)
  (embedding): PositionalEmbedding(
    (token_embeddings): Embedding(1544, 8)
    (position_embeddings): Embedding(60, 8)
  )
  (out): Linear(in_features=8, out_features=1544, bias=True)
  (dropout_1): Dropout(p=0.1, inplace=False)
```

Figure 3: Transformer Encoder-decoder Architecture for the captioning model

This model is trained to optimise the cross entropy loss.

2.3 Critics

There are 2 critics which are utilised to guide the actor. They are namely:

2.3.1 Encoder Decoder LSTM Critic

The working of this critic is as follows:

$$\begin{gathered} h_o^{enc} = \text{Extractor}(I) \\ \eta_t = \xi(S) \\ o_t^{enc}, h_t^{enc} = RNN_{enc}(\eta_t, h_{t-1}^{enc}) \\ h_o^{dec} = \psi_2(h_T^{enc}) \\ i_1^{dec} = \psi_1(o_T^{enc}) \\ o_t^{dec}, h_t^{dec} = RNN_{dec}(i_t^{enc} m, h_{t-1}^{dec} \\ o = \text{Encoder-Decoder}(f) \end{gathered} \tag{4}$$

Here, $S = (w_1, w_2, ...w_T)$ denotes a natural language description of the image. $(RNN)_{enc}$ and RNN_{dec} are encoder-decoder RNN respectively.ψ_1 and ψ_2 are linear functions with dimension equal to that of the embedding space of sentences, along with ReLU activation. This critic is trained to optimise for MSE between output of decoder and the features:

$$A_{gen} = \frac{\frac{\sum_{t=0}^{T} o_t^{dec}}{|S|} f}{||f|| || \frac{\sum_{t=0}^{T} o_t^{dec}}{|} ||} \tag{5}$$

where A_{gen} and A_{orig} are the accuracies of the network when captions are generated by the actor and ground truth captions are fed into the encoder respectively. The advantage function for this critic is defined as:

$$A_{ed} = A_{gen} - \delta_t A_{orig} \tag{6}$$

Setting δ_t to 1 directly results in a non-converging policy, hence δ_t is slowly increased to 1 over the epochs.

2.3.2 Value Network Critic

This critic consists of an RNN which outputs a value function v_θ^π, given the words predicted by the current policy and image features extracted. The hidden state of the RNN is initialised with these features. The output of this network is directed to be the expected value of future rewards for choosing a particular state given the current policy. The ROUGE score is used as a reward signal for the entire generated sentence. This network is optimised for the Huber loss.

$$L = \begin{cases} ||v_\theta^\pi - r_T||^2 & ||v_\theta^\pi - r_T|| \leq \delta = 0.5 \\ \delta ||v_\theta^\pi - r_T|| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

3 Experiments and Results

The actor and critic models are initially pretrained for 10 epochs. This is followed by 50 epochs of joint actor-critic training. In few cases, we have reduced the number of pretraining and training epochs. Initially training was done for the base actor-critic model (using CNN as encoder), followed by using Swin transformer as encoder and then using LayerNorm LSTM instead of the normal LSTM in the actor and critic modules. Finally, the LSTM and GRU in the actor is replaced by a transformer-based encoder-decoder captioning model. This model took the least time to train, due to the advantages of the transformer architecture. The training was done using NVIDIA RTX A-6000 GPU's. The results are summarised below:

Model	B1	B2	B3	B4	ROUGE
CNN	0.3059	0.0396	0.0396	8.11e-05	0.2223
Swin	0.3077	0.0412	0.0412	0.0001	0.2250
Layer Norm Swin	0.3341	0.0537	0.0027	0.0001	0.2383
Transformer Aug	0.3930	0.1002	0.0264	0.0085	0.2975
Layer Norm Aug	0.3922	0.0662	0.0052	0.0006	0.2745

Figure 4: Result Overview

Figure 5: Predicted: [['start', 'the', 'heart', 'is', 'normal', 'in', 'size', 'and', 'contour']]
Ground truth: [['start', 'the', 'cardiac', 'silhouette', 'and', 'mediastinum', 'size', 'are', 'within', 'normal', 'limits']]

Figure 6: Predicted: [['start', 'the', 'heart', 'is', 'normal', 'in', 'size', 'and', 'contour']]
Ground truth: [['start', 'the', 'cardiac', 'contours', 'are', 'normal']]

4 Conclusion

In this work, we explore using actor-critic frameworks for medical X-Ray image captioning, adapting a model from remote sensing. We replace the actor's LSTM with a transformer and evaluate both variants on the

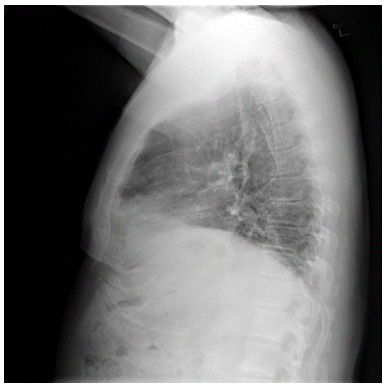

Figure 7: Predicted: [["start", "heart", "size", "suspicious", "pulmonary", "appear", "within", "normal", "limits"]]
Ground Truth: : [["start", "the", "heart", "is", "normal", "in", "size", "and", "contour"]]

Figure 8: Predicted: : [["start ", "the", "lungs", "are", "clear", "bilaterally"]]
Ground Truth : [["start", "low", "lung", "volumes", "are", "present"]]

IU chest X-Ray dataset. Our results demonstrate the effectiveness of these approaches, showcasing their potential for medical image captioning. Our architecture, featuring a Swin Encoder and a transformer-based actor, offers a promising approach for medical image captioning. By training the model to generate captions and evaluate their quality, we enable self-assessment and improve convergence. In conclusion, our study introduces novel methods for medical image captioning, demonstrating the potential of advanced deep learning techniques in this domain. Future work could refine our approach and explore its application to other medical imaging tasks. Reinforcement learning is a very new field in medical imaging tasks and hasn't been explored much. It holds a lot of potential.

References

1. Swin for Classification
   https://github.com/microsoft/Swin-Transformer/blob/main/get_started.md

2. RSTAC
   https://link.springer.com/article/10.1007/s00521-022-07848-4#:~:text=This%20model%20is%20composed%20of,comprised%20of%20the%20feature%20I_v

3. Actor Dual-Critic - Subhasis
   https://github.com/ruchikachavhan/ADC-image-captioning?tab=readme-ov-file

4. RTMIC
   https://link.springer.com/chapter/10.1007/978-3-030-32692-0_77#:~:text=In%20this%20paper%2C%20we%20have,Transformer%20based%20sequence%20generation%20model
5. Vision Transformers
   https://colab.research.google.com/drive/1P9TPRWsDdqJC6IvOxjG2_3QlgCt59P0w?usp=sharing#scrollTo=w0a8TAbg3KQd
   https://youtu.be/j3VNqtJUoz0?si=QTICret85RFRvBc9
6. Image captioning with transformers
   https://github.com/zarzouram/image_captioning_with_transformers/tree/main
7. Swin V2
   https://github.com/ChristophReich1996/Swin-Transformer-V2/blob/main/example.py
8. Swin module
   https://huggingface.co/docs/transformers/model_doc/swinv2
9. Swin transformer using Pytorch
   https://www.kaggle.com/code/pdochannel/swin-transformer-in-pytorch
10. Image captioning using RL
    https://github.com/aashay-m/image-captioning-through-rl/blob/master/Final%20Project%20Report.pdf
11. Image Captioning using Pytorch
    https://thepythoncode.com/article/image-captioning-with-pytorch-and-transformers-in-python#google_vignette

5 Link to Implementation

Github: https://github.com/RhishabhS/dh602_project