Image Caption Generator

Harsh Agarwal 21B090007 Prankul Yadav 21D070050 Jayveer Singh Shikarwar 21D070033

Abstract—In this project we made multiple encoder-decoder generative style models for image captioning.We utilize pre-trained CNNs (VGG19, ResNet152) as encoders and employ LSTM ,LSTM with modification and LSTM with attention as decoders.Our evaluation metrics including METEOR and BLEU gauge caption quality against reference annotations.The report extensively compares these models based on these metrics, providing a comprehensive analysis of their performance in caption generation.

Index Terms—Image Captioning, Encoder-Decoder Architectures, VGG19, ResNet152, LSTM, Attention Mechanism, METEOR, BLEU

1 Introduction

IMAGE captioning is a significant task within the domain of computer vision and natural language processing, involving the automatic generation of descriptive captions for input images. The complexity of this task extends beyond mere object recognition and includes the challenge of understanding the relationships between objects within the image.

Moreover, the above semantic knowledge has to be expressed in a natural language like English,which means that a language model is needed in addition to visual understanding. Achieving this seamless integration of visual perception and linguistic expression poses a paramount challenge, requiring robust models capable of effectively bridging the gap between visual content and linguistic context.

2 General Overview

The first deep learning solution to this problem was the Neural Image Caption(NIC) model [1] from which our project is heavily inspired. In NIC they use a deep convolution neural network(CNN) as an image **encoder** and a deep recurrent neural network(RNN) as a **decoder** which generates the captions. Our project is heavily inspired by this model, we also explore some improvements to this model. The next model which was a major improvement for this model is neural image caption generator using visual attention. We have made 6 models with 2 encoders (VGG19 and ResNet 152) and 3 different decoders (LSTM, Improved LSTM and LSTM with attention) and compare the METEOR and BLEU scores of all our models.

3 Long Short Term Memory (LSTM):

LSTM are special types of Recurrent Neural Network (RNN). RNNs use the inputs from previous layers to determine the output of the next layer of the model. RNNs are helpful for our model because we can determine the next word of the sentence by looking at previous words of the sentence and the input image features. LSTMs are designed to capture long-term dependencies in sequential data. The main components of LSTM include: Cell State(C_t),Hidden State(h_t),Input Gate(i_t),Forget Gate(f_t),Cell Update(g_t),Output Gate(o_t). The equations for LSTM operations are:

$$
\begin{aligned}
i &= \sigma(W_{ii}x_t + W_{hi}h_{t-1} + b_{ii}) \\
f_t &= \sigma(W_{if}x_t + W_{hf}h_{t-1} + b_{if}) \\
g_t &= tanh(W_{ig}x_t + W_{hg}h_{t-1} + b_{ig}) \\
o_t &= \sigma(W_{io}x_t + W_{ho}h_{t-1} + b_{io}) \\
C_t &= f_t \odot C_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot tanh(C_t)
\end{aligned}
$$

LSTMs are preferred to RNN as they address the vanishing gradient problem which is common in RNNs.

4 MODELS

We are using an encoder-decoder style architecture similar to NIC.Here encoder extracts the features from input image and the features from encoder are mapped as fixed length inputs to the decoder RNNs which generate captions.The decoder also receives variable length captions of the training set as input during training .We made 6 models with 2 encoders (VGG19 and ResNet152) and 3 different decoders (LSTM, Improved LSTM and LSTM with attention).

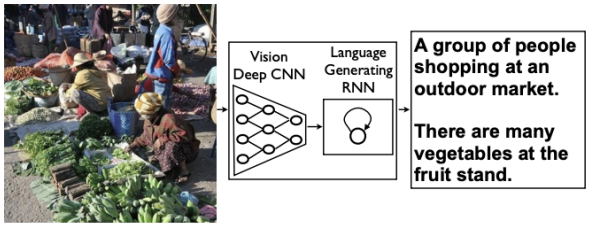

Fig. 1: Encoder-Decoder architecture [1]

4.1 Encoder

We used pre trained CNNs which are made for the purpose of image classification as encoders.

4.1.1 VGG-19 CNN Model

The VGG19 model is a deep convolutional neural network architecture introduced by the Visual Graphics Group (VGG) at the University of Oxford.It has 19 layers, including 16 convolutional and 3 fully connected layers.It is pre-trained on a large dataset of images known as **ImageNet**.VGG19 is celebrated for its strong feature extraction capabilities, enabling it to capture intricate patterns and hierarchies within images.

4.1.2 ResNet-152 CNN Model

ResNet-152 is a convolutional neural network architecture renowned for its depth and innovation in addressing the vanishing gradient problem.It is developed by Microsoft Research, ResNet-152 is a part of the Residual Network (ResNet) family, distinguished by its exceptional depth—comprising 152 layers—while mitigating the challenges associated with training very deep networks.

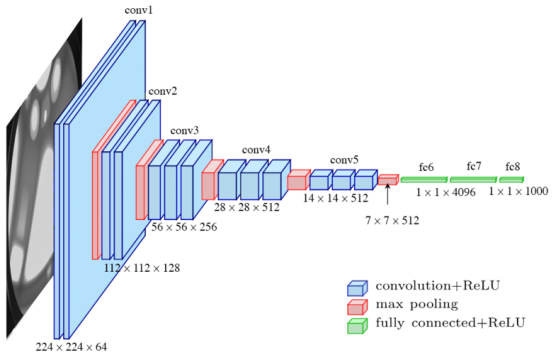

Fig. 2: VGG19 architecture [5]

The key innovation of ResNet lies in its introduction of residual connections, also known as skip connections or shortcut connections. These connections enable the network to learn residual functions, allowing information to bypass certain layers and propagate directly to deeper layers. By utilizing these shortcuts, ResNet-152 effectively alleviates the degradation problem encountered in training extremely deep networks, facilitating smoother gradient flow during backpropagation and enabling the training of deeper architectures without suffering from diminishing performance.

ResNet-152's architecture consists of a series of convolutional layers, followed by residual blocks with identity mappings and bottleneck structures to optimize computational efficiency. Pre-trained on vast image datasets like ImageNet, ResNet-152 excels in feature extraction, capturing intricate details and hierarchical representations within images. Its remarkable depth and ability to capture complex visual features make it an excellent choice as an encoder in various computer vision tasks, including image recognition, object localization, and, in your context, image captioning, where it serves as a robust feature extractor enhancing the performance and accuracy of the model.

4.2 Decoder

4.2.1 LSTM

This is the decoder used in our baseline model, it is a naive LSTM model which is im-

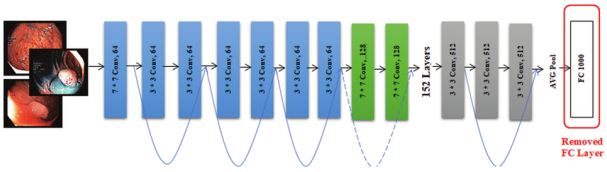

Fig. 3: ResNet 152 architecture [6]

ported from pytorch framework of python. The model does not work well on images which have multiple objects and is not able to focus on specific important details of the image.

4.2.2 Improvements on LSTM

On the quest to improve the performance of our decoder, we wanted to implement attention as our final goal, but we have implemented a model which improves on basic LSTM with the ideas of attention, we maintain another shallow neural network which sends context to each LSTM cell (works similar to attention which is discussed next section) and receives lstm input vector as input too with feature vector. While Training this context neural network is only used once for entire caption (i.e last caption word), which clearly underperforms when compared to actual attention but is slightly better than basic LSTM because we are having an extra neural layer with nonlinearities to classify (although the performance boost isnt too noteworthy)

4.2.3 LSTM with attention

The above problem of LSTM can be addressed by using attention. Attention in simple words says that while generating a word, we should focus more on certain prominent features of the image contrary to the naive LSTM, where we were giving same preference to all the features. Instead of taking features,hidden states as input to the LSTM, we take something called context and hidden state as input to generate a new word, here context is generated using features and hidden state as an input to a small neural network, the neural network is described by the following: Lets assume that features are denoted by F_t, context is denoted by C_t, α_i denotes the attention weights and h_{t-1} is the previous hidden state of the decoder,e_i denotes the relation between features and hidden state, usually two linear layers and a non linear activation applied on the sum of outputs, then C_t and α_i's are given by:

$$e_i = compat(h_{t-1}, F_t)$$
$$\alpha_i = \frac{\exp e_i}{\sum_j \exp e_j}$$
$$C_t = \sum \alpha_i F_i$$

5 Metrics

The metrics which we had used for this project were BLEU and METEOR. These metrics were used to compare the targeted captions and caption predicted by the model. These metrics were implemented by **NLTK** library and were used by us to compare the models.

5.1 BLEU

Bilingual Evaluation Understudy, better known as BLEU is one of the first metrics that were used by Neural image caption generator [1] and Neural image caption generator with visual attention [2]. This is a usual metric used in Natural language processing.

5.1.1 Calculation

The calculation of this metric starts with calculating precision, it is ratio of number words correctly predicted to the number of words predicted. But this has a obvious problem. that is if the model had predicted the same word more than 1, precision would be 1, so an modification was made to this.

That is clipped precision which is the ratio of number of words predicted correctly to the number of words predicted with number of times each word occurred being clipped to the number of times that word had occurred in the targeted caption. Now back to calculation of the BLEU, It considers n-grams of sentences. n-grams of sentences are n consecutive words in the sentence. BLEU is actually product of geometrically weighted precisions of 1-gram, 2-gram, 3-gram and 4-grams and brevity penalty which is way to penalise sentences for being too short.

The idea behind brevity penalty is that precision of 1-gram will be 1 if only one word

is predicted and it is correct. The equations are given below:

$$\text{precision} = \frac{\text{No. of words predicted correctly}}{\text{No. of words predicted}}$$

$$\text{Clipped precision} = \frac{\text{Clipped No. of words predicted correctly}}{\text{No. of words predicted}}$$

Say p_1, p_2, p_3, p_4 be the precision of 1-gram, 2-gram, 3-gram, 4-gram respectively, w_1, w_2, w_3, w_4 be the weight of 1-gram, 2-gram, 3-gram, 4-gram respectively and c, r be the length of predicted caption and length of targeted captions then

$$\text{Brevity penalty} = \begin{cases} 1 & c \geq r \\ e^{1-\frac{r}{c}} & c \leq r \end{cases}$$

$$\text{BLEU} = \text{Brevity penalty} \times \prod_{i=1}^{4} p_i^{w_i}$$

For BLEU-1, $w_1 = 1$ and other weights are zero. For BLEU-2, $w_1 = w_2 = 1/2$ and rest are zero. For BLEU-3, $w_1 = w_2 = w_3 = 1/3$ and $w_4 = 0$. For BLEU-4, $w_1 = w_2 = w_3 = w_4 = 1/4$.

5.1.2 Problem with BLEU

The problem with BLEU score is that it wouldn't allow reordering, it looks only for exact word matches and does not consider the meaning of words. So, we had to use another metric that is METEOR.

5.2 METEOR

Metric for Evaluation of Translation with Explicit ORdering better known as METEOR is a metric which allows different words (synonyms) being used in the predicted caption and allows reordering of words. It has also produced good correlation with human judgement at the sentence or segment level than BLEU.

5.2.1 calculation

Calculation of METEOR is slightly more complex than that of BLEU. It has been explained in [4]. But basically it is based on harmonic mean of unigram precision and recall with recall being more weighted than precision.

6 Result

Dataset used is Flickr8k, This dataset has 5 captions per image and in total 8091 images. The train validation split is 80:20. The seed has been fixed to 42, for reproducing the results.

The BLEU scores and METEOR scores obtained for each model(best model obtained during training for 10 epochs) is given in table 1. The hyperparameters used for these results is given in table 2.additionally, the number of parameters in each model is given in table 1 along with the BLEU scores. Even though the rounded value for the number of parameters is same for models with VGG and ResNet, the baseline model had lesser parameters than the modified LSTM model and the modified LSTM model had lesser parameter than LSTM with attention model. The dimension for attention in LSTM with attention models is 256 and the dimensions for the extra modified layer in modified LSTM model is also 256

Predicted captions for few images are given in the figures 4 and 5. In which 4 contains the images which are correctly captioned and 5 contains the images which are incorrectly captioned.

7 Inference

The model easily identifies the pictures with single object as the model in this case has to identify only an object and the environment in which object is. Observing the BLEU and METEOR scores, we get to know that VGG and LSTM with attention is the best model and models with ResNet152 didn't work as well as their VGG counterparts.

The Images which are incorrectly captioned reveal that the models couldn't predict well on the images with multiple images. The most mistakes the models do are in identifying the colour of a object or the number of people in the image. This happens as the images happen to have different shades of the same colour and the models tends to consider the boundaries of objects which have those shades. The number of people is also difficult for the model to identify because from the perspective of the image, only few of those people would be focused and the model, hence does not consider those people

TABLE 1: BLEU scores and METEOR scores

This BLEU scores and METEOR scores will change according to the seed too.

Model	Number of parameters	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
VGG and LSTM	25 million	0.3635	0.2053	0.1172	0.0634	0.2004
ResNet152 and LSTM	13 million	0.2971	0.1441	0.0633	0.0253	0.1641
VGG and Modified LSTM	25 million	0.3405	0.1797	0.0956	0.0508	0.1873
ResNet152 and Modified LSTM	13 million	0.1836	0.0807	0.0424	0.0242	0.1113
VGG and LSTM with attention	10 million	0.3764	0.2159	0.1248	0.0704	0.2135
ResNet152 and LSTM with attention	16 million	0.2546	0.1241	0.0607	0.0304	0.1535

TABLE 2: Hyperparameters

Model	learning-rate	embed dim	hidden dim	no. epochs
VGG and LSTM	0.0001	512	512	10
ResNet152 and LSTM	0.0001	512	512	10
VGG and Modified LSTM	0.0001	512	512	10
ResNet152 and Modified LSTM	0.0001	512	512	10
VGG and LSTM with attention	0.0003	300	512	10
ResNet152 and LSTM with attention	0.0003	300	2048	10

Fig. 4: predicted correct: dog is running through the water, dog is running through the water, group of people are walking in the snow, two dogs are playing in the grass, dogs jumps over hurdle,man is sitting on rock face, man on bike is riding bicycle on dirt path, two dogs(left to right)

in the caption, but the major problem is that the model does not learn how to count the number of objects in the code. The model also mis-captions child and adult or man and woman. These mistakes also occur due to the reason that the model does not learn to differentiate between the objects in training time. Also we observed that the model makes mistake as the length of the caption increases as the LSTM becomes less ineffective as the length of the caption increases, this can be observed from the attention heat maps as well.

8 Scope from Improvement

The models we had implemented and analysed are basic models to the current state of the art models. We had tried to implement an extra encoder (Inception CNN) and an extra decoder(transformer), but couldn't finish due to bugs in it. But there is a lot of scope of improvements under this basic architecture itself. One could use GRUs and customised CNNs to make the model more accurate.

9 Conclusion

We had used github [7] repository for the baseline model. The github link for our repo

Fig. 5: predicted wrong:man in black shirt and black hat is standing in front of building, two men in black clothes, two children are playing in the water,girl in pink shirt and blue shorts is jumping into the air,boy in blue shirt is jumping on trampoline,three men play in field,three children are playing in the snow,boy in blue shirt is swinging on swing

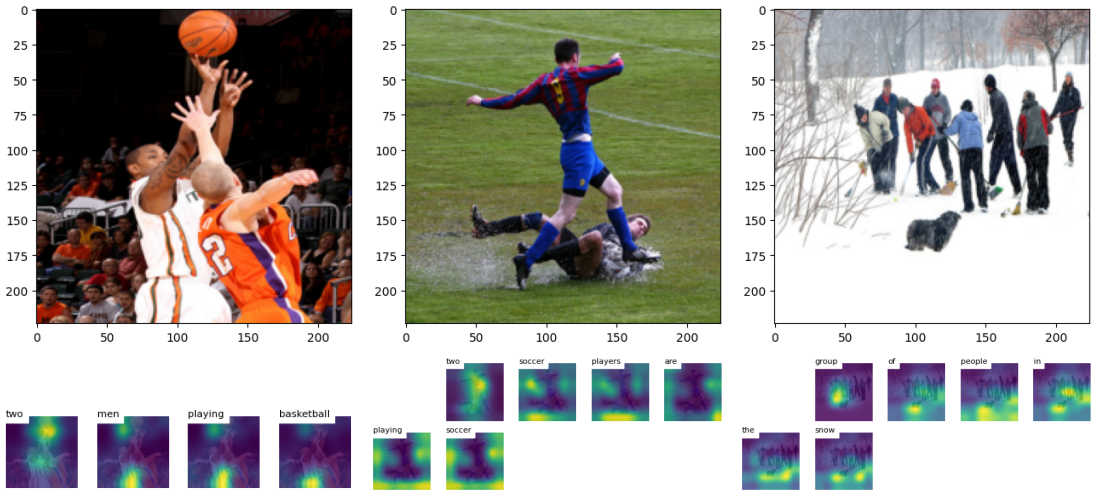

Fig. 6: Attention Heat maps for ”two men playing basketball”,”two soccer players are playing soccer”,”group of people in the snow”(left to right)

is [8].We had used various other github repositories for our project, few other major ones are listed below [9].

- LSTM with attention
- Papers with code
- other LSTM with attention models

REFERENCES

[1] Show and Tell: A Neural Image Caption Generator(Paper)
[2] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention(Paper)
[3] BLEU score explained(Medium page)
[4] METEOR Score explained(wikipedia page)
[5] vgg image from Medium
[6] RESNET image from ReserchGate
[7] Mostly refered : GitHub
[8] Our Repo
[9] Other reference: