AUTOMATIC IMAGE CAPTIONING

ISHITA SOJITRA

August 16, 2023

A Project Submitted to the
Faculty of Graduate Studies
in Partial Fulfillment of the Requirements for the Degree

Master of Science in Information Technology
Department of Mathematical and Physical Sciences

FACULTY OF GRADUATE STUDIES
Concordia University of Edmonton
Edmonton, Alberta

AUTOMATIC IMAGE CAPTIONING

ISHITA SOJITRA

Approved:

Nasim Hajari, Ph.D.

Supervisor Date

Committee Member Date

Patrick Kamau, Ph.D.

Dean of Graduate Studies Date

Abstract

The difficult and multidisciplinary process of automatically creating accurate and logical textual descriptions for photographs is known as automatic image captioning [1]. Modern Neural Networks excel in tasks like Computer Vision and Natural Language Processing, but their memory and compute appetite hinder deployment on resource-limited edge devices. Researchers have developed pruning and quantization algorithms to compress networks without compromising efficacy. The process typically involves two main steps: Image understanding and Caption generation. This work presents an unconventional end-to-end compression pipeline for a CNN(Convolutional neural network)-LSTM(Long short-term memory)-based Image Captioning model, achieving a 73.1percentage reduction in model size, 71.3percentage reduction in inference time, and 7.7percentage increase in BLEU(bilingual evaluation understudy) score compared to uncompressed models [9]. By comparing generated captions with reference captions created by humans, evaluation metrics like BLEU (bilingual evaluation understudy) and METEOR (metric for evaluation of translation with explicit ordering) are used to evaluate the quality of generated captions [16]. The purpose of Automatic image processing is to extract useful information from photos, making analysis, interpretation, and manipulation faster, precise, and effective in various fields, using computational algorithms.

Keywords: Image understanding, Caption generation, Computer vision, Natural language processing, Deep learning, Convolutional neural networks (CNNs), Recurrent neural networks (RNNs), Long short-term memory (LSTM), Evaluation metrics.

Acknowledgments

This work was supported in part by the Concordia University of Edmonton Master's study of the Faculty of Science program, and in part by a gift from Google Scholar. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s), and thanks to Chat GPT for the content. The code used in this paper is from Github with several changes. I also would like to thank Prof. Nasim Hajari for her help and guidance through the project and special thanks to Dr. Patrick Kamau, Dean of the Concordia University for giving me opportunity to complete my final project under his guidance.

Contents

1 Introduction 1

2 Objectives / Research Questions 2

3 Literature review (and theoretical framework) 3

3.1 Mobile Intelligence Assisted by Data Analytics and Cognitive Computing 2020 . . . 3

3.1.1 Attention Mechanism . . . 3

3.2 Efficient CNN-LSTM based Image Captioning using Neural Network Compression . . . 5

3.3 Improving Image Captioning with well-known metrics . . . 5

4 Project Design 7

5 Body 10

5.1 Working of Model . . . 10

5.2 Datasets . . . 13

5.2.1 MS COCO Dataset . . . 13

5.2.2 Flickr30K Dataset . . . 14

5.3 Predict Sample Images . . . 14

5.4 Experiments Results . . . 16

5.5 Future Scope . . . 17

6 Conclusions 17

List of Tables

1 Comparison for AICRL with and without attention[13] 6
2 Comparison for AICRL with and without attention[2] 6
3 The performance comparison in the Flick8K dataset [14]. 7
4 The performance comparison in the MS COCO 2014 dataset [7]. . . . 7

List of Figures

1 Model without attention[1] 3
2 Model with attention[1] 4
3 Model Architecture [15] 7
4 Demo Model 9
5 Demo Model 9
6 Working Model [4] 10
7 Other parameters for training the model 11
8 Word generation (LSTM output) 11
9 sample images from the MS COCO dataset[7] 13
10 sample images from the Flickr30K dataset[14] 14
11 Sample Image-1 [7] 15
12 Sample Image-2 [14] 15
13 Sample Image-3 15
14 Sample Image-4 15
15 Comparison Graph [4] 16

1 Introduction

Deep neural networks have been extremely popular in recent years thanks to their ability to produce cutting-edge results on tasks like classification, recognition, and prediction. Such complicated networks can't be easily transferred to low power mobile devices because of the massive computational footprint they have. Due to their light weight and sleek design, modern mobile devices' power and thermal capacity are further limited [12].

The process of creating a natural-language response that accurately captures the visual content of an image is known as automatic image description. Starting with the 2015 COCO challenge winners and continuing with a range of enhancements for a review, there has been an explosion of deep learning architecture-based solutions that have been presented for this purpose [15]. Utilising descriptions for picture indexing or retrieval is one of the practical uses of autonomous image description systems. Another is to assist people with visual impairments by converting visual signals into information that can be spoken aloud using text-to-speech technology [1]. Aligning, using, and advancing the most recent advancements at the nexus of computer vision and natural language processing is viewed as the scientific challenge.

Automatic image captioning has received a lot of research attention; it may be divided into three categories: template-based image captioning, retrieval-based image captioning, and novel image caption generation. Template-based image captioning finds the objects, attributes, and actions first, then fills in the spaces in a pre-determined template. The image caption is chosen from a group of visually comparable images with captions that are found using retrieval-based algorithms after the training dataset. These techniques can provide captions that are syntactically valid, but they cannot produce captions that are semantically or image-specific [17]. The novel techniques for creating image captions analyse the image's visual content before utilising a language model to create descriptions for the images. For a given image, novel caption generation can produce new captions that are semantically more correct than prior methods, as opposed to the first two categories. The majority of papers in this area use deep learning and machine learning, which is also the strategy used in this paper. Encoder-decoder frameworks are frequently used in this area for picture captioning [16]. This framework was first presented by Kiros et al. to explain a multimodal log-bilinear Wireless Communications and Mobile Computing model for image captioning with a fixed context window. Recent studies have employed deep recurrent neural networks (RNN) as the decoder and deep convolutional neural networks (CNN) as the encoder, which has shown to be promising [8, 10, 11]. It is still difficult to choose the appropriate CNN and RNN models for image captioning [15].

In order to sparsify the network's encoder and decoder components, we used magnitude-based pruning. Additionally, we put into practise and tested two different quantization schemes: post-training and quantization conscious training. The encoder used post-training quantization, while the decoder experimented with both quantization methods [17]. In this study, we provide the results of applying several compression architectures to the captioning model. It's interesting to note that several of them even outperformed the complete uncompressed model [19]. We fervently recommend a certain compression architecture to compress the captioning model in light of the results given. The compressed model delivers acceptable time savings during inference in addition to excellent storage efficiency.

2 Objectives / Research Questions

(1) Accurate and meaningful descriptions: Aims to provide accurate, meaningful descriptions of visual content, capturing key objects, actions, relationships, and context [12].

(2) Bridging vision and language: Integrates computer vision and natural language processing, transforming visual information into coherent textual descriptions [15].

(3) Contextual and coherent captions: Aims for grammatically correct, fluent, and human-aware captions that accurately represent images [1].

(4) Handling ambiguity and diversity: Aims to capture diverse perspectives and interpretations of images with multiple valid interpretations or ambiguous elements [12].

(5) Capturing fine-grained details: Captures high-level objects, actions, and finer details, generating descriptive, descriptive captions that highlight the scene's characteristics and attributes [16].

(6) Human-like captioning: Aims to create high-quality, styled, and linguistically fluent captions by developing algorithms that mimic human understanding [4].

(7) Real-time and scalable captioning: Develops efficient, scalable algorithms for real-time caption generation in large image volumes [3].

(8) Evaluation and benchmarking: Assesses quality and performance using metrics and benchmarks for fair comparison and progress measurement [12].

3 Literature review (and theoretical framework)

3.1 Mobile Intelligence Assisted by Data Analytics and Cognitive Computing 2020

The AICRL model is an automatic image captioning system using ResNet50 and LSTM with software attention. It generates description sentences using a CNN and RNN architecture, extracting visual features using ResNet50 networks. The decoder uses LSTM for sentence generation, and soft attention in the decoder allows the model to selectively focus attention on specific image parts for better prediction. The model is fully trainable using stochastic gradient descent. Extensive experiments and empirical determinations demonstrate its effectiveness [1].

3.1.1 Attention Mechanism

The soft attention mechanism is used to isolate image content in image classification problems, as it doesn't require processing all pixels. This approach is more efficient than convolutional neural networks, which spend computational resources on all parts of the image. It is implemented by adding an additional input of attention gate into LSTM that helps to concentrate selective attention. The main drawback of the model without attention is that it tries to decode the full image from the last hidden layer of in Figure 1. It is like an analogy with machine translation in the whole process. To do a translation of the whole text is just from the "last word." So it will lose a lot of useful information from the beginning of the text.

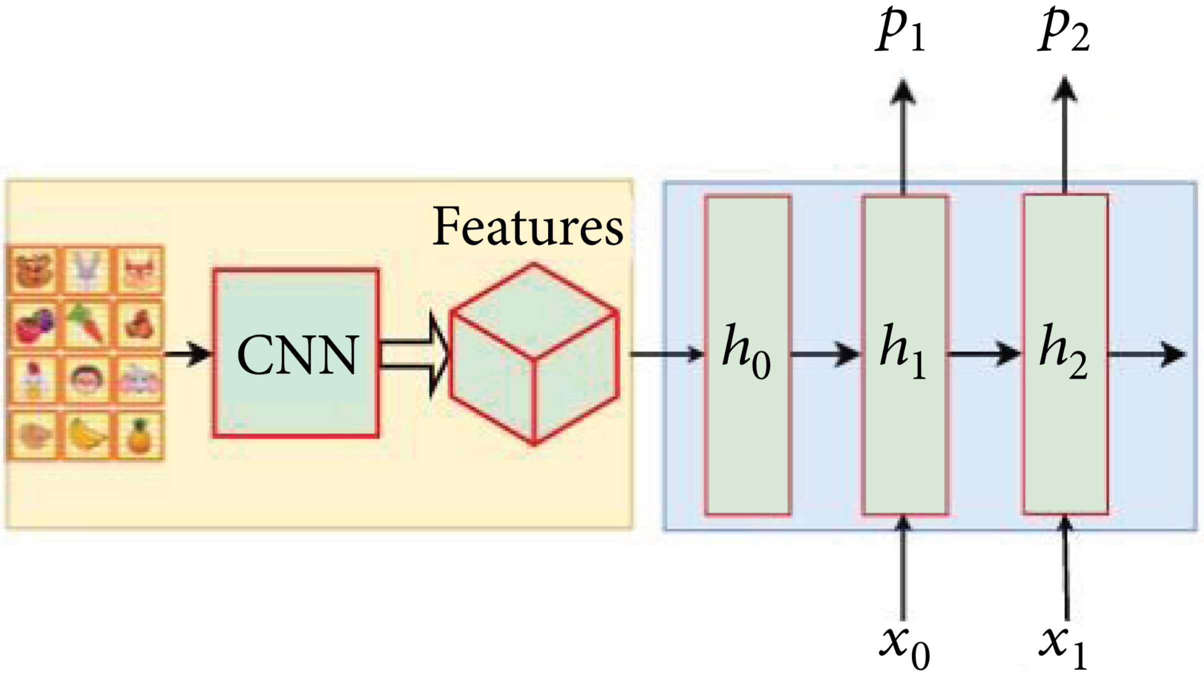


Figure 1: Model without attention[1]

The attention gate can be represented as an addition input for LSTM in Figure 2. The soft attention depends on the previous output of LSTM and extracted features of input image. Soft attention is differentiable and can be trained by the standard method of the backpropagation algorithm [1].

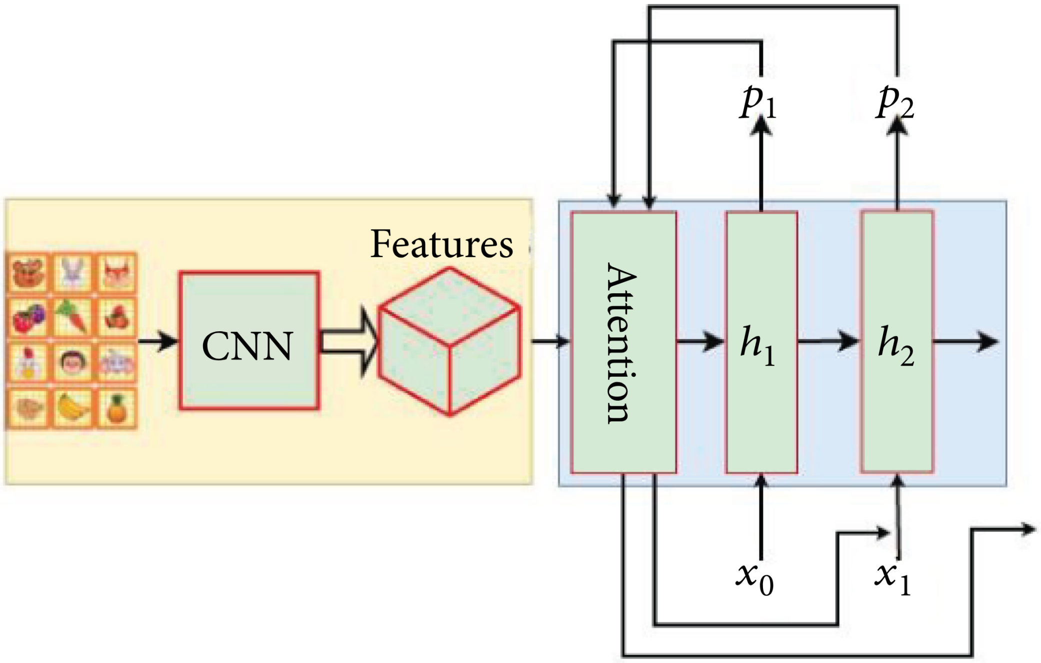


Figure 2: Model with attention[1]

The task of describing images with sentences has also been explored. A number of approaches pose the task as a retrieval problem, where the most compatible annotation in the training set is transferred to a test image or where training annotations are broken up and stitched together [8]. Several approaches generate image captions based on fixed templates that are filled based on the content of the image or generative grammars, but this approach limits the variety of possible outputs.

Most closely related to a log bilinear model that can generate full sentence descriptions for images, but their model uses a fixed window context while our Recurrent Neural Network (RNN) model conditions the probability distribution over the next word in a sentence on all previously generated words [10].

Multiple closely related preprints appeared on Arxiv during the submission of this work, some of which also use RNNs to generate image descriptions [8]. Our RNN is simpler than most of these approaches but also suffers in performance. We quantify this comparison in our experiments.

3.2 Efficient CNN-LSTM based Image Captioning using Neural Network Compression

The AICRL model is an automatic image captioning system using ResNet50 and LSTM with software attention. Its encoder-decoder architecture combines CNN and RNN, extracting visual features using ResNet50 networks [7]. The LSTM language model is used for decoding the vector into sentences, while soft attention is employed to selectively focus attention on specific image parts. The model is fully trainable using stochastic gradient descent and undergoes extensive experiments to determine its structure and fine-tune hyperparameters.

In the encoder-decoder method, the most likely description of the image is determined by maximizing the log-likelihood function of the expression S, considering the corresponding image I and the parameters of the model θ. where θ is the parameter of our model, I is the input image, and S is the correct description. Since S represents a sentence of any length, therefore, a chain rule is usually used to model the joint probability $S_1, ..., S_N$, where N is the length of this particular example.Where the dependence on θ is omitted for convenience. The network training is represented by the pair of (S,I), and we optimize the sum of the log likelihood functions. Over the entire training set using stochastic gradient descent.

$$h_{t+1} = f(h_t, x_t)_{[6]}$$

The likelihood log $_p(S_t|I, S_0, ..., S_{t-1})$ is modelled by a recurrent neural network, where there is a variable number of words that we define up to t-1. The hidden state of RNN (latent memory) h_t is updated after the new input x_t with the nonlinear function f [6].

3.3 Improving Image Captioning with well-known metrics

The model is evaluated using a number of well-known measures, including BLEU, METEOR, and CIDEr. An algorithm called BLEU (Bilingual Evaluation Understudy) measures the accuracy of an n-gramme between generated and reference captions. The length of the reference sentence, the generated word, the uniform weights, and the adjusted n-gramme precisions can all be used for calculating BLEU-N (N=1,2,3,4) scores [2]. The evaluation metric METEOR (Metric for Evaluation of Translation with Explicit Ordering) was first applied to machine translation. METEOR focuses on memory between the generated and ground truth captions in addition to measuring precision [17]. For the purpose of analysing image captioning, CIDEr (Consensus-based Image Description Evaluation) compares created captions to their source texts. This evaluation considers grammaticality and accuracy [6].

We first investigate how the soft attention mechanism affects AICRL. As shown in Table 1, implementing the soft attention method considerably enhances the model performances [5]. Performance in all metrics, including BLEU-4, METEOR, and CIDEr, is improved by the soft attention mechanism.

Table 1: Comparison for AICRL with and without attention[13]

Model	BLEU-4	METEOR	CIDEr
With attention	0.326	0.261	0.872
Without attention	0.262	0.209	0.803

There are also two questions following the generator model's training. The first is whether the model actually creates new descriptions, and the second is whether or not those descriptions are varied, qualitative, and easy for people to understand. In order to include people in the performance evaluation, we also ran another series of studies.

Twenty photographs and the resultant descriptions from the two distinct models are used to create a questionnaire. Participants are asked to assess how well the automatically generated caption describes the photos [2]. The outcomes are shown in Table 2 based on the description that was created from the MS COCO 2014 dataset. The results show that 71 percent of the captions for the model with soft attention are well generated, compared to 54 percent for the model without soft attention. Based on this, we will conduct the following experiments using AICRL and soft attention.

Table 2: Comparison for AICRL with and without attention[2]

Model	Right choosing of generated description
With attention	71 percent
Without attention	54 percent

The study compares AICRL with existing image captioning algorithms and AICRL-VGA16, using VGA16 as a CNN network. Results show that AICRL outperforms other systems in metrics such as BLEU-4, METOER, and CIDEr. The proposed model generates efficient captions and fluent language, while ResNet50 outperforms the VGA16 network, indicating its ability to capture image features well. AICRL achieves good performance by integrating ResNet50, LSTM, and soft attention into a joint model [17].

Tables 3 and 4 show the results based on the Flick8K dataset and MS COCO 2014 dataset.

Table 3: The performance comparison in the Flick8K dataset [14].

Model	BLEU-4	METEOR	CIDEr
Log bilinear[11]	0.177	0.173	—
DVS[20]	0.16	—	—
AICRL-ResNet50	0.262	0.209	0.803
AICRL-VGA16	0.225	0.186	0.743

Table 4: The performance comparison in the MS COCO 2014 dataset [7].

Model	BLEU-4	METEOR	CIDEr
Log bilinear[11]	0.243	0.2	—
DVS[20]	0.23	0.195	0.66
AICRL-ResNet50	0.326	0.261	0.872
AICRL-VGA16	0.295	0.236	0.857

4 Project Design

The model structure presented in the paper is a CNN-LSTM-based image captioning model. As shown in Figure 3, The process typically involves two main steps: Image understanding and Caption generation.

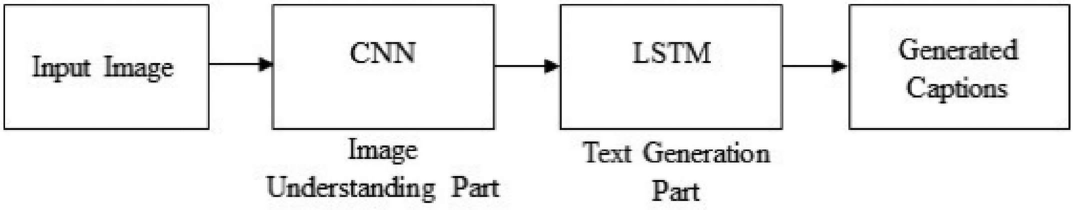


Figure 3: Model Architecture [15]

(1) Image understanding: Convolutional neural networks (CNNs), one type of deep learning model, are used to extract high-level visual information from the input image during the image understanding phase. Understanding the image's content, objects, and relationships relies on these characteristics [1].

(2) Caption generation: Recurrent neural networks (RNNs), in particular long short-term memory (LSTM) networks, are frequently employed to construct captions based on the extracted visual data during the caption creation phase. Based on the previously created words, the RNN models successively predict the following word in the caption, creating a phrase that makes sense and is appropriate for the context [8].

The proposed model consists of two main components: an encoder and a decoder [15].

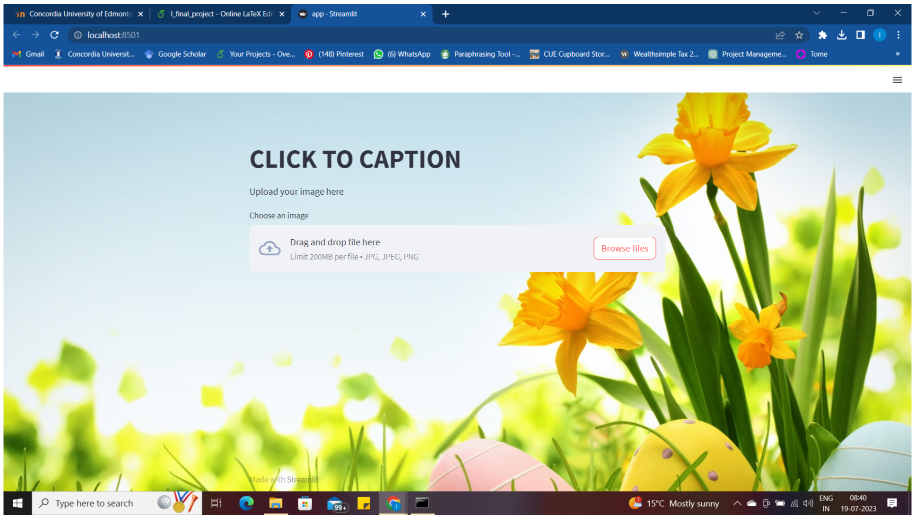
(1) Encoder: The encoder is in charge of sifting through input photos and identifying significant visual cues. The encoder is based on well-known CNN architectures, such as ResNet50 or VGG16, according to the authors. These CNNs can recognise complex visual elements since they have been pre-trained on large-scale picture classification tasks. The encoder transforms the input image into a feature representation that captures the visual information in the image.

(2) Decoder: The decoder makes captions or textual descriptions for the input images using the visual features produced by the encoder. The decoder was created by the authors utilising LSTM networks (Long Short-Term Memory). Recurrent neural networks (RNNs) of the LSTM type can handle sequential data and can recognise context and dependencies over time. The decoder LSTM creates a string of words that make up the image description using the visual attributes as input.

The encoder and decoder are trained independently during training. The VGG16 or ResNet50 architecture is used to pretrain the encoder, while the LSTM network is used to train the decoder from scratch. In order to train the decoder to produce accurate captions, images are fed into the encoder, which subsequently extracts their visual properties and uses them in conjunction with the appropriate captions [16]. The authors use pruning and quantization approaches to condense the model. While quantization lessens the precision of the weights, pruning includes deleting connections or weights from the network that are not as crucial. Without considerably compromising performance, these compression strategies aid in reducing the model size and computational needs.

Overall, the model combines the power of a pretrained CNN encoder (VGG16 or ResNet50) for visual feature extraction and an LSTM decoder for generating captions, providing an end-to-end image captioning system. The compression techniques applied to the model aim to make it more efficient for deployment on resource-limited edge devices.

Here, Figure 4 and Figure 5 demonstrate the use interface of develop system. Figure 4 shows home page of the system which allows users to upload image from their system while Figure 5 shows the result of an uploaded image with a suitable caption.



CLICK TO CAPTION

Upload your image here

Choose an image

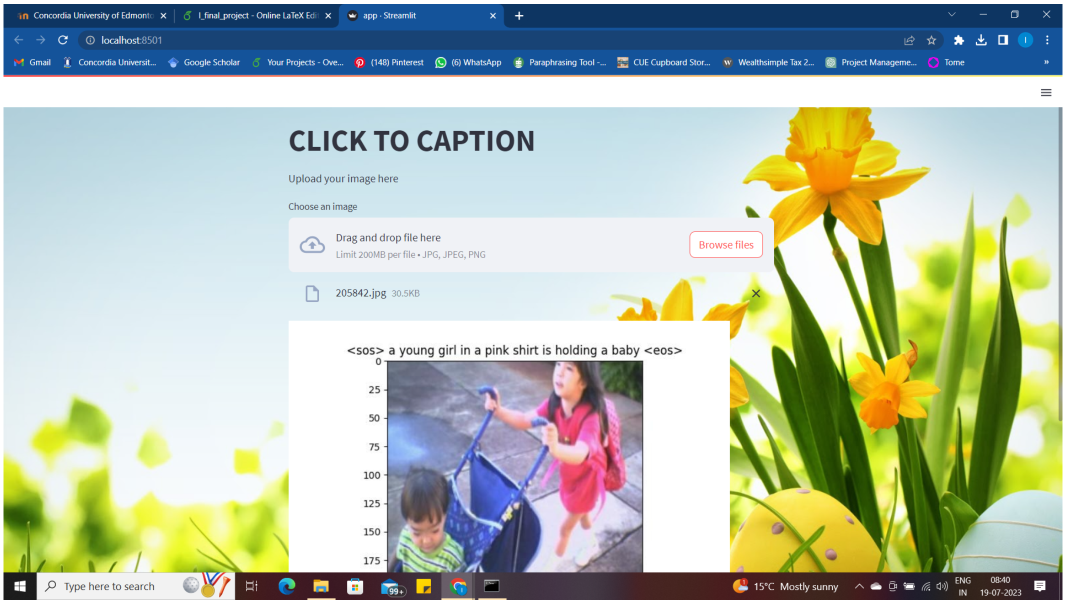
Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

Made with Streamlit

Figure 4: Demo Model



CLICK TO CAPTION

Upload your image here

Choose an image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

205842.jpg 30.5KB

<sos> a young girl in a pink shirt is holding a baby <eos>

Figure 5: Demo Model

5 Body

5.1 Working of Model

Here, Figure 6 represents the working model of the developed system. There are different components of working model like, Encoder, Decoder, Embedding layer which are explain below.

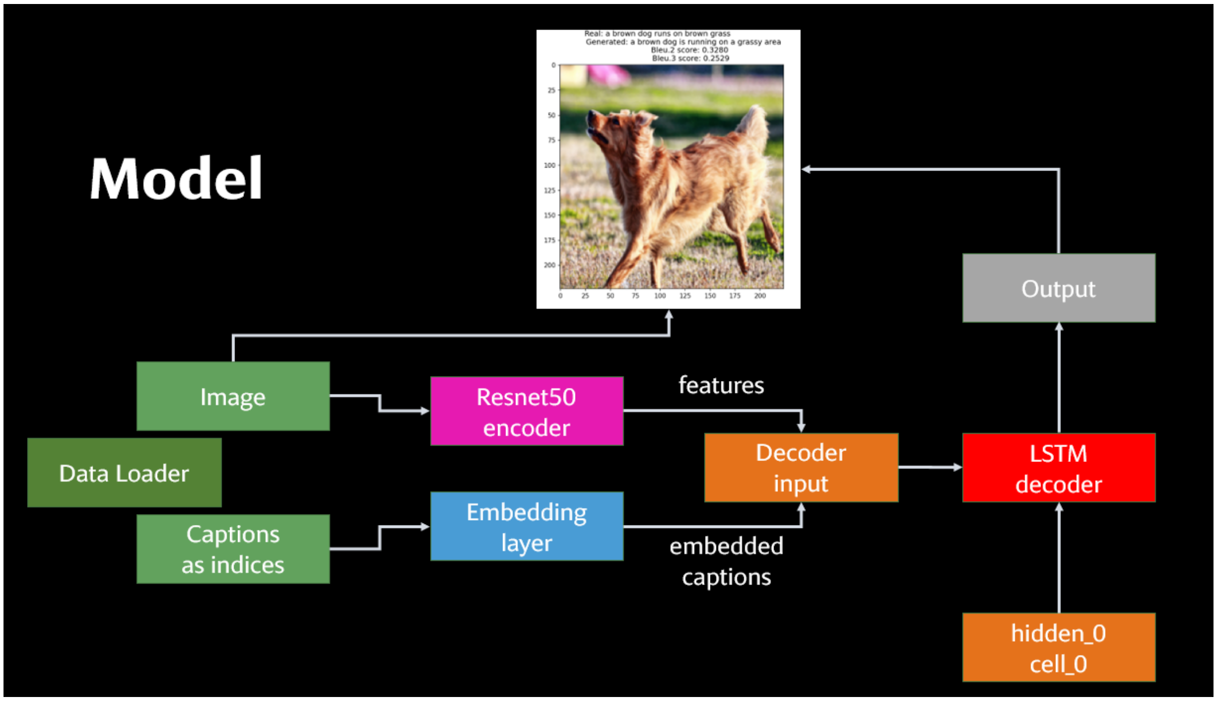


Figure 6: Working Model [4]

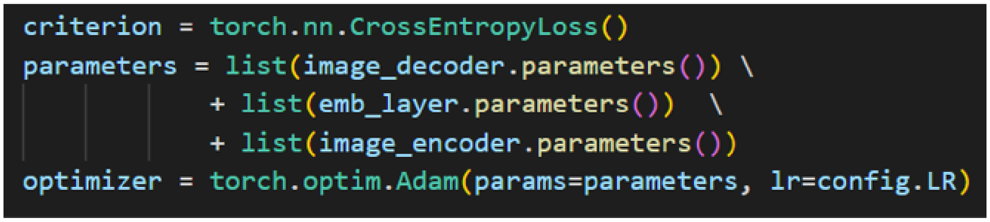
(1) Encoder: Image encoder to obtain features from images, Contains pretrained Resnet50 with last layer removed and a linear layer as final classifier, Final output dimension of features is (IMAGE-EMB-DIM)

(2) Embedding layer: Used to obtain embedded representation (as a dense vector) of captions of dimension (WORD-EMB-DIM), When training the model, the embedding layer is updated to learn better word representation through the optimization process.

(3) Decoder: Contains LSTM layer and a linear layer as final classifier whose output is of dimension (VOCAB-SIZE), The input for the LSTM layer is the concatenation of features from the encoder and the embedded captions from the embedding layer, Hidden and cell states are zero initialized.

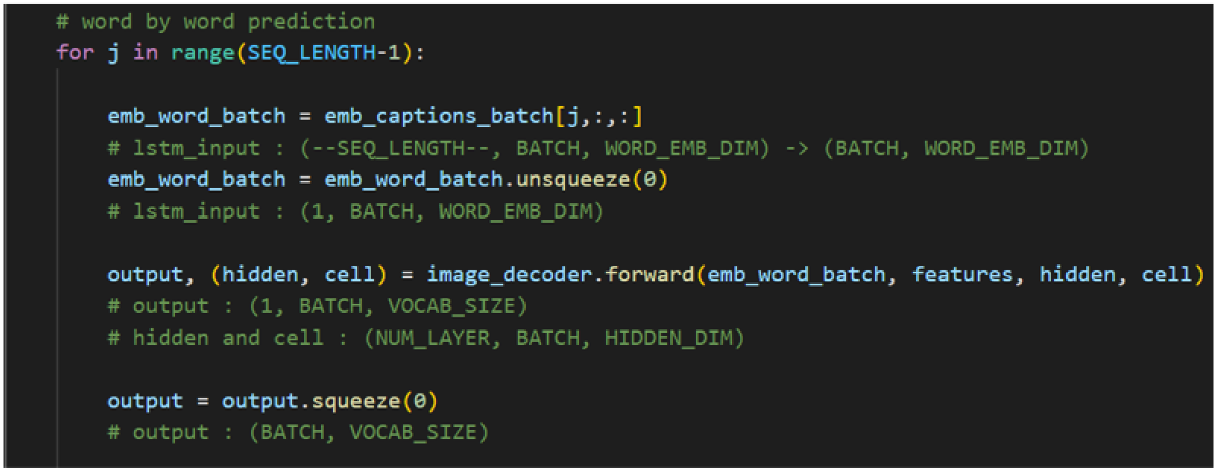
LSTM generates each word of the caption on at a time, based on the previous word (index), image features and hidden and cell states. It can potentially result in more diverse and interesting captions. However, this approach can be slower and more computationally expensive, especially if the length of the captions is long.

There are different parameters used to develop this system, which are mentioned below in Figure 7 and Figure 8. Figure 7 represents parameters of Encoder, Embedding layer and Decoder. While, Figure 8 is representation of hyper-parameters which used to predict captions.



```
criterion = torch.nn.CrossEntropyLoss()
parameters = list(image_decoder.parameters()) \
              + list(emb_layer.parameters())  \
              + list(image_encoder.parameters())
optimizer = torch.optim.Adam(params=parameters, lr=config.LR)
```

Figure 7: Other parameters for training the model



```
# word by word prediction
for j in range(SEQ_LENGTH-1):

    emb_word_batch = emb_captions_batch[j,:,:]
    # lstm_input : (--SEQ_LENGTH--, BATCH, WORD_EMB_DIM) -> (BATCH, WORD_EMB_DIM)
    emb_word_batch = emb_word_batch.unsqueeze(0)
    # lstm_input : (1, BATCH, WORD_EMB_DIM)

    output, (hidden, cell) = image_decoder.forward(emb_word_batch, features, hidden, cell)
    # output : (1, BATCH, VOCAB_SIZE)
    # hidden and cell : (NUM_LAYER, BATCH, HIDDEN_DIM)

    output = output.squeeze(0)
    # output : (BATCH, VOCAB_SIZE)
```

Figure 8: Word generation (LSTM output)

The hyperparameters are defined below:

1. ‘DEVICE‘: This hyperparameter specifies the device on which the model will be trained. In this case, it is set to ‘"cpu"‘, indicating that the model will be trained on the CPU. However, it can be changed to ‘"cuda"‘ to train on a GPU if available.

2. ‘BATCH‘: This hyperparameter determines the batch size used during training. The batch size is set to 32, meaning that 32 samples will be processed together in each training iteration.

3. ‘EPOCHS‘: The number of training epochs. Each epoch represents one full pass of the training data through the model. In this case, the model will be trained for 5 epochs.

4. ‘VOCAB-FILE‘: The filename of the file containing the vocabulary, which maps words to their corresponding indices. The file is expected to be in the same directory as the script.

5. ‘VOCAB-SIZE‘: The size of the vocabulary, which is set to 5000. This indicates that the model will work with the 5000 most frequent words in the dataset.

6. ‘NUM-LAYER‘: The number of layers in the decoder. The value is set to 1, meaning there is only one LSTM layer in the decoder.

7. ‘IMAGE-EMB-DIM‘: The dimension of the image embeddings. The value is set to 512, meaning that images will be encoded into 512-dimensional vectors.

8. ‘HIDDEN-DIM‘: The dimension of the hidden state in the LSTM layers. The value is set to 1024, indicating that the LSTM will have 1024 hidden units.

9. ‘LR‘: The learning rate used during training. The value is set to 0.001, which controls the step size at which the model's parameters are updated.

10. ‘EMBEDDING-WEIGHT-FILE‘, ‘ENCODER-WEIGHT-FILE‘, ‘DECODER-WEIGHT-FILE‘: These parameters specify the filenames of the pre-trained weights for the embedding layer, encoder, and decoder, respectively. These weights will be loaded if ‘LOAD-WEIGHTS‘ is set to ‘True‘ in the main script.

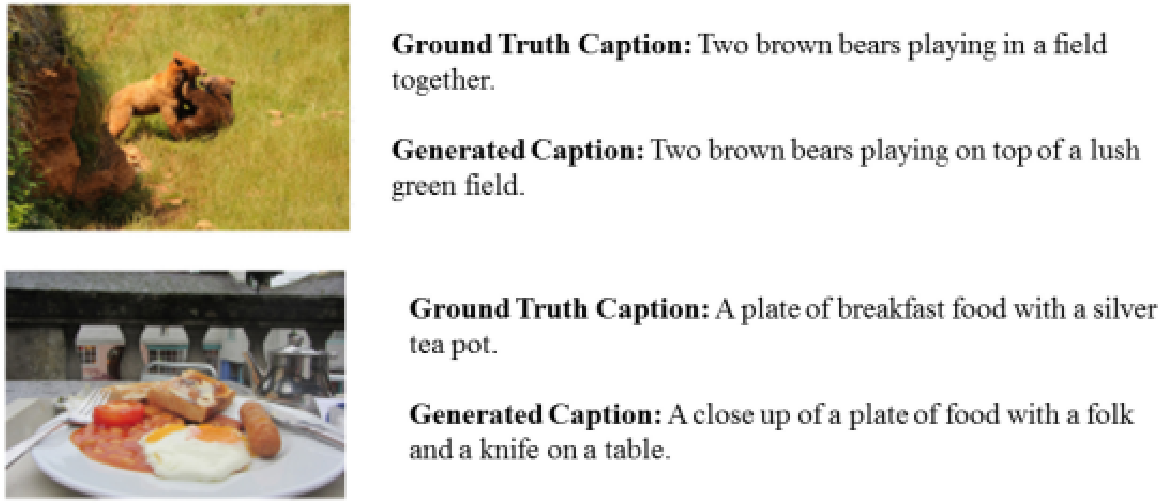
11. ‘ROOT‘: The root directory path where the images are located. The script will look for the specified ‘image-file‘ (from the command-line argument) in this directory.

5.2 Datasets

Image captioning methods are trained, tested, and evaluated using various datasets. Three popular datasets are Flickr8K, Flickr30K, and MS COCO [18]. Sample images with captions generated by these methods are shown, and evaluation metrics are used to measure their quality compared to the ground truth.

5.2.1 MS COCO Dataset

The Microsoft COCO Dataset is a large dataset for image recognition, segmentation, and captioning, featuring object segmentation, context recognition, multiple objects per class, over 300,000 images, 2 million instances, 80 object categories, and five captions per image [7]. It is used in image captioning methods, as shown in Figure 9.



Ground Truth Caption: Two brown bears playing in a field together.

Generated Caption: Two brown bears playing on top of a lush green field.

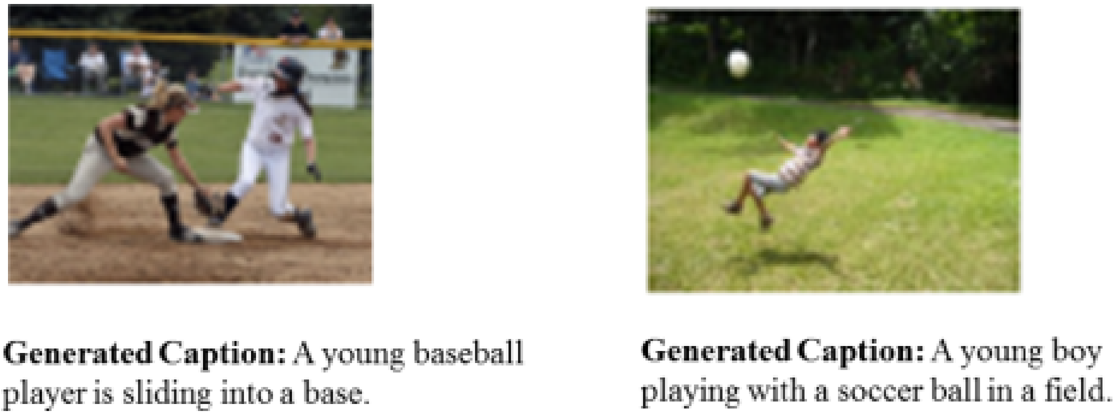
Ground Truth Caption: A plate of breakfast food with a silver tea pot.

Generated Caption: A close up of a plate of food with a folk and a knife on a table.

Figure 9: sample images from the MS COCO dataset[7]

5.2.2 Flickr30K Dataset

Flickr30K is a dataset for automatic image description and grounded language understanding, containing 30K images and 158K captions from human annotators. Researchers can choose their own training, testing, and validation numbers. The dataset includes detectors for common objects, a color classifier, and a bias towards larger objects [14]. Which is used in image captioning methods, as shown in Figure 10.



Generated Caption: A young baseball player is sliding into a base.

Generated Caption: A young boy playing with a soccer ball in a field.

Figure 10: sample images from the Flickr30K dataset[14]

5.3 Predict Sample Images

In Experiments, We tried different images to examine developed system. Figure 11,12,13,14 are the result of predicted images by the system. Figure 12 and Figure 13 are pretested images from known datasets. While Figure 14 and Figure 15 are random images from the user's system.

We can observe that Images from the dataset are captioned well by the developed system with good BLEU score. For Figure 11 (Bleu Score: 0.25) and for Figure 12 (Bleu score: 0.33), that means captions are generated by the system are closed to real captions. Although, In the case of random images as shown in Figure 13 and Figure 14, system generate good captions with low accuracy. In the future we are planning make it more accurate by doing necessary changes in hyper-parameters.

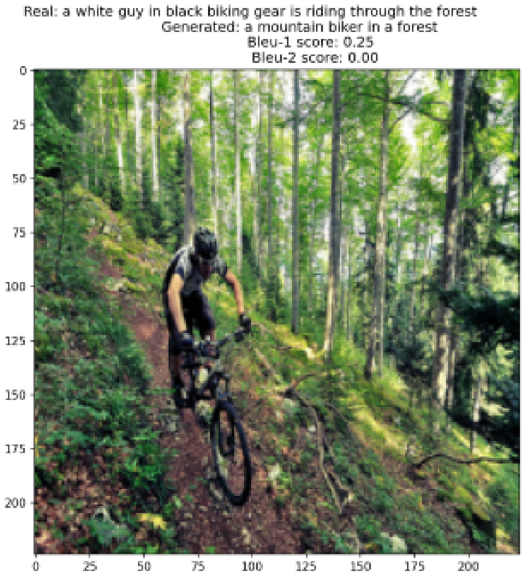

Real: a white guy in black biking gear is riding through the forest
Generated: a mountain biker in a forest
Bleu-1 score: 0.25
Bleu-2 score: 0.00
0
25
50
75
100
125
150
175
200
0
25
50
75
100
125
150
175
200

Figure 11: Sample Image-1 [7]

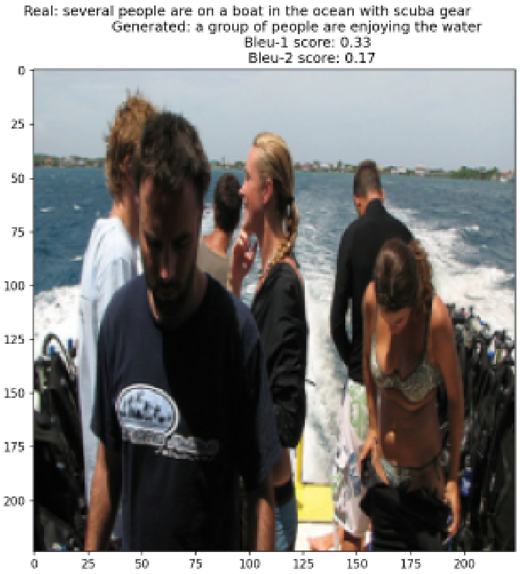

Real: several people are on a boat in the ocean with scuba gear
Generated: a group of people are enjoying the water
Bleu-1 score: 0.33
Bleu-2 score: 0.17
0
25
50
75
100
125
150
175
200
0
25
50
75
100
125
150
175
200

Figure 12: Sample Image-2 [14]

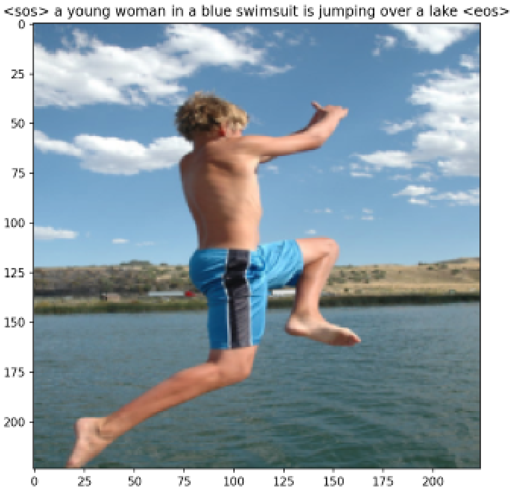

<sos> a young woman in a blue swimsuit is jumping over a lake <eos>
0
25
50
75
100
125
150
175
200
0
25
50
75
100
125
150
175
200

Figure 13: Sample Image-3

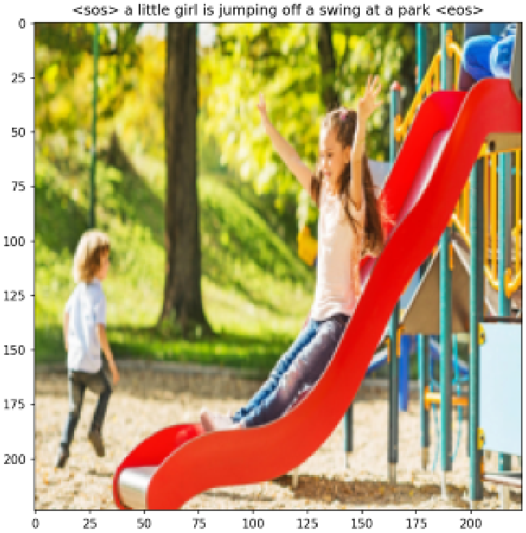


Figure 14: Sample Image-4

5.4 Experiments Results

Here, Figure 15 shows the training and validation loss.

The left graph is comparing the accuracy of prediction on training vs validation data. We can see that the accuracy are increasing with every next epoch until 4th epoch after which the accuracy of the validation data is constant.

The right graph is comparing the Losses of prediction on training vs validation data. We can see that the loss decreases with every epoch until 3,4 epochs after which the loss increases for the validation data. Using these graphs, we can know the ideal epoch number to be used in the model i.e. 3,4.

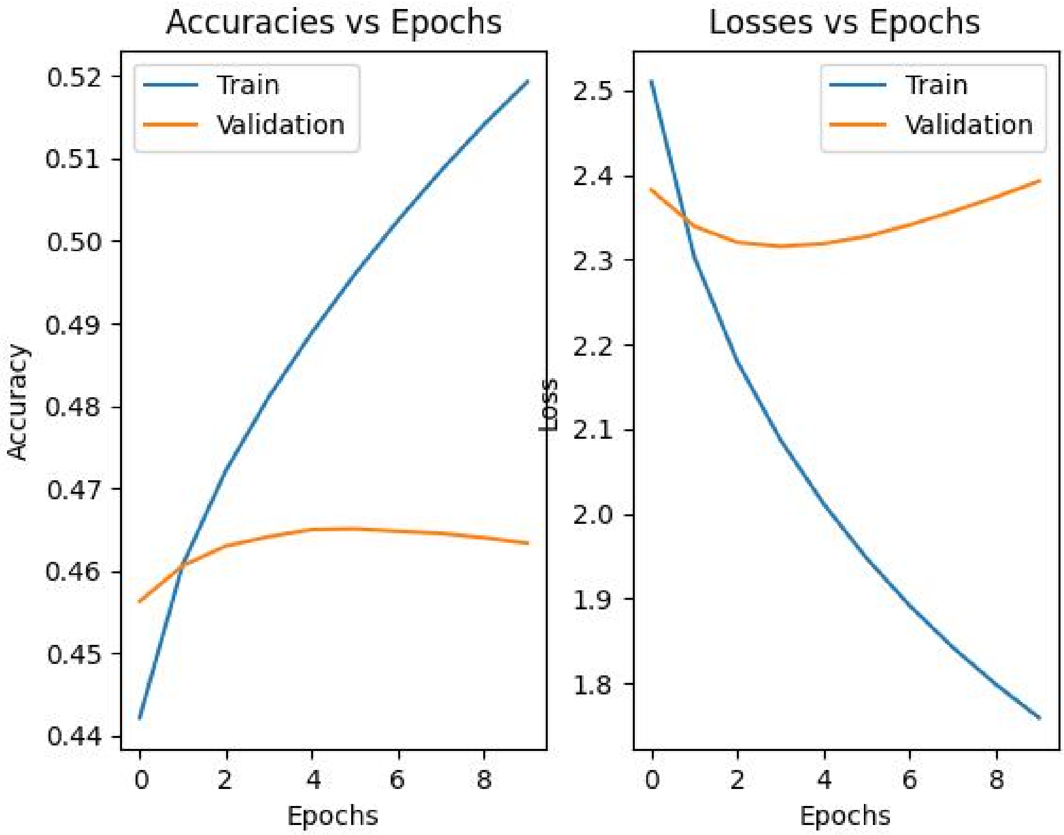


Figure 15: Comparison Graph [4]

5.5 Future Scope

Although the predicted captions were relevant for the majority of the photographs, they sometimes did not reflect the content of the picture. By training the model on a larger dataset, this problem can be resolved.This was a basic project and is capable of a lot of improvements. Some of the modifications that can improve the model are as follows:

Hyper parameter tuning (for example, batch size, dropout rate, number of layers, learning rate, batch normalization etc.).

Use of cross validation set to check overfitting

6 Conclusions

In this research, we have introduced a single joint model based on ResNet50 and LSTM with software attention for automatic image captioning. One encoder-decoder architecture was used for creating the suggested model. To compress an image into a small representation that can be represented graphically, we used ResNet50, a convolutional neural network. The decoder for the descriptive sentence was then chosen as a language model LSTM. In the meanwhile, we combined the LSTM with the soft attention model so that learning may be targeted at a specific area of the image to enhance performance. Using stochastic gradient descent, which facilitates training, the entire model is fully trainable. The results of the experimental assessments show that the suggested model is capable of producing quality image captions automatically.

References

[1] Automatic image captioning based on resnet50 and lstm with soft attention. *Wireless Communications and Mobile Computing*, 2020:7, 2020.

[2] S. Banerjee and 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*, pages 65–72, 2005.

[3] C. Chen, S. Mu, W. Xiao, Z. Ye, L. Wu, and Q. Ju. Improving image captioning with conditional generative adversarial nets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8142–8150, Jul. 2019.

[4] P. Diwakar. Automatic image captioning using deep learning. *Proceedings of the International Conference on Innovative Computing Communication (ICICC)*, Apr. 2021.

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[6] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 15–29. Springer, 2010.

[7] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[8] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[9] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

[10] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *International conference on machine learning*, pages 595–603. PMLR, 2014.

[11] A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648, 2007.

[12] J.-Y. Pan, H.-J. Yang, P. Duygulu, and C. Faloutsos. Automatic image captioning. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 3, pages 1987–1990 Vol.3, 2004.

[13] K. Papineni, S. Roukos, T. Ward, and 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*, pages 311–318, 2002.

[14] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[15] H. Rampal and A. Mohanty. Efficient cnn-lstm based image captioning using neural network compression, 2020.

[16] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[17] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[18] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[19] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[20] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.