A Deep Dive of Numerous Deep Learning Approaches for Image Captioning

Bharath Krishnamurthy
Computer Science Department,
University Of North Texas
Denton, TX, 76201, USA
bharathkrishnamurthy@my.unt.edu

Arun Kumar Reddy Kandula
Computer Science Department,
University Of North Texas
Denton, TX, 76201, USA
arunkumarreddykandula@my.unt.edu

Venkat Goutham Singh Kshatri Thakur
Computer Science Department,
University Of North Texas
Denton, TX, 76201, USA
gouthamkshatrithakur@my.unt.edu

ABSTRACT

In recent years, image captioning has emerged as a challenging yet essential task in the field of computer vision and artificial intelligence. The ability to automatically generate descriptive and contextually relevant captions for images holds great potential for applications in image understanding, content retrieval, and accessibility. Our project presents a comprehensive exploration of image captioning using the Flickr 8k dataset, employing deep learning techniques implemented through TensorFlow and PyTorch frameworks. [7] The study aims to compare the performance of these frameworks in the context of image captioning, evaluating their strengths and weaknesses. Additionally, our model incorporates an attention mechanism, enabling it to selectively focus on the most essential features of the images during both the image feature extraction phase using Convolutional Neural Networks (CNNs) and the sequential language generation phase using Long ShortTerm Memory networks (LSTMs). The integration of attention mechanisms enhances the model's ability to capture relevant details and improves the overall quality of generated captions. We will leverage state-of-the-art deep neural networks, combining CNNs for image feature extraction and LSTMs for sequential language generation.

Our experiments will be conducted on the Flickr 8k dataset, known for its diverse and well-annotated images. We will perform a detailed analysis of the model's ability to generate coherent and contextually relevant captions, exploring the impact of architectural choices, hyperparameter settings, and attention mechanism on performance. The comparative study between TensorFlow and PyTorch, with the inclusion of the attention mechanism, will provide valuable insights into the strengths and trade-offs of each framework in the specific domain of image captioning. [8]

KEYWORDS

Image Captioning, transfer learning, LSTMs, Deep Learning, Attention.

Bharath Krishnamurthy, Arun Kumar Reddy Kandula, and Venkat Goutham Singh Kshatri Thakur. 2023. A Deep Dive of Numerous Deep Learning Approaches for Image Captioning.

1 INTRODUCTION

The proliferation of digital images across various online platforms has underscored the need for advanced techniques in image understanding and interpretation. Image captioning, the task of automatically generating descriptive textual explanations for images, has garnered significant attention due to its potential applications in content retrieval, accessibility, and human-computer interaction. The seamless integration of computer vision and natural language processing have paved the way for sophisticated deep-learning models capable of discerning intricate details in images and articulating them in human-like language. [17]

This project focuses on image captioning using the Flickr 8k dataset, a benchmark dataset renowned for its diverse set of images, each annotated with multiple captions. Our approach involves the integration of convolutional neural networks (CNNs) for image feature extraction and Long Short-Term Memory networks (LSTMs) for sequential language generation, along with the incorporation of an attention mechanism. This multimodal architecture enables our model to capture both visual and semantic information, providing a holistic understanding of the image content.

To conduct a comprehensive analysis, we will implement our model using two prominent deep learning frameworks—TensorFlow and PyTorch. The choice of framework plays a crucial role in model development and training, and a comparative study between these two frameworks, considering the attention mechanism, can shed light on their respective strengths and limitations in the context of image captioning.

2 OUR APPROACH

To achieve our goals in this project, we will primarily focus on three approaches. Each of the three approaches is almost given similar weightage, but our primary focus is the first model upon which the other two models are varied. In the first model, we will use TensorFlow and Keras to build a network from scratch to work with the image captioning visual and textual data. The first step is to ensure that we clean the text data and remove any unnecessary information irrelevant to our project. We will also convert all the words into a lower case to allow the model to learn better initially.

We will utilize an encoder and decoder network for specific tasks. The encoder architecture will use a transfer learning model of an inceptionv3 network and we will fine-tune the last couple of layers to create our vector encoding for all our images. We will then use these vector encodings with our fine-tuned transfer learning model that stores the appropriate information for our image data. Once we

create image encodings, we will also create the appropriate word index for the natural language processing task and choose a specific maximum length for creating all the word embeddings accordingly.

The final step of our approach is to construct our own custom decoder network with LSTMs capable of performing the image captioning task by decoding the embeddings. We will save the best models and utilize them for predicting the images. Apart from this approach, we also plan to try two other different comparisons for our models. We will make a complete PyTorch implementation of the same project with the ResNet50 architecture as the encoder. Apart from that, we also plan to construct a third model for comparison with our initial model. We will use the attention mechanism that will be incorporated to selectively focus on the most essential features of the project.

3 RELATED WORKS

The Show, Attend and Tell influential paper by Xu et al. (2015) [18] introduces a pivotal concept to image captioning—visual attention. The model employs a combination of convolutional neural networks (CNNs) for image representation and long short-term memory (LSTM) networks for language generation. The introduction of a dynamic attention mechanism allows the model to selectively focus on different regions of the image during the caption generation process. This innovative approach significantly improves the alignment of generated words with relevant visual features, enhancing the overall quality of image captions.

Another popular research paper by Anderson et al. (2018) [2] proposes a two-step attention mechanism—combining bottom-up and top-down attention—for image captioning and visual question answering. The bottom-up attention involves extracting salient image regions and features, while the top-down attention guides the model to focus on specific regions relevant to the current word generation. This dual-attention approach enhances the model's ability to capture fine-grained details, contributing to improved performance on complex tasks such as image captioning and visual question answering.

The image transformer network [7] presents a novel application of transformer architectures, originally designed for sequence-to-sequence tasks in natural language processing, to image captioning. This work showcases that transformer models, when adapted to process image data, can achieve competitive results compared to traditional CNN-LSTM-based approaches. Leveraging the self-attention mechanism inherent in transformers, the model captures long-range dependencies in both visual and textual information, offering a fresh perspective on achieving state-of-the-art performance in image captioning

4 DATASETS

We will be using the Flicker 8k dataset from Kaggle, that was introduced in Hodosh et al. (2013) [9], this dataset contains 8,000 images with 5, single-sentence captions for each image. The images were manually chosen from 6 different Flickr groups that did not include any well-known people and locations. The captions were generated by people from the United States who passed a brief spelling and grammar test. For a single image multiple conceptual descriptions that depicted scenes, situations, events, and entities were written

(a)

- A child in a pink dress is climbing up a set of stairs in an entry way.
- A girl going into a wooden building.
- A little girl climbing into a wooden playhouse.
- A little girl climbing the stairs to her playhouse.
- A little girl in a pink dress going into a wooden cabin.

(b)

Figure 1: (a) Sample image from the Flicker dataset, (b) Corresponding captions of the image

by the annotators. Because there are different ways to describe an entity or action (man vs bike rider, doing tricks vs jumping).

From Figure 10, we can see that the corresponding captions of the image are diverse and highlight different versions of the same image. This diversity in the captions and image dataset will help in making the model robust and generalized.

5 METHODOLOGY

5.1 TensorFlow and Keras Model:

The project outlines an approach for developing an image captioning model using TensorFlow and Keras. The goal is to create a model

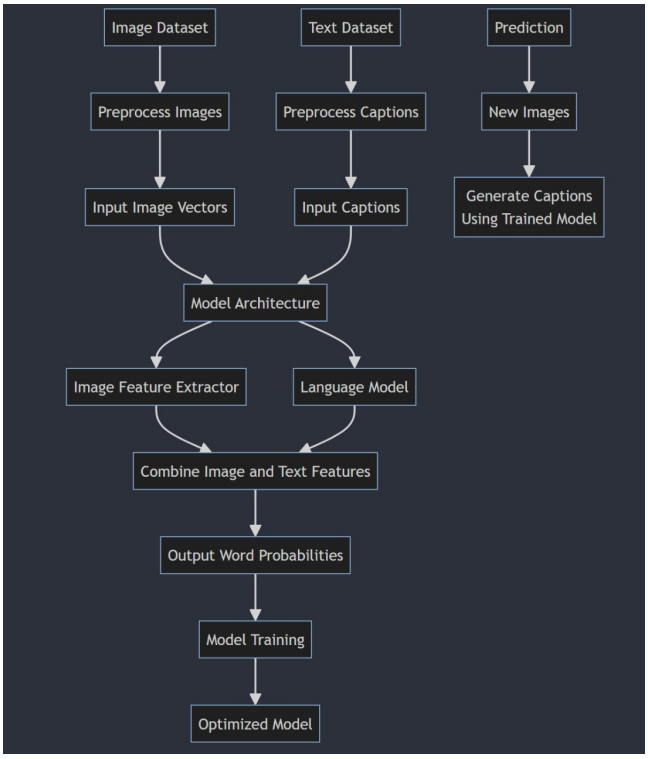


Figure 2: The TensorFlow Keras Worflow for Image Captioning.

that can automatically generate captions to describe images. The first step is collecting an image dataset and corresponding captions to use for training. In this case, the Flickr8K dataset is used which contains over 8,000 images paired with 5 captions each.

TensorFlow and Keras are used as the core deep learning frameworks for building and training the image captioning model. TensorFlow provides the backend engine for construction and executing the neural networks. Keras provides a convenient API and layers to build models on top of TensorFlow.

The images and captions then need to be pre-processed. For the images, the InceptionV3 model is used to extract image features. For the captions, the text is cleaned by converting to lowercase, removing punctuation, etc. The words are mapped to numerical indices to create the vocabulary. The InceptionV3 model is used as a pre-trained image feature extractor. InceptionV3 was trained for image classification on a large dataset.

An embedding layer is used to represent the caption words as 200-d vectors before inputting to the LSTM decoder. GloVe embeddings are initialized as the word encodings. This captures semantic similarity between words. A data generator is created to feed image vectors and caption sequences to the model during training. The captions are padded to a fixed length and input/output sequence pairs are created from each caption.

The final model architecture combines the InceptionV3 image encoding with the LSTM decoder network. It is trained end-to-end so these components are optimized together to improve the image captioning performance. Using transfer learning, the trained weights are reused up to the last layer to encode new images into a 2048-d vector capturing semantic features. This encoding is fed to the decoder. The model has an encoder-decoder architecture.

The image vector from InceptionV3 is the encoded representation of the image. This is input to the decoder LSTM network to generate the caption text sequence. The model architecture uses LSTM and dense layers. It takes the image vector and caption sequence as input to predict the next word in the caption. An embedding layer is used to encode the caption words as vectors.

The model is trained for 50 epochs due to time constraints but more would likely improve performance. The decoder generates the captions using a sequence-to-sequence model with an LSTM network. It is trained to predict the next word given previous words and the image encoding from the InceptionV3 encoder. This allows it to generate the caption word-by-word.

After training, the model can be used to predict captions for new images. The results show the model is able to generate relatively accurate captions after just 50 epochs, but has room for improvement with more training data and epochs.

In summary, the project provides a full walkthrough for developing an image captioning model by preprocessing data, constructing a model architecture, training the model, and making predictions. Multiple improvements are proposed, demonstrating this is an active area of research with room to expand.

Here are some key equations for the image captioning model architecture, along with brief descriptions for each.

5.1.1 Image Encoding. The InceptionV3 model encodes each image I into a 2048-dimensional vector $\mathbf{v}$:

$$\mathbf{v} = \text{InceptionV3}(\mathbf{I})$$

5.1.2 Word Encoding. An embedding matrix $E \in \mathbb{R}^{V \times D}$ is created to encode each word in the vocabulary V into a D-dimensional vector:

$$\mathbf{w} = \mathbf{E}[\text{word2idx}[w]]$$

where `word2idx` is a dictionary mapping words to indices. Typically, $D = 200$ or 300 based on the embedding dimensionality.

5.1.3 Sequence Encoding. The LSTM encodes the sequence (sentence) one word at a time, encoding the previous context $\mathbf{c}_{t-1}$ and the current word w_t into the current context vector $\mathbf{c}_t$:

$$\mathbf{c}_t = \text{LSTM}(\mathbf{w}_t, \mathbf{c}_{t-1})$$

5.1.4 Output Probability. Finally, the output word probability distribution is computed, conditioning on the image encoding $\mathbf{v}$ and context encoding $\mathbf{c}_t$ from the LSTM decoder:

$$P(w) = \text{Softmax}(\mathbf{W}[\mathbf{v}; \mathbf{c}_t] + \mathbf{b})$$

The model is trained to maximize $P(w)$ for the ground truth next word in the sentence.

5.2 TensorFlow with Attention:

Enhancing Model Attention with Convolution Block Attention Module (CBAM): CBAM is an attention mechanism designed to

increase the convolution neural network's focus on the important features in terms of both channels and spatial information by increasing the weights of those important features. This done by incorporation of the two key components Channel Attention and Spatial Attention.

Channel Attention (CA) is one of the main components in the Convolutional Block Attention Module (CBAM), it is designed to augment the effectiveness of convolutional neural networks (CNNs) by refining channel-wise feature representations. The motivation behind CA lies in recognizing that different channels within a feature map capture distinct and diverse information. Global Average Pooling (GAP) is employed to aggregate the channel-wise information, reducing it to a condensed representation. Subsequently, fully connected layers act as adaptive filters, discerning the significance of each channel through learned attention weights. The reshaped attention weights are then multiplied with the original feature maps, dynamically calibrating the contribution of each channel. By integrating CA, CBAM ensures that the model concentrates on the most informative channels, fostering a more discriminative and context-aware feature representation. Figure 3 shows the architecture of the CA, where the input features are passed to the Maxpool and Avgpool layers, and the output of these layers is passed to the Multi-layer Perceptron which helps us identify the important channels and of those 2 layers. Now the enhanced output of the Maxpool and Avgpool are combined to produce the final output that has the information on the channels that are contributing more to the prediction so we can make our network focus more on these important channels rather than all the channels.

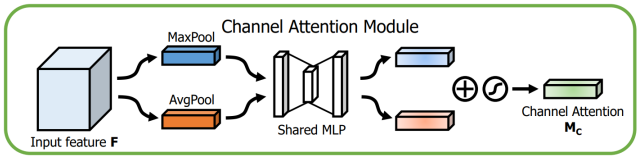


Figure 3: Channel Attention Architecture

Spatial Attention (SA) is also a main component of CBAM, designed to enhance the spatial focus of convolutional neural networks. Recognizing that different spatial regions within each channel contribute variably to the overall understanding of an image, SA introduces a mechanism to selectively amplify informative regions while suppressing irrelevant ones. Through a convolutional operation, spatial dependencies within each channel are captured, allowing the model to discern intricate patterns and spatial relationships. The subsequent activation function generates attention weights for each spatial location, signifying their relevance. By element-wise multiplication with the original feature maps, the spatial attention weights dynamically emphasize key regions, enabling the model to attend to specific spatial details crucial for accurate image understanding and captioning. From Figure 4, we can see that the channel refined features are passed as input to Maxpool and Avgpool layers which are convoluted and passed to and activation layer where the magic happens. This activation layer tries to capture the spatial regions that are helping in predicting by increasing the weights of the sections of those important regions, producing a spatially important regions called as spatial attention.

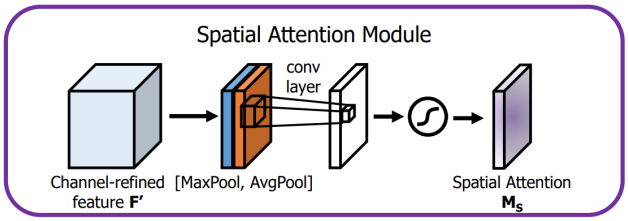


Figure 4: Spatial Attention Architecture

Figure 5 shows the overview of the CBAM architecture, where the Channel Attention and Spatial Attention modules are included to produce features that are enhanced both in terms of channel and spatial details. The input features are passed to the Channel Attention module to generate the channel attention features which are then combined with the input features by pair-wise multiplication to have channel-enhanced features which are then passed to the Spatial Attention module to generate the spatial attention map and combined with the channel-enhanced features by a pair-wise multiplication again to finally obtain feature output that are both enhanced with the channel and spatial information.

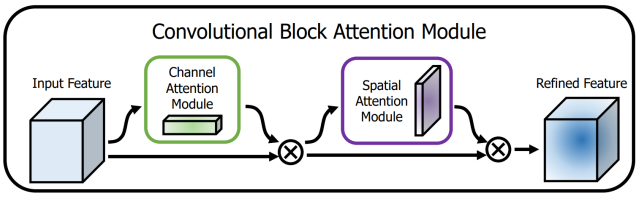


Figure 5: Overview of CBAM

For image captioning problems, attention will play an important factor in determining the important features of the Image both in terms of channel and spatial information. Traditional attention mechanisms focus on spatial information, but CBAM introduces channel-wise attention, providing a more comprehensive approach to feature selection.

To leverage the benefits of CBAM, we enhance our existing image captioning model by integrating CBAM modules. The CBAM modules are applied to the feature maps extracted from the InceptionV3 model, specifically targeting the last convolutional layer.

5.3 PyTorch with ResNet Model:

In the ever-evolving field of computer vision and natural language processing, the development of an image captioning model represents a significant stride towards bridging visual data with textual interpretation. This report outlines the implementation of such a model using PyTorch, a popular deep learning library, and the ResNet model, known for its efficacy in image classification tasks. The goal of this project is to create a model capable of generating accurate and contextually relevant captions for images, a task that has wide-ranging applications from aiding visually impaired users to enhancing image indexing for search engines.

Our model's training was conducted on a robust dataset, akin to the renowned Flickr8K dataset, comprising thousands of images

each paired with descriptive captions. The initial step involved meticulous preprocessing of this data. For images, we employed the ResNet-50 architecture, a variant of the ResNet model renowned for its deep layers and remarkable performance in image recognition tasks, to extract salient features. On the textual front, captions underwent a cleaning process, stripping away punctuation and transforming all text to lowercase to maintain consistency. Subsequently, these captions were tokenized, converting words into numerical indices, thus creating a comprehensive vocabulary essential for the model's learning process.

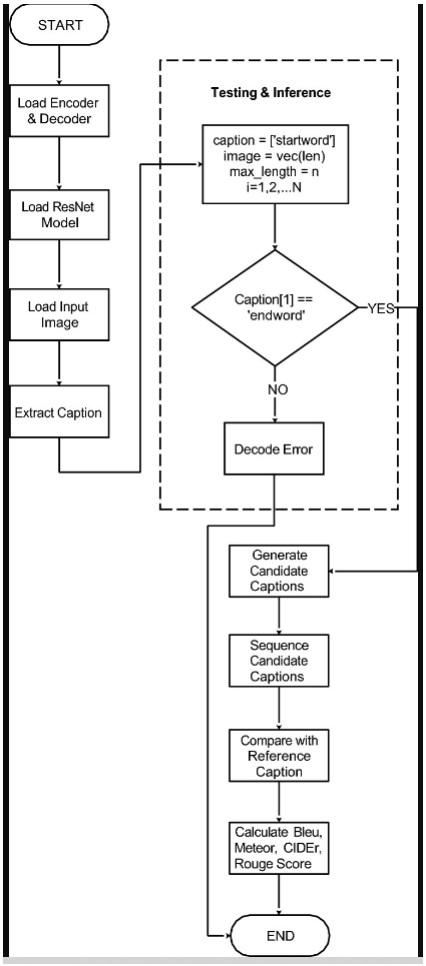


Figure 6: The Pytorch Worflow for Image Captioning.

The core of our implementation lies in the sophisticated architecture that harmoniously integrates image and text processing. The ResNet-50 model serves as the backbone for extracting high-dimensional feature vectors from input images. These vectors encapsulate the critical visual elements necessary for caption generation. Complementing this, we implemented a recurrent neural network, possibly an LSTM (Long Short-Term Memory) network, to process the sequential text data. The embedding layer, critical in text-based models, transforms words into meaningful vector representations, capturing the nuanced relationships between different terms. This dual-pathway architecture ensures a holistic understanding of both visual and textual inputs, setting the stage for effective caption generation.

The training regimen was meticulously designed to optimize the model's performance. We utilized common loss functions and optimizers suitable for the task at hand, iterating over numerous epochs. This phase was crucial in fine-tuning the model's parameters, ensuring a harmonious balance between the image feature extractor and the text processor. The integration of image and caption data during training allowed the model to learn the intricate associations between visual content and its corresponding textual description.

Post-training, the model exhibited promising capabilities in generating captions that were not only contextually apt but also linguistically coherent. While the captions generated after the initial 50 epochs were impressive, they indicated potential areas for enhancement. The model's ability to interpret and describe complex scenes was noteworthy, yet there were instances where nuances in the images were overlooked, suggesting room for refinement in future iterations.

6 EVALUATION METRICS

For the evaluation of the generated text, the following are the widely used metrics: BLEU (Bilingual evaluation understudy), proposed by Papineni et al. (2002) [13], compares the similarity of the generated text based on a set of reference texts. The machine-generated text is split into multiple text segments and compared with a set of reference texts and the scores are averaged to obtain the final score of the complete text.

METEOR (Metric for Evaluation of Translation with Explicit ORdering), proposed by Banerjee et al. (2005) [3], also compared the word segments of the generated text with the reference. In addition to this, stems of a sentence and synonyms of words are also considered for matching. Hence increasing its ability to find the correlation between the generated text and the reference text.

Apart from these two, there are also a few more widely used metrics, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [11], CIDEr (Consensus-based Image Description Evaluation) [15] and SPICE (Semantic Propositional Image Caption Evaluation) [1].

Each metric has its advantages and disadvantages. In the next phase of building and evaluating the model, we will analyze and decide on the metric that is/are most suitable for our dataset and apply it accordingly.

7 RESULTS

7.1 Training Results

We have trained all of our 3 models, Tensorflow, Tensorflow with Attention, and the Pytorch version for 50 epochs, with a batch size of 32. We used Categorical Accuracy as the metric to observe our training accuracy. Figure 7 shows the categorical accuracy trend of the attention model, the accuracy is a little low compared to the Tensorflow version, which is in contrast to what we expected from adding attention to the network. The possible reason for this decrease in the accuracy is, that we added the CBAM module to the Mixed 10 layer of the Inception model and the trainability of all the

Inception layers was set to False. We understood that on setting the trainability of those layers we could expect to see better results. To make an apples to apples to apples comparison we should also make the Mixed 10 layer trainable for both the Tensorflow and Pytorch version. Due to the time limitation of this course, we are not making these changes. We will make these changes and observe the results as part of our future work.

```
252/252 [==============================] - 22s 87ms/step - loss: 2.8297 - categorical_accuracy: 0.3383
252/252 [==============================] - 23s 91ms/step - loss: 2.8144 - categorical_accuracy: 0.3394
252/252 [==============================] - 23s 89ms/step - loss: 2.7986 - categorical_accuracy: 0.3425
252/252 [==============================] - 22s 87ms/step - loss: 2.7875 - categorical_accuracy: 0.3423
252/252 [==============================] - 22s 88ms/step - loss: 2.7738 - categorical_accuracy: 0.3456
252/252 [==============================] - 23s 91ms/step - loss: 2.7608 - categorical_accuracy: 0.3474
252/252 [==============================] - 22s 87ms/step - loss: 2.7481 - categorical_accuracy: 0.3471
252/252 [==============================] - 24s 95ms/step - loss: 2.7347 - categorical_accuracy: 0.3489
252/252 [==============================] - 22s 87ms/step - loss: 2.7286 - categorical_accuracy: 0.3501
252/252 [==============================] - 22s 87ms/step - loss: 2.7153 - categorical_accuracy: 0.3517
252/252 [==============================] - 23s 92ms/step - loss: 2.7069 - categorical_accuracy: 0.3536
252/252 [==============================] - 23s 93ms/step - loss: 2.6979 - categorical_accuracy: 0.3539
252/252 [==============================] - 23s 92ms/step - loss: 2.6911 - categorical_accuracy: 0.3549
252/252 [==============================] - 24s 95ms/step - loss: 2.6809 - categorical_accuracy: 0.3564
252/252 [==============================] - 22s 87ms/step - loss: 2.6748 - categorical_accuracy: 0.3568
252/252 [==============================] - 23s 91ms/step - loss: 2.6668 - categorical_accuracy: 0.3572
252/252 [==============================] - 22s 87ms/step - loss: 2.6617 - categorical_accuracy: 0.3586
252/252 [==============================] - 21s 85ms/step - loss: 2.6537 - categorical_accuracy: 0.3588
252/252 [==============================] - 22s 88ms/step - loss: 2.6500 - categorical_accuracy: 0.3599
252/252 [==============================] - 24s 93ms/step - loss: 2.6392 - categorical_accuracy: 0.3603
252/252 [==============================] - 22s 86ms/step - loss: 2.6296 - categorical_accuracy: 0.3622
252/252 [==============================] - 23s 92ms/step - loss: 2.6216 - categorical_accuracy: 0.3641
252/252 [==============================] - 21s 85ms/step - loss: 2.6131 - categorical_accuracy: 0.3656
252/252 [==============================] - 23s 93ms/step - loss: 2.6045 - categorical_accuracy: 0.3666
252/252 [==============================] - 22s 85ms/step - loss: 2.5979 - categorical_accuracy: 0.3673
252/252 [==============================] - 22s 85ms/step - loss: 2.5906 - categorical_accuracy: 0.3687
252/252 [==============================] - 22s 89ms/step - loss: 2.5842 - categorical_accuracy: 0.3687
```

Figure 7: Training results of Tensorflow version with Attention

(a)

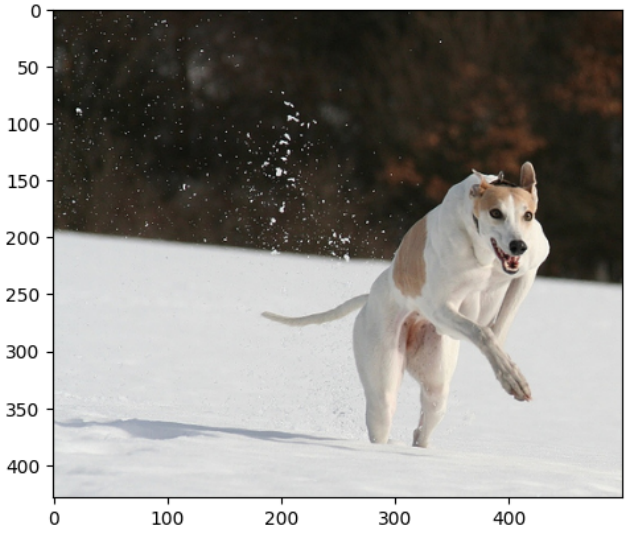
- **Tensorflow:** .
- **Tensorflow with Attention:** four fishing other fun boogie in little small.
- **Pytorch:**

(b)

Figure 8: (a) Test image 1, (b) Corresponding predicted captions of the image

8 OBSERVATION

Based on the results we should make some final observations



(a)

- **Tensorflow:** .
- **Tensorflow with Attention:** two tri-colored in little stump playing shore
- **Pytorch:**

(b)

Figure 9: (a) Test image 2, (b) Corresponding predicted captions of the image

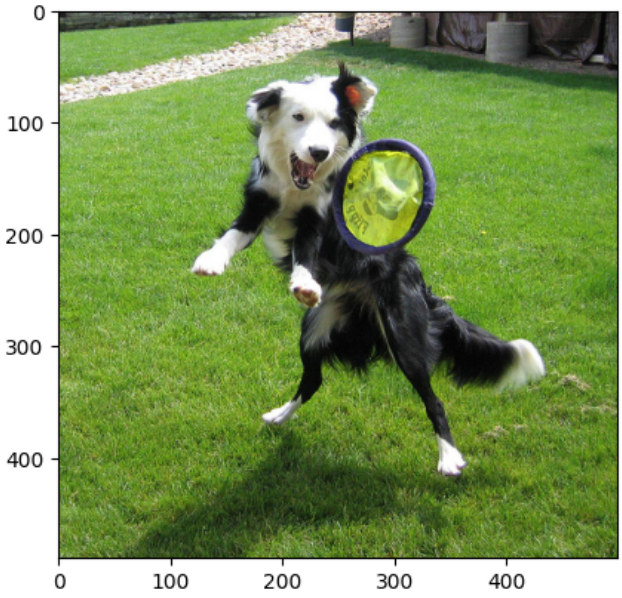
9 LIMITATIONS

Computational Resources: We are using Jupyer Notebooks and Google Colaboratory to train our model. This limitation impacts the model's training time and complexity. To address this, transitioning to more powerful computing environments, such as high-performance GPU clusters or cloud platforms with dedicated accelerators, could significantly improve model training and experimentation.

Timeframe Constraints: The project timeline imposes restrictions on the depth of exploration and the implementation of advanced techniques.

Dataset Size: We are using the Flickr 8k dataset, which could be suitable for initial model development, but could seriously impact the limitation on the model's generalization to diverse real-life scenarios. To enhance the model's capability to handle a broader range of images and scenes, we should train our model on larger and more diverse datasets, such as the Flickr 30k dataset.

Model Robustness: As the model is currently restricted by limited training data and computational resources, future enhancements may involve implementing techniques to improve model robustness. Including the exploration of data augmentation strategies, regularization techniques, and experimenting with more advanced pre-trained models for feature extraction.



(a)

- **Tensorflow:** .
- **Tensorflow with Attention:** two fishing playing shore in little cut playing catches
- **Pytorch:**

(b)

Figure 10: (a) Test image 3, (b) Corresponding predicted captions of the image

Evaluation Metrics: Implementing BLEU evaluation while training is making the training time of one epoch to reach close to 3 hours which is a lot and for our model we are training for about 50 epochs to get notable results. So including BELU evaluation would take about 150 hours of training which we cannot do at this moment. So we implemented categorical accuracy to determine the accuracy of our models.

Incorporating CBAM: The implementation of the Convolutional Block Attention Module (CBAM) represents a notable enhancement. However, due to computational constraints, the CBAM implementation is applied only to a specific layer, Mixed10 of the Inception model. Future exploration could involve integrating CBAM more extensively across the entire model architecture to harness its full potential in refining both channel and spatial attention.

10 CONCLUSION

Goutham

11 FUTURE WORKS

There are several potential improvements we can do to further improve our model. Below is a more detailed overview of potential future works and improvements for the image captioning project.

11.1 Novel Decoder Architectures

- Attention mechanisms: Incorporate spatial and channel-wise attention to enable the decoder to focus on salient image regions for caption generation. Methods like bottom-up top-down attention [2] are the current state-of-the-art.
- Transformers: Replace the RNN decoder with a transformer architecture which allows faster training times through parallelization. Transformer decoders have become prevalent in language tasks [5]. You can experiment with adapting them for sequence prediction in captioning.
- Object detection fusion: Detect objects like people, settings, activities etc in images using off-the-shelf detectors like Faster R-CNN pre-trained on Visual Genome. Inject these as extra decoder inputs to provide localization cues [12].

11.2 Training Approaches

- Reinforcement learning: Directly optimize non-differentiable test metrics like CIDEr, SPICE via REINFORCE algorithm and reward policy gradients instead of cross-entropy loss [14].
- Retrieval augmentation: Retrieve similar training captions to augment corpus for long-tail concepts. Mitigates vocabulary/context gaps [10].
- Interactive learning: Design UI to collect human preferences and corrections to iteratively improve model, customizing it to user needs [4].

11.3 Output Enhancements

- Caption post-processing: Use rule-based systems or learned models to paraphrase, compress, correct syntax etc [16].
- Creative captioning: Condition generative adversarial networks on images to produce more varied, interesting descriptions capturing nuances [6].

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. *European conference on computer vision* (2016), 382–398.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (2005), 65–72.

[4] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robin C Miller, Robert C Miller, Ana Tatarowicz, Brandyn White, Samuel White, et al. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. 333–342.

[5] Marcella Cornia, Matteo Stefanini, Tiberio Caselli, and Roberto Navigli. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10578–10587.

[6] Bo Dai, Zhilin Xie, Yiping Fang, Jian-Fang Li, Graham Neubig, and Jaime Carbonell. 2017. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*. 2970–2979.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations* (2020).

[8] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Doll'ar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2018. Neural image caption generation with visual semantic roles. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 428–436.

[9] Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* 47 (2013), 853–899.

[10] Ben Krause, Aidan N Johnson, Ranjay Krishna, and Li Fei-Fei. 2020. Retrieval augmentation reduces hallucination in conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7464–7471.

[11] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. *Text summarization branches out* (2004), 74–81.

[12] Jiasen Lu, Caiming Xiong, Steven Hoi, and Richard Socher. 2018. Neural baby talk. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7219–7228.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (2002), 311–318.

[14] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7008–7017.

[15] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 4566–4575.

[16] Yue Wang, Qi Huang, and Liwei Wu. 2019. Reconstructing well-written paragraphs from images for bidirectional image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9020–9027.

[17] Qi Wu, Chunhua Shen, and Anton van den Hengel. 2016. Image captioning with semantic attention. *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 4651–4659.

[18] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the International Conference on Machine Learning (ICML)* (2015).