Image Caption Generator

Class: MSML 612
Professor: Yuntao Liu
Group Number: 11
Group member: Xiyuan Liu, Xin Li, Gan Wang

Abstract

This work explores different encoder-decoder architectures for image captioning. Various encoders including ResNet-50, DenseNet201, and others are combined with LSTM decoders with and without attention. Experiments demonstrate that the choice of encoder significantly impacts caption generation performance. The ResNet-50-LSTM-Attention model achieves the best results, generating accurate and semantically aligned captions. Attention mechanisms consistently improve performance by enhancing contextual relevance. The DenseNet201-LSTM model also emerges as a strong contender, balancing linguistic and semantic quality. Overall, the encoder-decoder framework proves effective for image captioning. Key findings suggest the critical role of the encoder choice and the benefits of attention. Our results indicate that combining DenseNet201 and LSTM+attention could further improve performance. This work provides meaningful insights into model architectures and components for advancing image captioning.

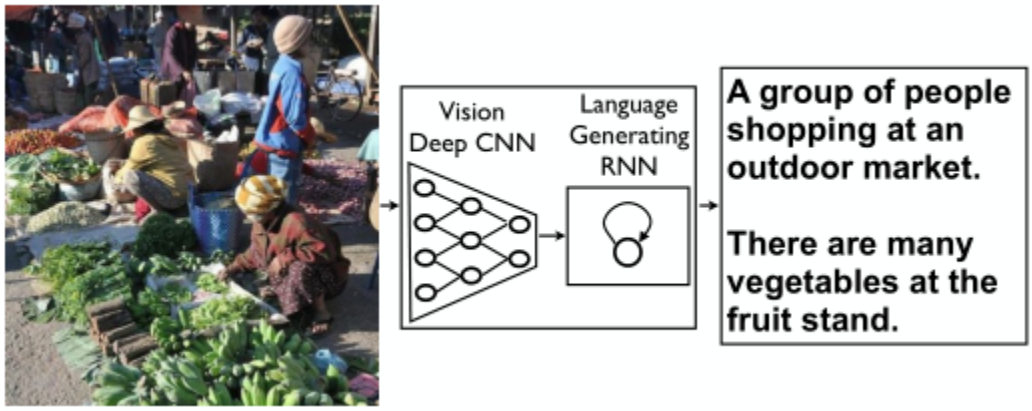
1.Introduction

The task of automatically crafting coherent and linguistically precise descriptions for images represents a formidable challenge, the successful resolution of which holds profound potential, particularly in aiding visually impaired individuals in comprehending web images. This undertaking surpasses conventional image classification or object recognition, areas extensively explored within the computer vision community [1], due to the intricate demand of capturing not only the objects depicted in an image, but also their interrelationships, attributes, and engaged activities. Accomplishing this requires expressing semantically rich knowledge in natural language, necessitating both visual understanding and proficient language modeling.

Our project finds inspiration in recent strides made in machine translation, where the objective is to transform a source language sentence, S, into its target language translation, T, by optimizing p(T|S). While conventional machine translation comprised a sequence of distinct

tasks (individual word translation, alignment, reordering, etc.), contemporary methodologies have demonstrated the efficiency of Recurrent Neural Networks (RNNs) in achieving this objective. However, RNNs are constrained in their ability to handle longer sequences, affecting performance as sequence length increases. To address this limitation, we introduce the long short-term memory model (LSTM), a specialized variant of the RNN architecture. LSTM incorporates three control units ("cells") - input gate, output gate, and forget gate - that regulate the flow of information through the network. This architecture excels in retaining relevant information while disregarding noise, effectively addressing the challenge of long-term sequence dependencies within neural networks.

In the context of our proposed approach, we draw upon the architectural elegance of our LSTM model, replacing the decoder LSTM with a deep convolutional neural network (CNN). Demonstrated advancements in recent years have established CNNs as adept tools for extracting comprehensive image representations, thereby lending themselves naturally as "encoders" for image data. By leveraging a pre-trained CNN, initially fine-tuned for image classification, we harness its final hidden layer as an input to the LSTM decoder. This combined architecture, illustrated in Figure 1, forms our Neural Image Caption (NIC) model. It harmoniously melds the visual prowess of CNNs with the linguistic generative abilities of LSTMs, offering a cohesive framework for generating coherent and contextually relevant natural language descriptions from input images.



T

Figure1. NIC, our model, is based end-to-end on a neural network consisting of a vision g CNN followed by a language generating LSTM. It generates complete sentences in natural language from an input image, as shown in the example above.

2. Prior work

Image captioning stands as a substantial challenge due to its dual demand for generating coherent natural language descriptions and comprehending intricate visual intricacies. This intricate task involves assimilating diverse factors like colors, positions, sizes, and object relationships within each image pixel. Amidst this complexity, the algorithm must adeptly unravel intricate correlations within multivariate data while navigating the persistent structural variations that even arise among identical image categories. These structural disparities emerge from differences in shooting conditions, lighting, angles, and other contributing variables, yielding significant content differences. The very essence of bridging the chasm between visual and textual realms compels machine learning algorithms to adroitly navigate the intricate connections that intertwine these distinct dimensions. This intricate connection goes beyond the confines of a simplistic pixel-to-word mapping, operating instead across a spectrum of complexities. Undoubtedly, the availability of extensive and meticulously curated datasets holds paramount significance in the training and evaluation of the algorithm. However, it remains that these datasets often carry the burden of noise, errors, and gaps, subsequently posing challenges that directly influence the training efficacy and the overarching generalization capacity of deep learning models.

In the exploration of image captioning solutions, an array of approaches have been reviewed. Notably, attention-based methods draw inspiration from human attention patterns and emulate the focus of the human eye on images. These methods incorporate an attention module into the conventional CNN+LSTM framework, enabling the model to prioritize significant elements within the image. Some representative algorithms, including Neural Machine Translation by Jointly Learning to Align and Translate [8], and Long short-term memory have proven the credibility of this approach. Building upon attention-based techniques, the Attention-based+Spatial & Semantic Relations method seeks to enhance the correlation between image and text features. This approach leverages spatial and semantic relationships within images to produce more contextually fitting descriptions. One of the representative algorithms in this category involves Image Captioning: Transforming Objects into Words.[9]. Alternatively, convolutional-based methods constitute the foundational architecture of image captioning, encompassing CNN+LSTM configurations. The advent of SeqToSeq models has paved the way for fully convolutional image captioning methods. Diverse algorithms, such as Convolutional Image Captioning [10] and Convolutional Decoders, exemplify this approach. Also in the wake of Transformer advancements, the Transformer-based method has gained prominence. Scholars recognize the innate attention mechanism of Transformers and its aptness for text-related tasks. Consequently, a plethora of Transformer-based Image Caption algorithms have emerged. Among them are approaches like Attention on Attention for Image Captioning[11]. In our report, our team decided to complete image captioning solutions by reproducing attention-based methods and convolutional methods.

3. Dataset

In this study, we leverage the training, validation, and test datasets of the Flickr 8k dataset to facilitate the training, refinement, and comprehensive evaluation of our proposed methodology, thereby substantiating its efficacy and performance. Originally curated in 2008 by researchers affiliated with the University of Illinois, the Flickr 8k dataset comprises a collection of 8,000 images sourced from the prominent online photo-sharing platform, Flickr. Each image is meticulously paired with a set of five crowd-annotated textual descriptions, resulting in an aggregate compilation of over 40,000 human-authored narratives. This corpus effectively encompasses a wide spectrum of visual scenes, objects, and entities. The textual captions, meticulously curated via the utilization of Amazon Mechanical Turk, employ a diverse lexicon of approximately 8,791 distinct terms to articulate the explicit visual constituents of each image in a natural linguistic manner. Characterized predominantly by dimensions of approximately 500 x 500 pixels, the dataset presents a rich assortment of full-color photographs spanning an assorted array of subject matters. Since its inaugural conception, the Flickr 8k dataset has evolved into a widely acknowledged yardstick for automated image captioning pursuits, consequently serving as an instrumental platform for training and assessing computer vision methodologies that entail the extraction of visual constructs from images coupled with the generation of corresponding textual explications. The concurrent amalgamation of dual modalities within this dataset, in conjunction with the meticulously annotated ground truths, has firmly cemented its role in propelling advancements within the domain of multimodal machine learning algorithms that seamlessly integrate facets of computer vision, natural language processing, and intricate image-text correspondences. We use first 85% of dataset as the training data, and the rest of 15% of dataset as the testing set.

4. LSTM Generative Model(Decoder)

In the present investigation, we intend to employ a combined neural and probabilistic paradigm for the purpose of generating descriptive captions derived from images. This methodological approach involves the utilization of a recurrent neural network (RNN) to decipher features that have been extracted via a convolutional neural network (CNN) encoder. Subsequently, these decoded features are employed in conjunction with word embeddings derived from pre-existing captions to facilitate the generation of textual descriptions[12].

Our primary objective is to enhance the likelihood of generating accurate descriptions corresponding to a given image. This objective is pursued through the maximization of the probability associated with the correct description, given the input image. This optimization endeavor is encapsulated by the subsequent mathematical formulation:

$$\theta^{\star} = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta) \quad (1)$$

The parameters denoted by θ govern the configuration of our model. In this context, we symbolize an image as I, while S corresponds to its accurate textual representation. Given that S encapsulates a generic sentence structure, its length remains unbounded. Consequently, it is customary to apply the chain rule to effectively model the collective probability across the sequence of elements represented as S0, S1, up to SN, where N signifies the length of the specific instance under consideration.

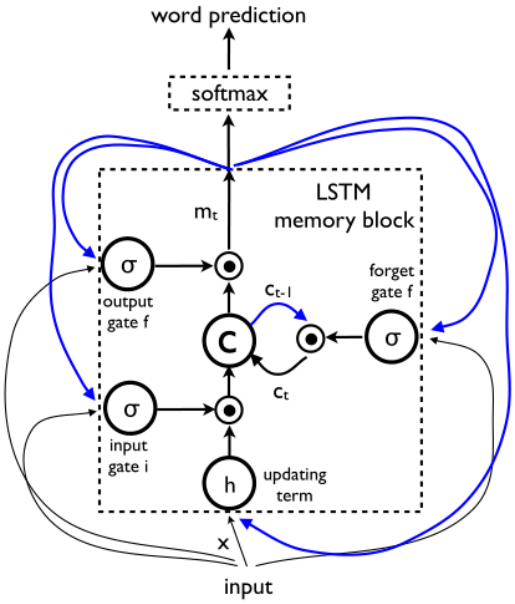
$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1}) \quad (2)$$

For the sake of simplification, we have omitted explicit reference to the θ dependency. During the training phase, the (S, I) pair represents an exemplar utilized for learning. The optimization process is characterized by the maximization of the cumulative logarithmic probabilities, as elucidated in equation (2), across the entirety of the training dataset. This optimization endeavor is operationalized through the utilization of stochastic gradient descent, facilitating the refinement of our model's parameters to align with the training data distribution.

It is inherent to our approach to model the conditional probability, $p(S_t|I, S_0, \ldots, S_{t-1})$, employing a Recurrent Neural Network (RNN). This neural architecture accommodates the variability in the number of words conditioned upon, spanning from S_0 to S_{t-1}, through the utilization of a fixed-length hidden state or memory, denoted as ht. This memory construct undergoes iterative updates subsequent to the introduction of a new input, xt, facilitated by a nonlinear function encapsulated by equation (3):

$$ht + 1 = f(h_t, x_t) \quad (3)$$

To impart greater clarity to the aforementioned RNN framework, two pivotal design choices demand consideration: the specification of the precise form of the function f, and the manner in which inputs, encompassing both images and words, are presented as xt. To address the former, we opt for the incorporation of a Long-Short Term Memory (LSTM) network, a proven performer in the domain of sequence-oriented tasks such as translation, boasting state-of-the-art outcomes. This strategic selection aligns with the pursuit of optimized performance within our context.



4.1 Training

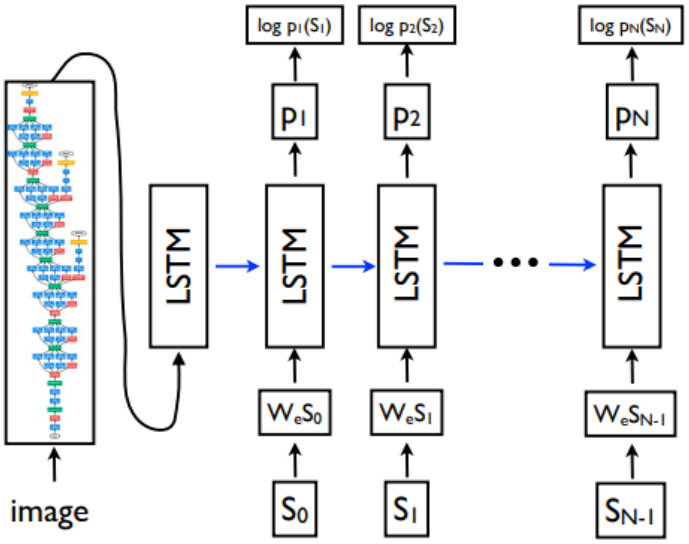
The training of the LSTM model involves the task of word prediction for each element of the sentence, subsequent to the LSTM's exposure to both the image and all preceding words, as delineated within the contextual framework of $p(S_t|I, S_0, \ldots, S_{t-1})$. To foster a more insightful comprehension, it proves instructive to conceptualize the LSTM architecture in an "unrolled" manifestation. This conceptualization entails the creation of an individual instance of the LSTM memory for both the image and each constituent word of the sentence. Importantly, all instantiated LSTMs share identical parameters, and the output, m_{t-1}, emanating from the LSTM at time t − 1, serves as an input for the LSTM at time t. Consequently, the conventional recurrent connections are transformed into feed-forward connections within this unrolled portrayal.

To provide a more detailed perspective, considering I as the input image and $S = (S_0, \ldots, S_N)$ as the accurate descriptive sentence corresponding to this image, the procedural sequence of unrolling can be delineated as follows:

$$x_{-1} = CNN(I) \quad (4)$$

$$x_t = W_e S_t, \; t \in \{0 \ldots N-1\} \quad (5)$$

$$p_{t+1} = LSTM(x_t), \; t \in \{0 \ldots N-1\} \quad (6)$$



Within this schema, individual words are represented by one-hot vectors denoted as S_t, characterized by dimensions equivalent to the size of the designated dictionary. It's noteworthy that S_0 signifies a distinct start word, while S_N indicates a designated stop word, both serving to mark the commencement and conclusion of the sentence, respectively. The emission of the stop word serves as a signal from the LSTM that a complete sentence has been generated.

Furthermore, it is paramount to recognize that both the image and the words are projected onto a common spatial domain. This alignment is achieved by employing a visual Convolutional Neural Network (CNN) for the image and word embeddings We for the textual components. Importantly, the image I is introduced solely once, at t = −1, serving to provide initial context regarding the image content.

5.Model architecture

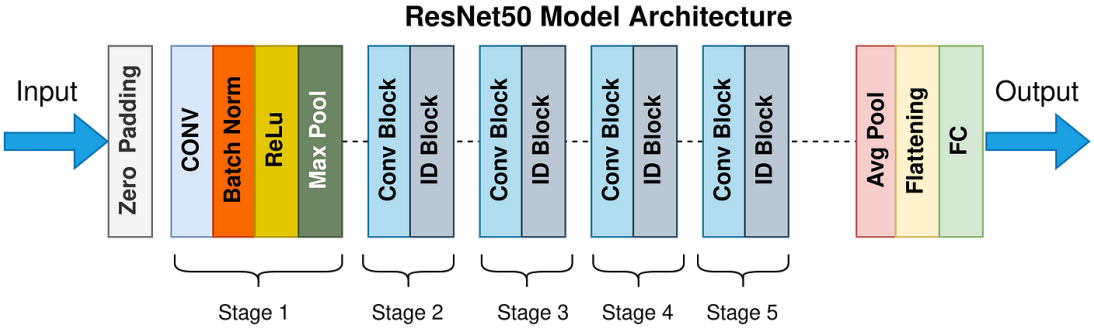
In this section, we present two distinct technical approaches for image captioning, with the first approach employing a deep Convolutional Neural Network (CNN) as the encoder. Recent years have witnessed compelling evidence that CNNs possess the capability to yield a comprehensive representation of input images by condensing them into fixed-length vectors. These vectors, characterized by their richness, have demonstrated utility across a spectrum of visual tasks. Subsequently, we leverage the potency of Long Short-Term Memory (LSTM) networks as decoders to transform these encoded image representations into coherent textual descriptions. We will discuss the combination of ResNet50, VGG16, and DenseNet201 with LSTM in detail in the 5.1 section.

It's worth noting that an alternative approach, elaborated upon in section 5.2, capitalizes on a CNN as the encoder. This approach, however, introduces an additional layer of sophistication by integrating the Bahdanau attention mechanism with the LSTM decoder. This

augmentation enhances the model's capacity to generate descriptive textual sequences by selectively focusing on pertinent aspects of the encoded visual features.

5.1.1 ResNet50 and LSTM

First, we use the Resnet 50 and LSTM combination as a baseline. The image captioning model uses a modified ResNet50 convolutional neural network as the encoder to extract visual features from the input image. The last two fully connected layers of ResNet50 are removed, leaving only the avg_pool output layer that generates a 2048-dimensional spatial feature vector for each image. This compressed image representation encodes hierarchical visual concepts learned by the pre-trained ResNet50 layers. The encoder passes just the avg_pool average image features to an LSTM recurrent neural network decoder to generate textual captions.

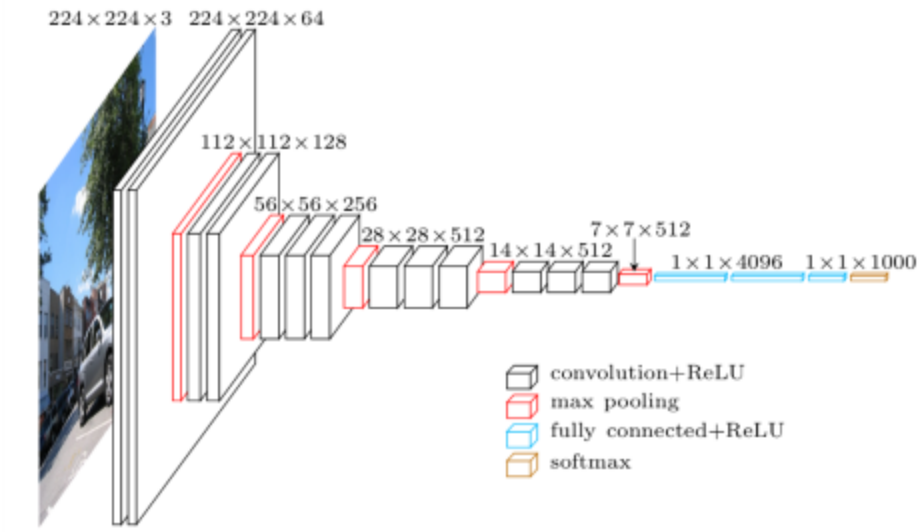


Before the LSTM decoder takes the feature vector extracted from ResNet50, the encoded 2048-dimensional image features need to go through a dense layer and reshape to 256-dimension to match the word embedding dimension. Then the image features would be fed into LSTM as initial hidden state and the sentence as input at the timestamp and outputs a word prediction, maintaining an internal state to model temporal context. The decoder is trained using teacher forcing to learn the mapping between encoded image features and target captions. The modified ResNet50 CNN encoder and LSTM RNN decoder are jointly trained end-to-end on image-caption pairs to maximize the likelihood of generating the correct textual description for a given image. By removing the fully connected layers, the model learns to map directly from encoded visual features to language using the RNN, with ResNet50 feature extraction frozen.

5.1.2 VGG16 and LSTM

The architecture utilizes a modified VGG16 convolutional neural network as the encoder to extract image features. VGG16 is pre-trained on ImageNet but the last 3 layers are removed. The modified VGG16 contains 13 layers of convolutions and max pooling to extract hierarchical feature representations of the input image. A flatten layer is added to flatten the features into a 1D vector. This is followed by a dense layer to reduce the dimensions from 25088 to 4096.

Similar to the previous combination, This 4096-dimensional encoded image feature vector also needs to go through a dense layer and reshape to a 256-dimensional image feature



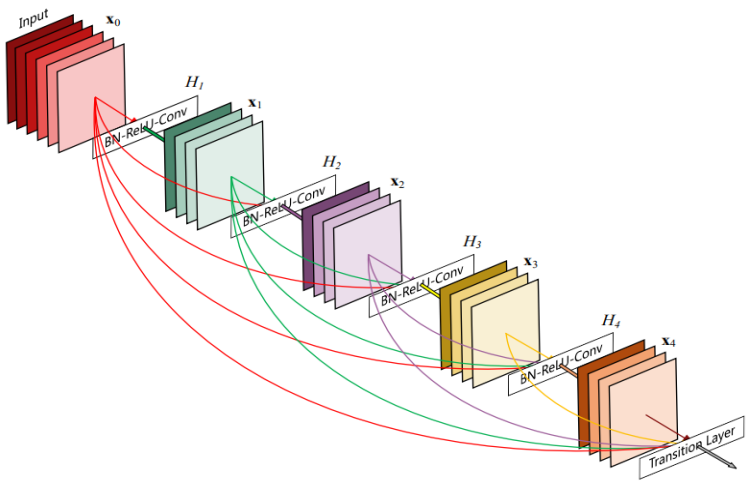
vector. Then fed into the decoder LSTM recurrent neural network to generate the image caption. The LSTM also takes as input the previous words of the caption encoded as 256-dimensional word embeddings. The embeddings and image features are concatenated at each timestep as input to the LSTM. As the LSTM outputs words, the generated words are recursively fed back in to predict the next word in the caption sequence.

By using a truncated VGG16 encoder and LSTM decoder, this architecture can extract semantic image features and model the language caption while being computationally cheaper than using the full VGG16. The modified CNN and RNN components enable generating relevant captions conditioned on visual context.

5.1.3 DenseNet201 and LSTM

This image captioning model uses a DenseNet201 encoder to extract image features. DenseNet201 is a convolutional neural network that consists of dense blocks with shortcut connections between layers to improve information flow. The encoder outputs a 1920-dimensional feature vector for each image. This feature vector is passed through a fully-connected layer to reduce the dimensions to 256.

For the decoder, the model uses an LSTM recurrent neural network. The 256-dimensional image feature vector is reshaped to (1,256) and provided as the first input to the LSTM. The LSTM also receives an embedding vector for each word in the caption so far, resulting in an input shape of (caption_length, 256). The LSTM outputs a 256-dimensional vector encoding the context of the full caption. This vector is passed through linear layers and a softmax to predict the next word.



The encoder and decoder are trained end-to-end. The DenseNet201 encoder extracts semantic features from the image, while the LSTM decoder models the captions as sequences conditioned on the image features. By optimizing both parts together, the model learns to generate relevant captions based on the image contents. The overall architecture combines the strengths of CNN image features and RNN sequence modeling for image captioning.

5.1.4 LSTM Architecture

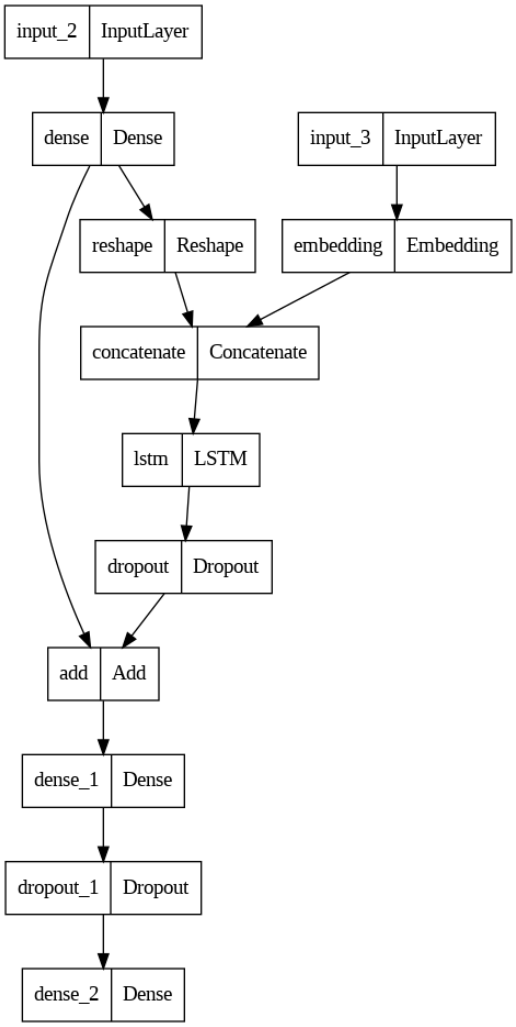
The LSTM decoder is designed to fuse image features and textual context to generate coherent captions. The model consists of two main inputs: input2, representing the extracted image features and input3, signifying the textual context with a variable maximum length. The image features are initially processed through a dense layer with 256 neurons and a ReLU activation function, resulting in a condensed representation. This representation is then reshaped into a format of (1, 256) to align with subsequent concatenation operations.

On the textual side, the sentence_features are generated through an embedding layer with a vocabulary size of vocab_size and 256 dimensions, aiding in capturing the semantic meaning of words. The reshaped image features and the embedded sentence features are combined using concatenation along the time axis, facilitating a coherent fusion of visual and textual information. This merged representation is passed through an LSTM layer with 256 units, enabling the model to capture intricate temporal relationships and generate captions that reflect both image content and contextual language.

To enhance model robustness and prevent overfitting, dropout layers are introduced. A dropout rate of 0.5 is applied after the LSTM output, followed by an element-wise addition of the initial image features to the LSTM output. The resulting representation is then further processed through a dense layer with 128 neurons and a ReLU activation function, adding a layer of

semantic refinement. Another dropout layer with a rate of 0.5 is incorporated to mitigate overfitting concerns before the final prediction step.

The ultimate output is generated through a dense layer with softmax activation, aiming to predict the next word in the caption vocabulary. The entire architecture is encapsulated within a Keras Model object named caption_model, which takes input2 and input3 as inputs and produces the output. The model is compiled using categorical cross-entropy loss and the Adam optimizer, thus making it ready for training and generating descriptive captions for images. This comprehensive architecture effectively harnesses the synergy between visual and textual information to create meaningful and contextually relevant captions.



LSTM Architecture

5.2 ResNet50 and LSTM with Attention

The model employs a convolutional neural network (CNN) encoder and recurrent neural network (RNN) decoder with Bahdanau attention. The encoder is a pretrained ResNet50 that extracts feature representations from input images. The decoder is a long short-term memory (LSTM) network that generates textual captions word-by-word.

The Bahdanau attention, also known as additive attention, computes relevance scores between the decoder hidden state and encoder outputs through a multi-layer perceptron (MLP). Specifically, the decoder hidden state and each encoder output are projected into an attention space using learned weight matrices W and U respectively. The projected decoder state and encoder outputs are then summed and passed through a tanh activation to obtain combined attention vectors. These combined vectors are scored by another learned weight matrix A to obtain unnormalized relevance scores. A softmax function is applied to normalize the scores into a probability distribution over the encoder outputs. The attention weights are multiplied with the encoder outputs to take a weighted sum, concentrating more weight on relevant parts of the image features. This weighted sum forms the context vector that the decoder receives as additional input at each timestep along with the word embedding.

At each timestep, the decoder LSTM receives two inputs - the word embedding of the previous word and a dynamic context vector computed via Bahdanau attention. Specifically, the decoder hidden state is projected into an attention space and compared with projected encoder outputs through a multi-layer perceptron to obtain unnormalized relevance scores. These scores are normalized using a softmax function to derive attention weights over the encoder outputs.

The attention weights concentrate higher values on encoder features relevant to the current decoding step. A weighted sum of the encoder outputs is computed using these attention weights to form the context vector containing pertinent visual information. By learning to choose which parts of the image to focus on through the attention mechanism, the decoder can produce more accurate context-aware predictions.

The initial hidden and cell states of the decoder LSTM are learned projections of the mean-pooled encoder features. At each timestep, the LSTM output is projected into the vocabulary space to predict a distribution over the next word. Teacher forcing is employed during training where ground truth words are fed as inputs. During inference, the model recursively predicts the next word to auto-regressively generate the full caption.

In summary, the CNN encoder extracts semantic visual features, while the LSTM decoder models long-range dependencies in language and uses the attention mechanism to selectively focus on relevant image regions while generating captions word-by-word.

6.Loss Function

In our experiments, we utilize the Categorical Cross-Entropy (CCE) loss function, commonly applied in multi-class classification scenarios. CCE assesses the disparity between predicted and actual class distributions, guiding neural network training by generating larger loss values in the presence of discrepancies. Since encoding words resembles multi-category classification, CCE is well-suited. Computation involves contrasting predicted probabilities with a one-hot encoded true label, quantifying information distribution variance. Minimizing CCE aims to align model predictions closely with true labels, thereby enhancing classification accuracy. His formula is as follows:

$$L_{CCE} = -\sum_{i=1}^{output} y_i \bullet \log \hat{y}_i$$

7. Evaluations

Regrettably, the realm of image captioning lacks tailored evaluation metrics to precisely gauge its performance. Consequently, a pragmatic approach is essential. To address this gap, we opt to employ the BLEU (Bilingual Evaluation Understudy) metric, which, although initially designed for machine translation, has found relevance in assessing image caption quality. BLEU quantifies the overlap between generated captions and reference captions, providing an approximate measure of the caption's fluency and adequacy. Additionally, we harness semantic evaluation through the utilization of the Sentence Transformer model. This step delves into capturing semantic similarities between generated and reference captions, offering a more nuanced perspective on the generated text's quality and relevance. By leveraging these two complementary evaluation methodologies, we endeavor to paint a more comprehensive picture of the effectiveness and accuracy of our image captioning solution . While the absence of specialized metrics presents a challenge, our chosen evaluation strategy aims to bridge this gap and provide valuable insights into the performance and potential enhancements of our model.

In our pursuit of a comprehensive evaluation strategy for image captioning, we have found that the BLEU (Bilingual Evaluation Understudy) metric, while widely used, presents limitations in capturing nuanced language nuances. Consequently, we have opted to focus solely on the 1-gram BLEU score, as other n-gram scores have yielded small values that are not truly representative of the caption quality. This deliberate choice enables us to extract a more meaningful and interpretable assessment of the generated captions' linguistic coherence and resemblance to the reference captions.

To delve deeper into the realm of semantic evaluation, we have employed the distilbert-base-nli-mean-tokens model, a Sentence Transformer that generates vectors

encapsulating the semantic essence of both original and generated captions. By harnessing this approach, we are able to quantify the semantic similarities between captions, transcending syntactical correctness to encompass the intrinsic meaning and relevance of the text. The calculation involves determining the cosine similarity between the vectors, providing a nuanced insight into how well the generated captions capture the intended semantic essence of the reference captions.

Throughout this evaluation process, we meticulously compute these scores for individual instances. However, for a more holistic representation of the model's performance, we aggregate these scores into average values. These averages stand as proxies for the model's overall efficacy, encapsulating its ability to generate captions that align linguistically and semantically with the reference captions.

Encoder	Decoder	Avg 1-gram Bleu score	Avg Sentence Semantic Similarity Score (distilbert-base-nli-mean-tokens)
Resnet 50	LSTM	0.3838	0.3531
VGG16	LSTM	0.4436	0.4364
Densenet201	LSTM	0.5041	0.5302
Resnet 50	LSTM + Attention	0.4266	0.587

Model Performance Metrics

Across the encoder-decoder combinations, the model featuring a "Densenet201" encoder and "LSTM" decoder stands out with the highest scores in both the 1-gram Bleu score (0.5041) and Sentence Semantic Similarity Score (0.5302). This suggests that the Densenet201-LSTM combination effectively captures linguistic nuances and semantic meanings in the captions it generates.Comparatively, models with "VGG16" and "Resnet 50" encoders combined with "LSTM" decoders also demonstrate competitive performance in terms of both linguistic similarity and semantic consistency, with Bleu scores of 0.4436 and 0.4266 respectively, and corresponding Semantic Similarity Scores of 0.4364 and 0.587.

The model combining "Resnet 50" with an "LSTM + Attention" decoder achieves a notably high Semantic Similarity Score of 0.587, indicating its strong ability to generate captions that align semantically with reference captions. However, its 1-gram Bleu score of 0.4266 suggests that there might be room for improvement in terms of linguistic similarity.

In summary, the table highlights how different encoder-decoder combinations influence the quality of generated captions, considering both linguistic fidelity and semantic relevance. The Densenet201-LSTM model appears as a leading performer, excelling in both linguistic and semantic aspects. while the Resnet-50-LSTM-Attention model demonstrates competitive performance on semantic relevance.

In addition to evaluating the average Bleu score and average sentence semantic similarity score, we also took a look at the best and worst case of sentence semantic similarity in each model. In the realm of image captioning, the pursuit of evaluating models against best and worst case scenarios holds paramount significance. Assessing performance extremes, encompassing both optimal and challenging conditions, allows us to glean comprehensive insights into the model's strengths and limitations. The best-case scenarios provide a benchmark for the model's upper potential, showcasing its ability to generate accurate, fluent, and semantically aligned captions that closely resemble human-generated descriptions. On the other hand, investigating worst-case scenarios exposes the model's vulnerabilities and areas requiring improvement. Such situations involve complex or ambiguous images, poor lighting, intricate object relationships, and inherent structural variations.

Encoder	Decoder	Best Case Image ID	Avg Best Case Sentence Semantic Similarity Score	Worst Case Image ID	Avg Worst Case Sentence Semantic Similarity Score
Resnet 50	LSTM	526661994_21838fc72c.jpg	0.977	94232465_a135df2711.jpg	-0.0884
VGG16	LSTM	526661994_21838fc72c.jpg	0.984	94232465_a135df2711.jpg	0.0054
Densenet201	LSTM	526661994_21838fc72c.jpg	0.9821	3713882697_6dd30c7505.jpg	-0.0379
Resnet 50	LSTM + Attention	526661994_21838fc72c.jpg	0.973	3729525173_7f984ed776.jpg	-0.0005

Best, worst case score

According Images

In the best-case scenarios, where images are well-lit, clearly composed, and exhibit discernible objects, all the models received the best average sentence semantic similarity score on the same image. VGG16-LSTM model achieved the highest average Sentence Semantic Similarity Score of 0.984. The Resnet 50-LSTM and Densenet201-LSTM models also demonstrate strong performance in best-case scenarios, with average scores of 0.977 and 0.9821, respectively. These results highlight the models' abilities to generate semantically aligned captions in favorable image conditions.

However, the worst-case scenarios, involving challenging images with complexities like poor lighting or intricate object relationships, reveal varying levels of model struggles. Notably, the Resnet 50-LSTM model encounters difficulty in worst-case scenarios, as indicated by its lowest average Worst Case Sentence Semantic Similarity Score of -0.0884. In contrast, the VGG16-LSTM model exhibits relatively stable performance with an average score of 0.0054, even in worst-case conditions. The Densenet201-LSTM model showcases a balanced response to worst-case scenarios, achieving an average score of -0.0379. Interestingly, the addition of the attention mechanism in Resnet 50-LSTM + Attention appears to enhance its performance in worst-case scenarios, resulting in a relatively favorable average score of -0.0005.

8.Conclusion

In conclusion, our project delved into the intricate landscape of image captioning, shedding light on several key findings that have significant implications for this field. Through a comprehensive exploration of different encoder architectures, we illuminated the pivotal role they play in the image captioning process. Our results demonstrated substantial disparities in performance arising from diverse architecture choices, emphasizing the fundamental importance of encoder selection in this context. Notably, the Resnet-50-LSTM-Attention model emerged as the standout performer, showcasing superior capabilities in generating semantically aligned captions while the Densenet201-LSTM model demonstrates competitive performance on both linguistic accuracy and semantic coherence.

The incorporation of attention mechanisms emerged as a potent tool for enhancing model performance. Our analysis revealed that attention not only boosted the average sentence semantic score but also mitigated the impact of challenging scenarios, as evidenced by the increase in the lowest sentence semantic score compared to the same model without attention. This underscores the potential of attention mechanisms in elevating the quality and contextual relevance of generated captions. Our team holds the belief that the fusion of Densenet201 as the encoder and LSTM+attention as the decoder has the potential to yield enhanced performance compared to all the models we have developed thus far.

A noteworthy observation from our study was the consistency in model performance across identical images, indicating a clear alignment of strengths and competencies. Conversely, when faced with varying images, models exhibited fluctuations in performance, highlighting the complex and context-dependent nature of image captioning.

Our project substantiates the efficacy of the encoder-decoder approach as an efficient and promising avenue for image captioning. Leveraging diverse encoder architectures in tandem with decoder mechanisms, we showcased the viability of this approach in generating captions that strike a balance between linguistic accuracy and semantic coherence.

9. Contribution Factor

Name	contribution factor
Xiyuan Liu	0.95
Xin Li	0.9
Gan Wang	0.85
Total C	2.7

Reference

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014.

[2] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In arXiv:1502.03167, 2015.

[3] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8), 1997.

[4] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[5] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[6] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[7] Graves A, Graves A. Long short-term memory[J]. Supervised sequence labelling with recurrent neural networks, 2012: 37-45.

[8] Bahdanau, Dzmitry, et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv.Org, 19 May 2016, arxiv.org/abs/1409.0473.

[9]Herdade, Simao, et al. “Image Captioning: Transforming Objects into Words.” *arXiv.Org*, 11 Jan. 2020, arxiv.org/abs/1906.05963v2.

[10]Aneja, Jyoti, et al. “Convolutional Image Captioning.” *arXiv.Org*, 24 Nov. 2017, arxiv.org/abs/1711.09151.

[11]Huang, Lun, et al. “Attention on Attention for Image Captioning.” *arXiv.Org*, 21 Aug. 2019, arxiv.org/abs/1908.06954.

[12]Oriol, V., Google, Google, A., Google, S., & Google, D. (n.d.). *Show and Tell: A Neural Image Caption Generator*. https://arxiv.org/pdf/1411.4555.pdf