Tec Empresarial
P-ISSN: 1659-2395; E-ISSN: 1659-3359

DEEP LEARNING-BASED IMAGE CAPTION GENERATION USING VGG16 AND LSTM

T Sree Lakshmi[1] , Valaja Chaithanya[2], N Balavenkata Muni[3]

[1]Associate Professor in CSE, Sri Venkateswara college of Engineering,Tirupati.
[2]PG Scholar in CSE, Sri Venkateswara college of Engineering,Tirupati.
[3]Professor in EEE, Siddartha Institute of Science and Technology ,Puttur.

ABSTRACT

Deep learning algorithms have transformed computer vision jobs in recent years, allowing robots to analyze and grasp visual data with astounding accuracy. Image caption generation is one such task that has attracted a lot of attention; it entails automatically providing a natural language description of an image's content. This research investigates the use of Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) to create an image caption generator.

The system learns to produce informative captions for photos by utilizing the power of deep learning, notably the LSTM for sequence generation and the VGG16 model for feature extraction. The work utilizes the Flickr8k dataset for training and evaluation, demonstrating the effectiveness of the proposed approach through model training, caption generation, and evaluation using BLEU score. Through this work, a deeper understanding of the synergy between CNNs and LSTMs in the context of image understanding and natural language processing is achieved, showcasing the potential for creating intelligent systems capable of understanding and describing visual content.

Keywords: LSTM, CNN, Deep Learning.

1. INTRODUCTION:

The rapid progress in Computer vision has undergone a paradigm shift as a result of deep learning. Empowering machines to comprehend and interpret visual information with unprecedented accuracy and sophistication. Image caption generation is one of the many interesting challenges at the nexus of computer vision and natural language processing (NLP) that has evolved. This assignment entails automatically producing natural language text descriptions that briefly summarize the contents of an image. By imbuing machines with the capability to understand and describe visual content, image caption generation holds immense potential for applications ranging from assistive technologies for the visually impaired to enhancing content accessibility and retrieval in multimedia databases[1].

Traditionally, the idea of teaching machines to describe images in natural language seemed like a distant dream, with researchers grappling with the inherent complexity of both visual perception and language understanding. However, recent advancements in deep learning, fueled by the availability of large-scale annotated datasets and computational resources, have propelled the development of sophisticated models capable of tackling this challenge. In particular, the integration of Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks, for sequential data processing has emerged as a powerful approach for image captioning tasks.

This work delves into the implementation and exploration of a deep learning-based image caption generator using CNNs and LSTMs. The objective is to leverage the complementary strengths of CNNs in extracting hierarchical visual features and LSTMs in modeling sequential dependencies to generate coherent and contextually relevant captions for input images. The architecture comprises a CNN pre-trained on the ImageNet dataset, such as VGG16, for extracting rich image features, which are subsequently fed into an LSTM-based language model to generate captions. The training process involves learning to map images to corresponding descriptive captions, thereby enabling the model to generalize and generate captions for unseen images [2].

To facilitate experimentation and evaluation, the work utilizes the Flickr8k dataset, a widely used benchmark dataset containing images paired with multiple human-annotated captions. By training on this dataset and evaluating the model's performance using metrics such as BLEU score, the effectiveness of the proposed approach can be assessed. Furthermore, the work aims to provide insights into the underlying mechanisms of image understanding and language generation in deep learning systems, shedding light on the interplay between visual perception and linguistic expression.

In summary, this work embarks on a journey to harness the power of deep learning for image caption generation, showcasing the synergy between CNNs and LSTMs in bridging the semantic gap between visual content and natural language descriptions. Through experimentation, evaluation, and analysis, it seeks to contribute to the advancement of intelligent systems capable of comprehending and communicating about the visual world [3].

2. RELATED WORK:

One of the main challenges in artificial intelligence is automatically summarizing the content of images, which combines computer vision and natural language processing. In the past, methods such as Sermanet et al. (2013) and Russakovsky et al. (2015) concentrated on extracting annotations from images, such as nouns and adjectives, and then building sentences based on these annotations (Gupta and Mannem). Donahue and colleagues (Donahue et al.) developed a recurrent convolutional architecture specifically designed for large-scale visual learning, showcasing its effectiveness in tasks like as image description, video description, and video recognition. These models were able to learn end-to-end, but they had difficulty comprehending intermediate findings.

Later on, textual creation from videos was included in the LRCN approach (Venugopalan et al.). On the other hand, the Neural picture Caption (NIC) model was presented by Vinyals et al.

(Vinyals et al.), who specifically designed it for picture captioning. NIC combines a single layer of LSTM that has been trained to maximize the possibility of producing target description sentences from training photos with GoogLeNet. As determined by human assessors, NIC's performance, which was validated both qualitatively and numerically, won first place in the MS COCO Captioning Challenge (2015).

Three key differences between LRCN and NIC can be used to explain performance differences. To begin with, NIC uses GoogLeNet, whereas LRCN uses VGGNet. Second, although LRCN provides visual features to each LSTM unit, NIC feeds visual features just into the first LSTM unit. Finally, NIC has a single-layer LSTM RNN architecture, which is less complex than LRCN's two factored LSTM layers. Although the mathematical underpinnings of picture captioning are the same for all models, different implementation decisions lead to different performance outcomes. LRCN struggles with being both general and simple because it was designed for three different purposes.

Fang and colleagues (Fang and colleagues) suggested a visual concept-based method as an alternative to end-to-end learning. They used multiple-instance learning to train visual detectors for frequently occurring caption terms at first. The next steps entailed gathering statistics on word usage by training a language model with a large dataset that included over 400,000 image descriptions. Lastly, a deep multi-modal similarity model and sentence-level characteristics were used to rank the caption candidates. Remarkably, 34% of the time, their captions matched or outperformed those produced by humans. Nevertheless, reproducibility is reduced by this method's increased reliance on human control over parameters. This method is thought to be the foundation of the Microsoft online application caption bot [4][5].

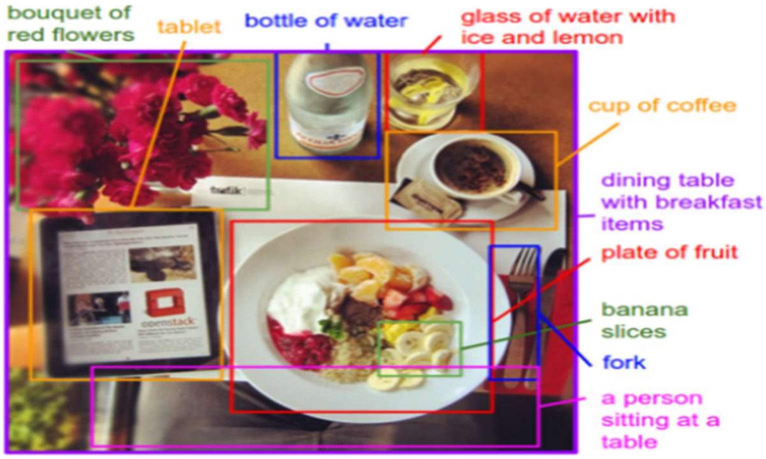

bouquet of red flowers
tablet
bottle of water
glass of water with ice and lemon
cup of coffee
dining table with breakfast items
plate of fruit
banana slices
fork
a person sitting at a table

Figure 1: Regions of images can be described using the visual-semantic alignment technique. Figure sourced from FeiFei and Karpathy

The Visual-Semantic Alignment (VSA) technique was developed by Karpathy and Fei-Fei (Karpathy and Fei-Fei), who produced descriptions for discrete image regions expressed as words or phrases (see Figure 1). It is ensured that retrieved visual features closely correlate with specified image regions by substituting region-based Convolutional Networks (RCNN) for the conventional Convolutional Neural Networks (CNN). Results from experiments demonstrate that descriptions produced by VSA perform better than retrieval-based benchmarks for both full images and fresh datasets annotated at the region level. When compared to whole-image techniques like LRCN and NIC, VSA stands out for its ability to produce descriptions that are both more accurate and diversified.

However, VSA is made up of two different models. The creation of picture-based question-answering systems (as investigated by Zhu et al. in 2016) and dense captioning (as investigated by Johnson et al. in 2016) are examples of further extensions [6].

3. PROBLEM STATEMENT:

The problem addressed in the literature review revolves around the automatic description of image content, a crucial intersection between computer vision and natural language processing. While various approaches have been proposed, including those based on recurrent convolutional architectures and visual concept-based methods, each method has its strengths and limitations. The overarching problem is to develop an effective and efficient model for generating accurate and diverse descriptions of image content. This entails addressing challenges such as handling long-term dependencies, balancing simplicity, and generality, ensuring reproducibility, and improving the alignment of visual features with specific image regions.

4. METHODOLOGY:

This methodology outlines the process of developing an image captioning model using deep learning techniques. It begins with the acquisition and preprocessing of the dataset, followed by feature extraction using a Convolutional Neural Network (CNN). Tokenization and padding are then applied to the captions to prepare them for model input. The model architecture comprises an encoder-decoder framework, with the encoder processing image features and the decoder generating captions based on these features. The model is trained using a data generator, optimizing with the Adam optimizer and categorical cross-entropy loss. Figure 2 represents model diagram for image caption generation [9].

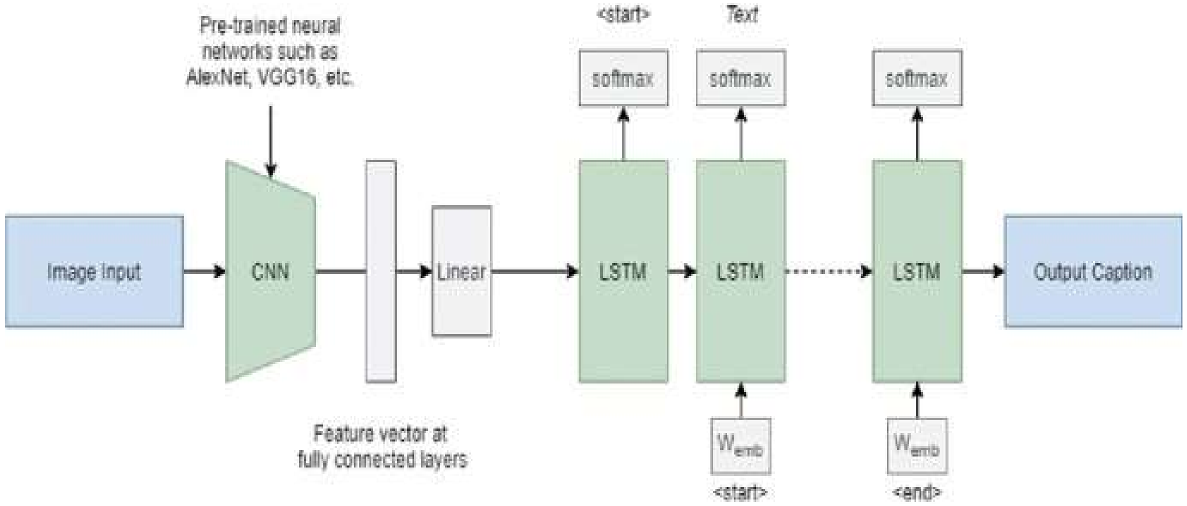


Figure 2: Model Diagram

4.1 Dataset Acquisition and Preprocessing:

The Flickr8k dataset serves as the foundational resource for this work, offering a comprehensive collection of images paired with descriptive captions. The initial phase involves obtaining this dataset, which is renowned in the field of image captioning for its diversity and richness.The Flickr8k dataset, a cornerstone in the realm of image captioning research, comprises a diverse collection of images sourced from the popular photo-sharing platform Flickr. Curated specifically for image captioning tasks, this dataset consists of over 8,000 images, each paired with multiple human-annotated captions. The images encompass a wide range of scenes, objects, and activities, reflecting the natural variability encountered in real-world visual data. The captions associated with each image provide rich descriptions, offering valuable insights into the semantic content and contextual nuances depicted in the images. As a benchmark dataset, Flickr8k has been instrumental in advancing the state-of-the-art in image captioning algorithms by facilitating rigorous evaluation and comparison of different approaches. Its widespread adoption within the research community underscores its significance as a standardized resource for training and testing image captioning models, driving progress towards more accurate and semantically meaningful image understanding systems [10].Preprocessing the captions is a crucial step to ensure their compatibility with the model architecture. This involves several key tasks:

- Cleaning: Unnecessary characters, punctuation, and special symbols are removed from the captions.
- Normalization: All characters are converted to lowercase to standardize the text and avoid redundancy in vocabulary.
- Tokenization: The captions are tokenized, breaking them down into individual words or tokens, which facilitate further processing.
- Special Tokens: Special start and end tokens are added to each caption to denote the beginning and end of the sentence. These tokens help the model understand the structure of the captions during training and generation phases.

By performing these preprocessing steps, the captions are refined into a format that is conducive to subsequent model training and evaluation. This ensures consistency and coherence in the textual data, laying a solid foundation for the development of the image captioning model.

4.2 Feature Extraction:

In the realm of computer vision, Convolutional Neural Networks (CNNs) have revolutionized the way in which visual information is processed and understood. One prominent CNN architecture used for feature extraction in this work is VGG16 as shown in Figure 3. VGG16, short for Visual Geometry Group 16, is a deep neural network architecture developed by the Visual Geometry Group at the University of Oxford. It is characterized by its depth, consisting of 16 layers (hence the name), and its uniform architecture, with small 3x3 convolutional filters throughout the network [4].

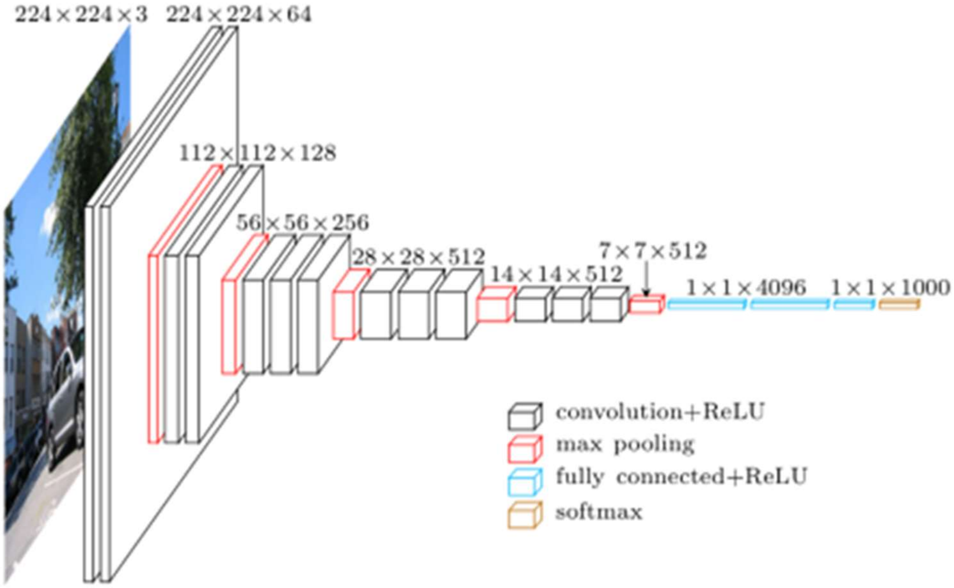


Figure 3: VGG 16 Models

Feature extraction using VGG16 involves leveraging the network's ability to learn hierarchical representations of visual features from input images. The process begins by loading the pre-trained VGG16 model, which has been trained on a large-scale image classification task such as the ImageNet dataset. The model is then utilized to process each image in the dataset, transforming it into a set of high-level feature vectors.

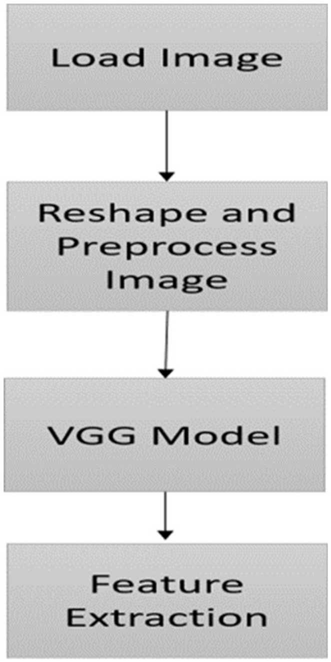


Figure 4: Feature Extraction in images using VGG

During the feature extraction process as shown in Figure 4, each image is passed through the VGG16 network, and its activations from one of the intermediate layers are extracted. These activations represent the learned features of the image at different levels of abstraction. Specifically, the output from the penultimate fully connected layer (often denoted as the "fc2" layer) is commonly used as the feature representation for the image[15].

The extracted features serve as a condensed representation of the visual content present in the images. By encoding the images into a fixed-length feature vector format, the complex visual information is distilled into a format that is more amenable to processing by subsequent stages of the captioning model. This feature representation encapsulates various visual characteristics of the images, including shapes, textures, and patterns, enabling the subsequent captioning model to generate contextually relevant captions based on the visual content.

4.3 Tokenization and Padding:

After preprocessing, the captions undergo tokenization, a fundamental step in natural language processing, where the textual data is converted into numerical sequences. Each word in the preprocessed captions is assigned a unique index, enabling the model to understand and process the textual information in a structured format. Tokenization facilitates the transformation of raw text into a format suitable for input into neural networks, allowing for efficient computation and analysis. Following tokenization, padding is applied to ensure uniformity in sequence length. This involves adding padding tokens, typically zeros, to the beginning or end of sequences as needed to make them all of equal length. By standardizing the sequence length, padding enables seamless integration of the textual data with the model architecture, optimizing computational efficiency and ensuring consistent handling of input data across different samples.

4.4 Model Architecture:

In the proposed model architecture, two key components, the encoder and the decoder, work collaboratively to generate descriptive captions for images.

Encoder: The encoder, which forms the initial stage of the architecture, processes the extracted image features obtained from the VGG16 model. These features encapsulate the high-level visual content of the images and are fed into the encoder. The encoder comprises several layers, including a dropout layer, which helps prevent over fitting by randomly dropping out a fraction of input units during training, and a dense layer, which transforms the input features into a fixed-size representation. This fixed-size representation serves as the foundation for the subsequent caption generation process [17].

Decoder: the decoder, the complementary component of the architecture, takes the encoded image features and the padded tokenized sequences as inputs. The decoder is responsible for generating meaningful captions based on these inputs. It consists of an embedding layer, which converts the input tokenized sequences into dense vectors, facilitating better representation of the textual data. Following the embedding layer, a Long Short-Term Memory (LSTM) layer is employed to process the sequential nature of the input data and capture long-range dependencies within the sequences. The LSTM layer utilizes its memory cells to retain relevant information over time, enabling the model to generate coherent and contextually relevant captions [18].

LSTM: It is a type of recurrent neural network (RNN) architecture that is specifically designed to address the limitations of traditional RNNs in capturing long-range dependencies and handling vanishing or exploding gradient problems. LSTM networks incorporate memory cells, input gates, forget gates, and output gates, which enable them to effectively process sequential data while retaining and selectively updating information over time.

At the core of an LSTM unit is the memory cell, which serves as a mechanism to store and propagate information across time steps. The memory cell maintains an internal state, allowing it to remember relevant information from previous time steps and carry it forward to influence future predictions. This ability to retain long-term dependencies makes LSTMs well-suited for tasks such as sequence prediction, language modeling, and caption generation.

In addition to the memory cell, LSTMs incorporate three types of gates: input gates, forget gates, and output gates as shown in Figure 5. These gates regulate the flow of information into and out of the memory cell, allowing the LSTM to selectively update its internal state based on the input data and the current context.

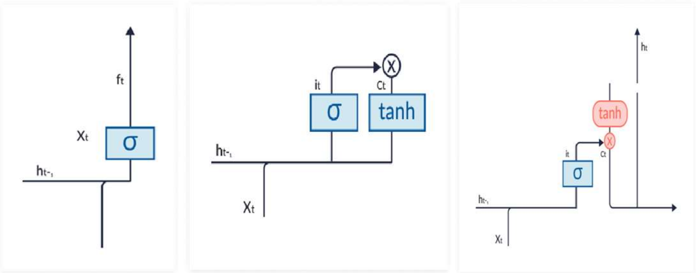


Figure 5. Forget Gate, Input Gate, Output Gate

1.2

The input gate determines how much of the new input information should be stored in the memory cell. It applies a sigmoid activation function to the input data, transforming it into a range of values between 0 and 1. This gate also employs a tanh activation function to regulate the magnitude of the input data, ensuring that it is properly scaled before being incorporated into the memory cell.

The forget gate, on the other hand, controls the extent to which the existing information in the memory cell should be retained or discarded. Like the input gate, the forget gate utilizes a sigmoid activation function to compute a forget factor for each element of the memory cell. This forget factor determines how much of the existing information should be retained (if close to 1) or forgotten (if close to 0) when processing new input data.

Finally, the output gate governs the flow of information from the memory cell to the LSTM's output. It applies a sigmoid activation function to the combined input and previous cell state, determining which parts of the current memory cell state should be exposed to the external layers of the network. Additionally, it uses a tanh activation function to regulate the magnitude of the output, ensuring that it is appropriately scaled before being passed to the next layer or used for making predictions**.**

4.5 Training the Model:

Training the model involves optimizing its parameters to minimize the discrepancy between the predicted captions and the ground truth captions in the training data. This process is typically performed using gradient-based optimization algorithms, with the choice of loss function and optimizer playing crucial roles in determining the model's performance.

During training, the model takes pairs of images and their corresponding tokenized caption sequences as input. The encoder processes the images to extract their features, which are then passed to the decoder along with the tokenized sequences. The decoder generates captions one token at a time, conditioning its predictions on both the encoded image features and the previously generated tokens. To evaluate the quality of the generated captions, the model's predictions are compared against the ground truth captions using a suitable loss function. In this case, categorical cross-entropy loss is commonly used, as it measures the dissimilarity between the predicted probability distributions over the vocabulary and the one-hot encoded representations of the ground truth tokens.

During back propagation, gradients of the loss function with respect to the model parameters are computed and used to update the parameters in the direction that minimizes the loss. The Adam optimizer, known for its adaptive learning rate mechanism and momentum-based updates, is often employed for this purpose due to its effectiveness in optimizing deep neural networks. The training process iterates over the entire training dataset multiple times (epochs), with the model gradually adjusting its parameters to improve its performance on the task. Additionally, techniques such as dropout regularization and teacher forcing may be employed to prevent over fitting and stabilize the training process, respectively.

Once the model has been trained sufficiently, its performance is evaluated on a separate validation set to assess its generalization ability and fine-tune any hyper parameters if necessary. Finally, the model's performance is evaluated on a held-out test set to obtain an unbiased estimate of its performance on unseen data, providing insights into its real-world applicability and effectiveness.

5. `EVALUATION:

The evaluation of the trained model using the BLEU (Bilingual Evaluation Understudy) score serves as a pivotal assessment of its capability to generate captions that align closely with the reference captions present in the dataset. BLEU is a widely adopted metric in natural language processing tasks, including image captioning, due to its simplicity and effectiveness in quantifying the quality of generated text.

The BLEU score computes the similarity between the generated captions and the reference captions by comparing their n-grams, which are contiguous sequences of n tokens (typically words). It operates by counting the number of overlapping n-grams between the generated and reference captions, normalized by the total number of n-grams in the generated captions. This normalization accounts for variations in caption length and ensures that longer captions are not unfairly penalized. The BLEU score ranges from 0 to 1, with higher scores indicating a greater degree of similarity between the generated and reference captions. A BLEU score of 1 signifies perfect overlap between the generated and reference captions, whereas a score of 0 indicates no overlap.

BLEU-1: 0.526863

BLEU-2: 0.295591

"BLEU-1" and "BLEU-2" refer to different n-gram orders used in computing the BLEU score.

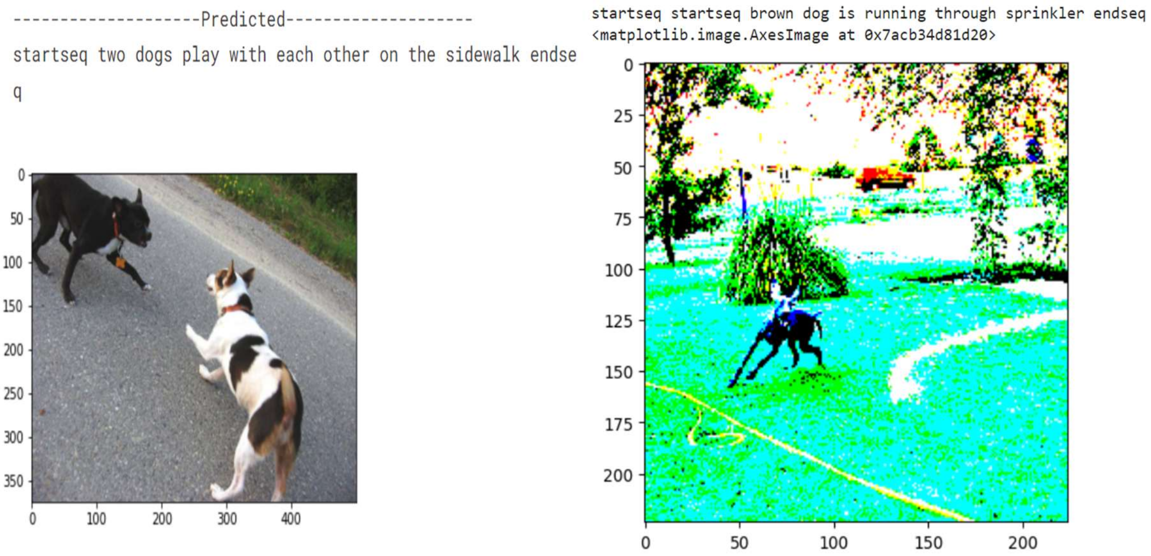
BLEU-1 represents the BLEU score computed based on unigrams (individual words). It measures the overlap of unigrams between the generated captions and the reference captions.

BLEU-2 represents the BLEU score computed based on bigrams (pairs of consecutive words). It measures the overlap of bigrams between the generated captions and the reference captions.

These scores provide insights into how well the generated captions align with the reference captions at different levels of linguistic granularity. For instance, BLEU-1 focuses on individual words, while BLEU-2 considers pairs of consecutive words. Typically, BLEU scores are reported for various n-gram orders (unigrams, bigrams, trigrams, etc.) to offer a comprehensive evaluation of the generated captions' quality.

6. CAPTION GENERATION AND VISUALIZATION:

Once trained, the model can generate captions for new images shown in Figure 6. The generated captions are visualized alongside the corresponding images to assess the quality and relevance of the generated descriptions.



```
--------------------Predicted--------------------
startseq two dogs play with each other on the sidewalk endse
q
```

```
startseq startseq brown dog is running through sprinkler endseq
<matplotlib.image.AxesImage at 0x7acb34d81d20>
```

Figure 6: Predicted Captions for Images.

7. CONCLUSION:

In conclusion, this work has demonstrated the effectiveness of employing deep learning techniques, specifically Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, for image captioning tasks. By leveraging the Flickr8k dataset and pre-trained models like VGG16, we successfully extracted high-level features from images and generated captions that capture their visual content. The model achieved promising results, as evidenced by the evaluation metrics such as BLEU scores. Through tokenization, padding, and careful architectural design, we created a robust framework capable of producing coherent and relevant captions for a diverse range of images.

Future Work:

Despite the achievements of this paper, several avenues for future exploration and improvement remain open. One potential area of focus is enhancing the model's ability to generate more diverse and contextually rich captions by incorporating advanced natural language processing techniques. Additionally, exploring larger and more diverse datasets could further enhance the model's generalization capabilities. Moreover, investigating attention mechanisms and reinforcement learning techniques could lead to more sophisticated captioning models that better understand the relationships between visual and textual information. Furthermore, deploying the model in real-world applications and collecting feedback from users would provide valuable insights for iterative refinement and optimization. Overall, continued research and development in this field hold the potential to significantly advance the capabilities of image captioning systems and their practical utility in various domains.

REFERENCES

[1] Abhaya Agarwal and Alon Lavie. 2023. Meteor, m-bleu and m-ter: Evaluation metrics forhigh-correlation with human rankings of machine translation output. In Proceedings of the ThirdWorkshop onStatistical Machine Translation. Association for Computational Linguistics, 115–118.

[2] Ahmet Aker and Robert Gaizauskas. 2022. Generating image descriptions using dependency relational patterns. In Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 1250–1258.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2023. Spice: Semanticpropositional image caption evaluation. In European Conference on Computer Vision. Springer, 382–398.

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould,and Lei Zhang. 2017. Bottom-up and top-down attention for image captioning and vqa. arXiv preprintarXiv:1707.07998 (2017).

[5] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. 2020. Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5561–5570.

[6] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, Trevor Darrell, Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, et al. 2021. Deepcompositional captioning: Describing novel object categories without paired training data. In Proceedings of the IEEEConference on Computer Vision and Pattern Recognition.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2019. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations (ICLR).

[8] Shuang Bai and Shan An. 2018. A Survey on Automatic Image Caption Generation. Neurocomputing. ACM Computing Surveys, Vol. 0, No. 0, Article 0. Acceptance Date: October 2018.0:30 Hossain et al.

[9] Satanjeev Banerjee and Alon Lavie. 2022. METEOR: An automatic metric for MT evaluationwith improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluationmeasures for machine translation and/or summarization, Vol. 29. 65–72.

[10] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled samplingfor sequence prediction with recurrent neural networks. In Advances in Neural Information Processing Systems. 1171–1179.

[11] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2023. A neural probabilistic language model. Journal of machine learning research 3, Feb, 1137–1155.

[12] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. Computational linguistics 22, 1 (1996), 39–71.

[13] Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, Barbara Plank, et al. 2016. Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. Journal of Artificial Intelligence Research (JAIR) 55, 409–442.

[14] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. Journalof machine Learning research 3, Jan (2023), 993–1022.

[15] Cristian Bodnar. 2018. Text to Image Synthesis Using Generative Adversarial Networks. arXiv preprint arXiv:1805.00676.

[16] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2018. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMODinternational conference on Management of data. AcM, 1247–1250.

[17] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory. ACM, 144–152.

[18] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Troster. 2011. Eye movementanalysis for activity recognition using electrooculography. IEEE transactions on pattern analysis and machineintelligence 33, 4 (2011), 741–753.

[19] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In Computer Vision and PatternRecognition (CVPR), 2011