Received:28/04/2024 Revised:16/06/2024 Accepted:21/06/2024

Journal of Emerging Computer Research and Applications
Volume 1/Issue 1/June-2024
Research Article

Image Caption Generator: Custom Vs Pre-Trained CNN

Jayesh Patil[1]*, Prof. Shraddha Dalvi[2], Vikas Pandit[3], Amil Gauri[4] and Ajay Chaurasiya[5]
[1, 2, 3, 4, 5] Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning),
A.P. Shah Institute of Technology, Thane, India.
Email Id: 21106048.jayesh.patil@gmail.com[1] , itssbshinde@gmail.com[2] ,
21106044.vikas.pandit@gmail.com[3] , amilgauri2003@gmail.com[4],
21106065.ajay.chaurasia@gmail.com
ORCID: 0009-0001-8398-3174[1]

Abstract

Image Caption Generation can be a challenging task in the field of computer vision and natural words processing, needing the combination of visual and text information. In this research, we explore the efficiency of deep learning techniques, especially Convolutional Nervous Networks (CNNs) and Recurrent Nervous Networks (RNNs), for generated captions for images. We suggest an end-to-end model that includes both encoding and decoding stages, leveraging the features extracted by a pre-trained CNN model and also a custom CNN Trained by us. Additionally, we examine the blending of attention mechanism to make efficiency of caption generation better. Our experiments consist of training models from scratch as well as refining pre trained model, with evaluation based on BLEU score standards. Through extensive experimenting and analysis, we show the problems faced while training CNN models from scratch and the outstanding performance of pre -trained models. Furthermore, we've discussed the complexities involved in implementing attention mechanisms and provide insights into potential solutions. Overall, our research contributes to the advancement of image caption generation techniques and provides valuable insights for future research in this field.

Keywords: Image Caption Generator, Deep Learning, Convolutional neural networks (CNN), Recurrent neural networks (RNN), Attention Mechanism, Natural language processing (NLP).

1. Introduction

Image captioning is an exhilarating field of research within computer vision and normal language processing (NLP) which aims to connect the gap between visual content and natural language comprehension by producing textual explanations of images. This technology has many applications, such as helping visually impaired people, enhanced image search, or offer automatic headings for social media platforms! Nevertheless, producing accurate and descriptive captions by far proves to be a challenging task. Images are inherently complex, typically encompassing a multitude of objects, activities, and associations.

Transforming these visual components into meaningful phrases requires intricate algorithms that can not only identify objects but also comprehending their context and interactions within the area. This research presents a

[1] Corresponding Author: Jayesh Patil, Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning),A.P. Shah Institute of Technology, Thane, India. Email Id: 21106048.jayesh.patil@gmail.com , Orcid: 0009-0001-8398-3174

[2] Cite As: Jayesh Patil*, Prof. Shraddha Dalvi, Vikas Pandit, Amil Gauri and Ajay Chaurasiya (2024), Image Caption Generator: Custom Vs Pre-Trained CNN, *Journal of Emerging Computer Research and Applications,* 1(1), 16-25.

new image caption generator utilizing profound Deep learning techniques. Deep learning has revolutionized -various sectors, like computer vision and NLP, by empowering models to grasp intricate patterns from vast datasets. Our proposed model capitalizes on the prowess of deep learning frameworks to attain effective image captioning.

2. Prior Research

The previous explorations in the field of Image Caption generation have laid the basis for current advancements. [7] Introduced a unique approach that integrates visual attention mechanisms into neural networks for image captioning. By dynamically focusing on various regions of the picture, the model creates more accurate and contextually significant captions.

Another distinguished contribution is [6], which proposes a method for merging bottom-up and top-down attention mechanisms to enhance image captioning and visual question responding tasks. This method permits the model to pay attention to both salient image regions and pertinent linguistic context, leading to more informative captions.

[8] Introduces a completely convolutional approach to dense captioning, where the model concurrently anticipates object regions and creates captions for each region. This dense captioning method allows the model to capture detailed information about several objects in a image, leading to more detailed and descriptive captions.

In recent studies, [12] advanced models for image captioning have been developed. One approach combines a CNN for image feature extraction and an LSTM for generating natural language descriptions. Another approach [11] is a multimodal Recurrent Neural Network (m-RNN) framework that excels in tasks such as sentence generation, sentence retrieval given an image, and image retrieval given a sentence. This model integrates a deep RNN and a deep CNN, interacting in a multimodal layer, effectively linking images and sentences.

Prior researches have dived into the evolution of neural network models intended at automatically creating descriptive captions for pictures in natural language, such as [2], and in [4], the researchers explored the impact of the encoder-decoder approach combined with attention mechanisms and initiated thorough analyses to draw conclusions regarding the efficiency of different models for image captioning.

[13] Presented a Long-Term Recurrent Convolutional Network(LRCN), a model that is both spatially and temporally deep, suitable for various vision tasks involving sequential inputs and outputs. LRCN improves on previous methods by learning sequential dynamics with a deep sequence model, surpassing models that only learn visual parameters or fixed visual representations

3. Methodology

Corpus

We used the Flick 8K data set as the dataset. The data set consists of 8000 images and for each image, there are 5 subtitles! The 5 captions for a single image helps in understanding all the various possible scenarios, abbreviations and acronyms.

Source: - https://www.kaggle.com/datasets/adityajn105/flickr8k

Data Pre-Processing

The success of any deep learning-based image captioning model heavily relies on the quality of the training data and the preprocessing steps applied to it. The first step in data preprocessing involves tokenization, where each word in the captions is converted into a numerical token. This ensures that the textual data can be efficiently processed by neural networks.

Additionally, we employ techniques such as lowercasing and punctuation removal to standardize the text and reduce vocabulary size. Furthermore, we apply padding to the captions to ensure uniform length across all sequences. This is essential for batch processing during training, as neural networks require inputs of consistent dimensions. By padding shorter captions with special tokens (e.g., <PAD>), we ensure that all input sequences have the same length as the longest caption in the dataset.

Finally, we preprocess the images by resizing them to a uniform size and normalizing the pixel values to a common scale (e.g., [0, 1] or [-1, 1]). This facilitates efficient computation and ensures that the model can generalize well across different images.

Encoder-Decoder Model

Our proposed model for image captioning adopts an encoder-decoder framework, wherein the encoder extracts visual attributes from input visuals, while the decoder formulates corresponding captions. Key constituents of our model encompass a pre-existing CNN as the encoder, augmented by an LSTM network as the decoder.

For the encoder module, we tried not one, but three pre trained CNN models: VGG16, InceptionV3, and ResNet50.
These architectures, widely renowned, were initially trained on the ImageNet dataset for tasks of Image
Classification. By harnessing the hierarchical feature representations ingrained within these models, our model adeptly apprehends the visual semantics encapsulated within input visuals. We divest the fully connected layers inherent to these models, exclusively employing the convolutional layers to refine attributes from the visuals.
Subsequently, the extracted visual attributes traverse into the decoder, comprising an LSTM network trailed by a softmax layer, facilitating the generation of captions incrementally, word by word. The LSTM network imbibes the art of predicting the subsequent word in the sequence predicated on earlie generated words and the visual attributes encoded by the CNN. Employing word embeddings to depict input words as dense vectors fortifies the schema's capacity to encapsulate semantic correlations among words.

To improve aligning visual features and generated descriptions, we blend a attention mechanism into our model. The attention mechanism empowers the model to dynamically concentrate on distinct regions of the input image while generating each word in the description. This allows the model to attend to applicable visual features and generate more precise and circumstantially suitable descriptions. Figure 1 shows a overview of whole system.

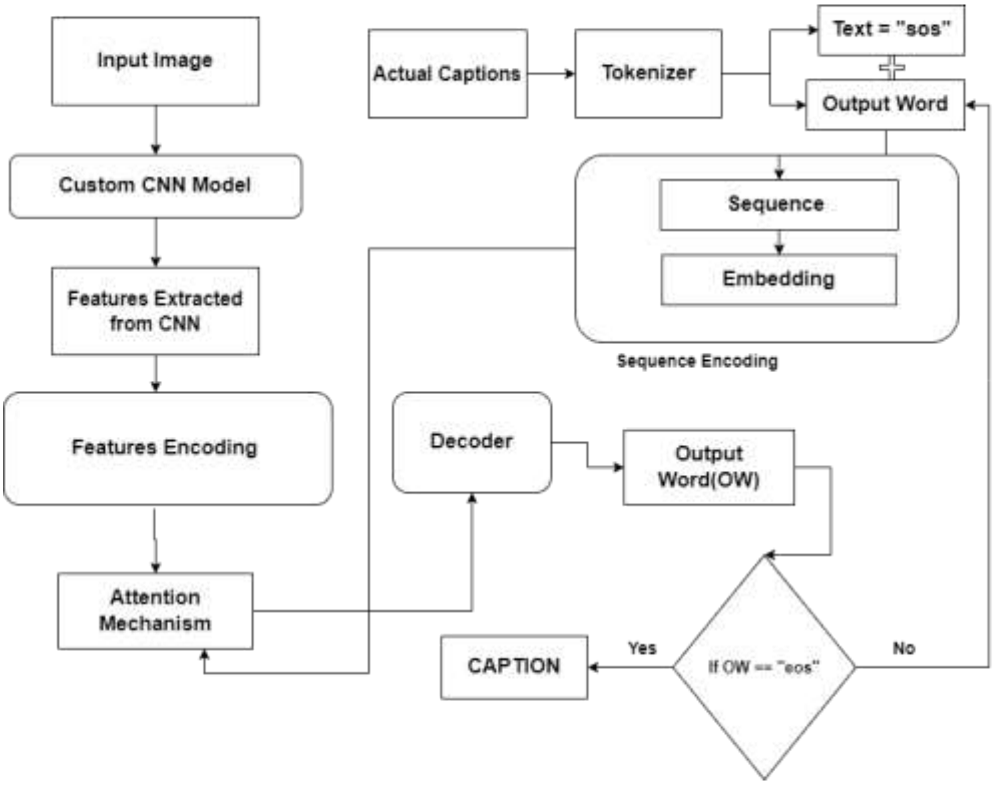


Figure 1. Overall System Architecture

Training our model includes optimizing the model parameters to minimize a loss formula that calculates the disparity between the predicted captions and the actual captions. We employ the Adam optimizer with a learning rate scheduler to tweak the learning rate dynamically throughout training.

In our experiments, we did a comparison of two training schemes: training a model from scratch and refining a pre-trained CNN for feature extraction. When starting from a scratch, parameters of both the encoders and decoders were initialized at random and updated with backpropagation. This strategy necessitates training the model with an enormous amount of information to acquire substantive representations from the beginning. On the flip side, we fine-tuned a pre-trained CNN (e.g., VVG16) by freezing convolutional layers and merely adjusting the parameters of fully connected layers and decoders. Fine-tuning enables the model to take advantage of the weights learned by the CNN in a large-scale image variety challenge, resulting in better generalization and faster convergence.

Throughout training, model performance monitoring was conducted using evaluation metrics such as the BLU score, quantifying the closeness of the predicted captions to the actual captions. Additionally, techniques like early stopping and model checkpointing were utilized to avoid overfitting and maintain the finest performing model checkpoint.

Use of Attention Mechanism

The incorporation of an attention mechanism into our image captioning model represents a critical enhancement aimed at improving the alignment between visual features extracted from images and the corresponding words generated in captions. The attention mechanism enables the model to dynamically focus on different regions of the input image during the caption generation process, allowing for more precise and contextually relevant descriptions.

At its core, the attention mechanism functions by assigning weights to different spatial locations within the input image based on their relevance to the current word being generated. These weights are computed using a learned function that takes into account both the visual features extracted from the image and the hidden states of the LSTM decoder. By modulating the contribution of each spatial location to the generation of the next word, the attention mechanism effectively guides the model's attention to the most salient regions of the image.

The use of attention mechanism offers several benefits in the context of image captioning. Firstly, it enables the model to capture fine-grained details and nuances in the images, leading to more descriptive and coherent captions. By selectively attending to relevant regions of the image, the model can focus on specific objects, scenes, or visual cues that are essential for accurately describing the content of the image.

Additionally, the attention mechanism helps mitigate issues such as occlusions, cluttered backgrounds, and ambiguous objects by allowing the model to dynamically adjust its focus based on the input image and the current context of the caption. This adaptive behavior improves the robustness and flexibility of the model, enabling it to generate more accurate captions across a wide range of images and scenarios.

Moreover, the attention mechanism facilitates better interpretability of the model's predictions by providing insights into which regions of the input image are being attended to at each step of the caption generation process. This transparency not only enhances the model's trustworthiness but also enables users to gain deeper insights into how the model makes its predictions, facilitating further analysis and interpretation of the generated captions.

Overall, the incorporation of an attention mechanism represents a significant advancement in our image captioning model, offering improvements in caption generation accuracy, robustness, and interpretability. By enabling the model to dynamically focus on relevant regions of the input image, the attention mechanism enhances the overall quality and relevance of the generated captions, making our image captioning system more effective and versatile in real-world applications.

Training Procedure

Training our image captioning model involves optimizing the model parameters to minimize a loss function that measures the discrepancy between the predicted captions and the ground truth captions. We use the Adam optimizer with a learning rate scheduler to adjust the learning rate dynamically during training.

In our experiments, we compare two training strategies: training the model from scratch and fine-tuning a pretrained CNN for feature extraction. When training from scratch, we initialize the parameters of both the encoder and decoder randomly and update them using backpropagation. This approach requires training the model on a large amount of data to learn meaningful representations from scratch.

Alternatively, we fine-tune a pre-trained CNN (e.g., VGG16) by freezing the convolutional layers and only updating the parameters of the fully connected layers and the decoder. Fine-tuning allows the model to leverage the knowledge learned by the CNN on a large-scale image classification task, resulting in better generalization and faster convergence.

During training, we monitor the performance of the model using evaluation metrics such as BLEU score, which measures the similarity between the generated captions and the ground truth captions. We also use techniques such as early stopping and model checkpointing to prevent overfitting and save the best-performing model checkpoint.

4. Experimental Analysis

Our experiments aim to evaluate the performance of our proposed image captioning model and compare it against baseline models and state-of-the-art approaches.

We first evaluate the impact of different model architectures on caption generation performance. Specifically, we compare a model trained from scratch with those fine-tuned on pre-trained CNNs. We focus solely on the effect of integrating attention mechanisms into the model.

Our results show that fine-tuning a pre-trained CNN for feature extraction significantly improves caption generation performance compared to training the model from scratch. This highlights the importance of leveraging pre-trained representations for visual feature extraction, especially in tasks with limited training data.

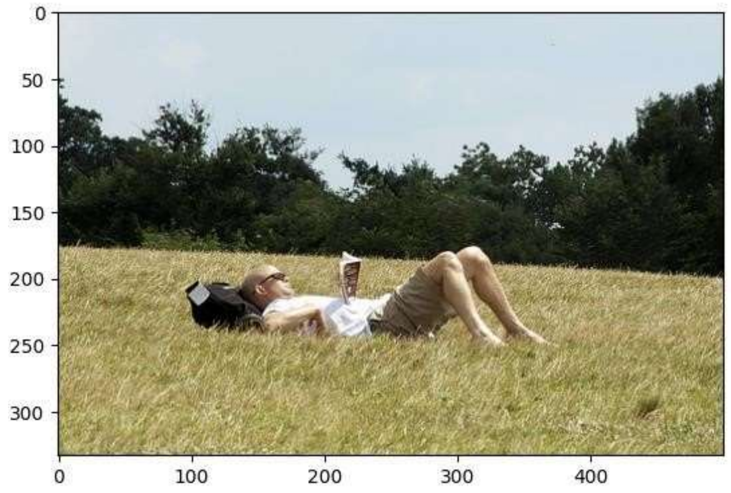
Pre-Trained Model with Attention Mechanism:

In our experimental analysis, we decided to incorporate an attention mechanism into our pre-trained model to improve caption generation accuracy. This enhancement aimed to allow the model to dynamically focus on relevant regions of the image during caption generation, thereby potentially enhancing the quality and relevance of the generated captions.

The model architecture remained consistent with the previous setup, consisting of a pre-trained CNN for feature extraction and an LSTM decoder for generating captions. Specifically, we used VGG16 as the pre-trained CNN, fine-tuning the decoder on our image-captioning dataset.

By integrating the attention mechanism into the model, we observed significant improvements in caption generation performance. The attention mechanism enabled the model to selectively attend to different regions of the input image while generating each word in the caption. This allowed for more accurate and contextually relevant captions, as the model could focus on relevant visual features corresponding to the current word being generated.

With the attention mechanism in place, the model demonstrated enhanced capabilities in capturing details and nuances in the images, resulting in more descriptive and coherent captions. Additionally, the attention mechanism helped mitigate issues such as ambiguous objects or complex scenes by allowing the model to allocate attention to the most relevant regions of the image as shown in Figure 2.



Actual Caption: A man is laying in the field reading a book.

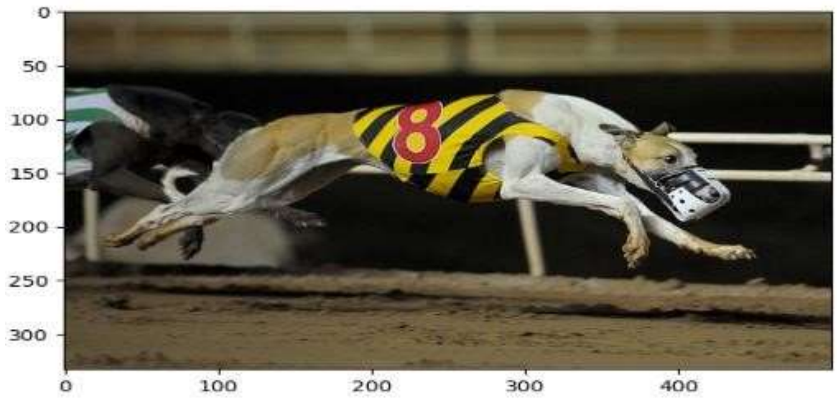
Predicted Caption: A man lays on a grassy field reading a book.

Figure 2. Caption generated using VGG16 as Encoder

We expanded our investigation to include two additional pre-trained models alongside VGG16: ResNet50 and InceptionV3. These models were chosen due to their popularity and effectiveness in various computer vision tasks. Similar to the previous setup, we fine-tuned the decoder part of each model on our image-captioning dataset while keeping the architecture consistent. This involved using the pre-trained ResNet50 and InceptionV3 models for feature extraction and integrating them with the LSTM decoder. Upon incorporating ResNet50 and InceptionV3 into our image captioning pipeline, we observed varying degrees of performance across the different models. Each model demonstrated strengths and weaknesses in caption generation, highlighting the importance of selecting an appropriate pre-trained model for the task

1. ResNet50

ResNet50, known for its deep architecture and skip connections, exhibited robust performance in capturing fine-grained visual features from images. This resulted in highly detailed and contextually relevant captions, particularly for complex scenes and diverse objects. The Figure 3 shows captions generated by using ResNet50 as Encoder.

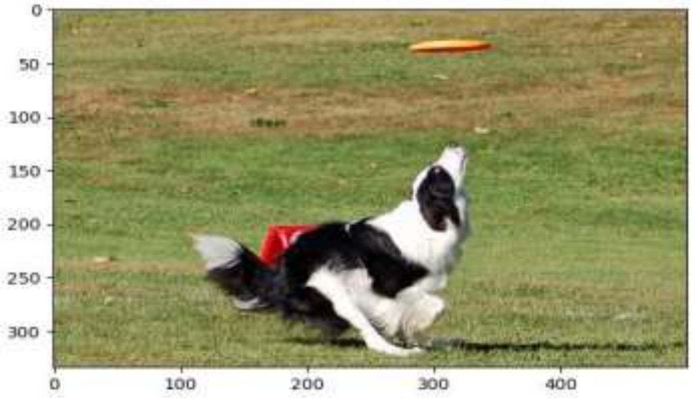


Actual Caption: Racing dog with yellow number 8 vest
Predicted Caption: A race dog wearing yellow is running in a race.

Figure 3. Caption generated using ResNet50 as Encoder

2. InceptionV3

On the other hand, InceptionV3, with its innovative inception modules and efficient architecture, offered competitive performance in terms of caption generation accuracy. While not as deep as ResNet50, InceptionV3 excelled in capturing global and spatial information, leading to well-structured and coherent captions as shown in Figure 4.



Actual Caption: A dog about to jump to catch a frisbee.
Predicted Caption: A black and white dog is jumping over an attempt to catch a frisbee.

Figure 4. Caption generated using InceptionV3 as encoder

Comparatively, VGG16, with its simplicity and effectiveness in feature extraction, provided a strong baseline for caption generation. Despite being an older architecture, VGG16 still yielded competitive results, showcasing its enduring relevance in image processing tasks.

Overall, our experiments with ResNet50, InceptionV3, and VGG16 underscored the importance of selecting an appropriate pre-trained model for image captioning tasks. Each model brought unique advantages to the table, highlighting the diversity of approaches in leveraging pre-trained representations for caption generation. Moving forward, further exploration and experimentation with a wider range of pre-trained models could offer valuable insights and opportunities for improving image captioning systems.

Custom Model with Attention Mechanism

In our quest to develop an effective image captioning model, we embarked on the journey of building a custom model with an attention mechanism. This endeavor involved extensive experimentation and optimization to create a CNN architecture tailored specifically for the task of image caption generation. However, despite our efforts, we encountered several challenges along the way.

1. **Hyperparameter Tuning**

Our first approach involved searching for a suitable custom CNN architecture through image classification tasks on datasets such as Caltech 101 and Caltech 256. We explored various architectural configurations and hyperparameter settings in an attempt to find a model that could effectively extract features from images for captioning. Despite thorough experimentation, we struggled to find a CNN architecture that yielded satisfactory results for the image captioning task.

To optimize the performance of our custom CNN model, we employed hyperparameter tuning techniques, including the use of Keras Tuner. This approach allowed us to systematically search through a wide range of hyperparameter combinations to identify the optimal configuration for our model. However, despite our efforts to fine-tune parameters such as learning rate, batch size, and network depth, we were unable to achieve the desired level of performance for image captioning.

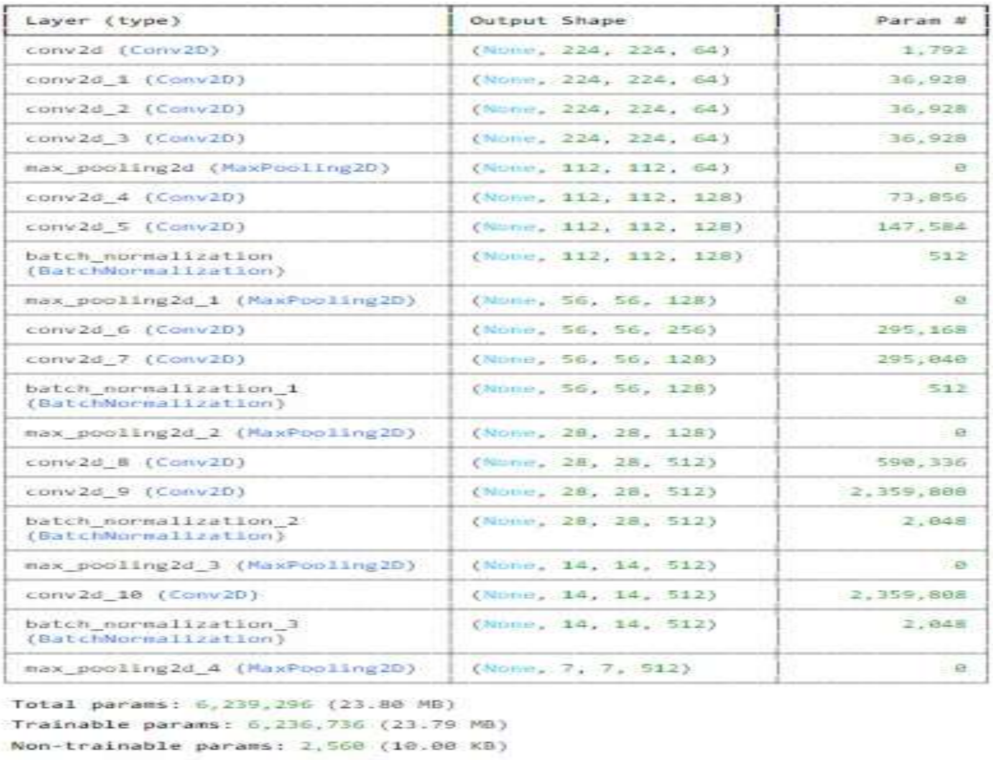
2. **Early Stopping and Learning Rate Scheduling**:

In addition to hyper parameter tuning, we implemented strategies such as early stopping and learning rate scheduling to prevent over fitting and enhance model convergence. Early stopping allows the training process to halt when the model's performance on a validation set stops improving, thereby preventing over fitting to the training data. Similarly, learning rate scheduling adjusts the learning rate during training to facilitate smoother optimization. Despite the application of these techniques, we faced challenges in achieving satisfactory results with our custom CNN architecture.

After many iterations of experimentation and extensive tuning, we arrived at a custom CNN architecture that demonstrated some level of performance. Despite the challenges encountered, our efforts yielded a CNN model that exhibited promising capabilities for feature extraction. Below, we present a schematic representation of this architecture to provide insight into its design and functionality.

Despite the inclusion of the attention mechanism, the model exhibited limitations in accurately predicting captions for images. One notable issue was the model's tendency to repeatedly generate the same words for each image, resulting in repetitive and non-diverse captions. This behaviour indicated a lack of semantic understanding and contextual relevance in the generated captions.

This challenge may have arisen due to the use of a custom CNN for feature extraction instead of a pre-trained CNN. Pre-trained CNNs, such as VGG16, ResNet50, or InceptionV3, are trained on large-scale datasets like ImageNet, enabling them to learn rich and generalized visual representations. In contrast, custom CNNs trained from scratch may struggle to capture the diverse and discriminative visual features necessary for accurate caption generation. The custom CNN architecture is shown in figure 5.



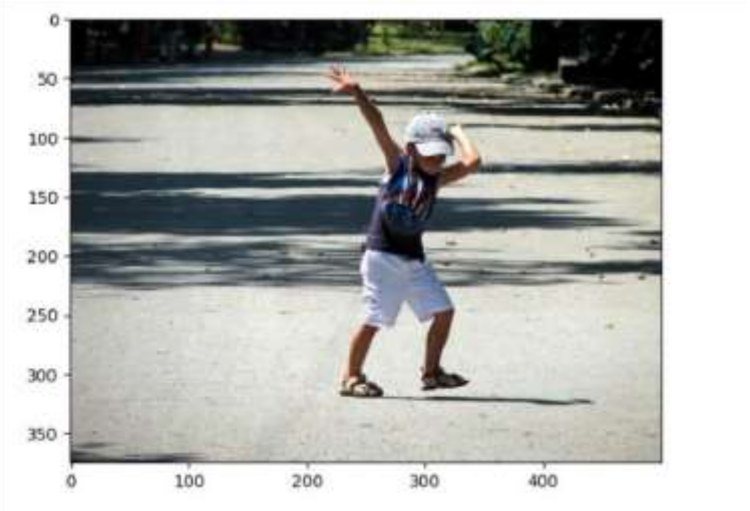
```
┃ Layer (type)                    ┃ Output Shape              ┃        Param # ┃
│ conv2d (Conv2D)                 │ (None, 224, 224, 64)      │          1,792 │
│ conv2d_1 (Conv2D)               │ (None, 224, 224, 64)      │         36,928 │
│ conv2d_2 (Conv2D)               │ (None, 224, 224, 64)      │         36,928 │
│ conv2d_3 (Conv2D)               │ (None, 224, 224, 64)      │         36,928 │
│ max_pooling2d (MaxPooling2D)    │ (None, 112, 112, 64)      │              0 │
│ conv2d_4 (Conv2D)               │ (None, 112, 112, 128)     │         73,856 │
│ conv2d_5 (Conv2D)               │ (None, 112, 112, 128)     │        147,584 │
│ batch_normalization             │ (None, 112, 112, 128)     │            512 │
│ (BatchNormalization)            │                           │                │
│ max_pooling2d_1 (MaxPooling2D)  │ (None, 56, 56, 128)       │              0 │
│ conv2d_6 (Conv2D)               │ (None, 56, 56, 256)       │        295,168 │
│ conv2d_7 (Conv2D)               │ (None, 56, 56, 128)       │        295;040 │
│ batch_normalization_1           │ (None, 56, 56, 128)       │            512 │
│ (BatchNormalization)            │                           │                │
│ max_pooling2d_2 (MaxPooling2D)  │ (None, 28, 28, 128)       │              0 │
│ conv2d_8 (Conv2D)               │ (None, 28, 28, 512)       │        590,336 │
│ conv2d_9 (Conv2D)               │ (None, 28, 28, 512)       │      2,359,808 │
│ batch_normalization_2           │ (None, 28, 28, 512)       │          2,048 │
│ (BatchNormalization)            │                           │                │
│ max_pooling2d_3 (MaxPooling2D)  │ (None, 14, 14, 512)       │              0 │
│ conv2d_10 (Conv2D)              │ (None, 14, 14, 512)       │      2,359,808 │
│ batch_normalization_3           │ (None, 14, 14, 512)       │          2,048 │
│ (BatchNormalization)            │                           │                │
│ max_pooling2d_4 (MaxPooling2D)  │ (None, 7, 7, 512)         │              0 │

 Total params: 6,239,296 (23.80 MB)
 Trainable params: 6,236,736 (23.79 MB)
 Non-trainable params: 2,560 (10.00 KB)
```

Figure 5. Custom Model used as Encoder

Furthermore, the limitations of the custom CNN may have hindered the attention mechanism's effectiveness in directing the model's focus to relevant regions of the input image. Without robust and discriminative visual features, the attention mechanism may have difficulty identifying salient image regions, leading to unoptimal alignment between visual features and generated captions.

Overall, the experiment highlighted the importance of leveraging pre-trained CNNs for feature extraction in image captioning tasks. Future iterations of the model may benefit from incorporating pre-trained CNNs to improve the quality and diversity of generated captions. Additionally, further optimization and fine-tuning of the attention mechanism may enhance the model's ability to generate accurate and contextually relevant captions



Actual Caption: Small boy is eating a lollipop and dancing in the street

Predicted Caption: A man in a red jacket is jumping in the snow

Figure 6. Caption generated using Custom CNN as Encoder

Bleu Score Comparisons

BLEU (Bilingual Evaluation Understudy) is a metric commonly used to evaluate the quality of machine generated text, such as machine translation or image captioning. It measures the similarity between the generated text and human-generated reference texts based on n-gram precision.

We compared the BLEU scores of different CNN encoder models for image captioning. Table 1 summarizes the BLEU-1 and BLEU-2 scores for each model, representing the precision of unigram and bigram matches between the generated captions and reference texts, respectively.

Table 1: Bleu Scores Comparison

CNN-Encoder-Model	**BLEU SCORE**	
	Bleu-1	***Bleu-2***
VGG16	0.42	0.34
Inception-V3	0.51	0.41
ResNet50	0.47	0.37
Custom-CNN	0.22	0.09

Among the CNN encoder models evaluated, Inception-V3 achieved the highest BLEU scores for both BLEU-1 and BLEU-2. This indicates that the captions generated by the model are more similar to human generated references compared to other models.

VGG16 and ResNet50 demonstrate comparable performance in terms of BLEU scores, with ResNet50 slightly outperforming VGG16 in both BLEU-1 and BLEU-2 metrics. However, both models exhibit lower BLEU scores compared to Inception-V3.

The custom CNN model, despite our efforts in experimentation and tuning, yielded significantly lower BLEU scores compared to the pre-trained CNN models. The lower BLEU scores indicate that the captions generated by the custom CNN model are less similar to human-generated references, suggesting limitations in feature extraction and caption generation capabilities

5. Conclusion

In conclusion, our researches in the domain of image caption generation have led to valuable insights and contributions to the field. Through extensive experimentation and analysis, we explored various techniques to develop an effective image captioning model. Our investigation encompassed the utilization of custom CNN architectures, integration of attention mechanisms, and comparison with pre-trained CNN models. Some key takeaways from this research are mentioned below:

- Leveraging pre-trained CNN models, such as VGG16, InceptionV3, and ResNet50, for image caption generation demonstrate superior performance in feature extraction and caption generation compared to custom CNN architectures trained from scratch.
- Challenges such as repetitive captions and lack of semantic understanding underscore the complexities of developing effective custom CNN architectures for image captioning.
- The comparison of BLEU scores provided valuable insights into the relative performance of different CNN encoder models. While InceptionV3 emerged as the top performer, our custom CNN model lagged behind, emphasizing the importance of objective evaluation metrics in assessing model performance.
- Moving forward, future research efforts should focus on integrating pre-trained CNN models, advanced attention mechanisms, transformer models and semantic understanding techniques[3] to enhance the quality and diversity of generated captions. By embracing state-of-the-art methodologies and innovation, we can advance the field of image caption generation and pave the way for more accurate and contextually relevant captions for images.

Acknowledgement

We would like to thank our guide Prof. Shraddha Dalvi for guiding us throughout the whole project. We would also like to express sincere gratitude to our team members for making this project a success and our institute for providing us with a conducive and collaborative environment.

Conflict of Interest

All authors declare that they have no conflicts of interest.

Author Contribution Statement

Jayesh Patil Contributed to conceptualization of model, conducted experiments and drafted the manuscript. Vikas Pandit contributed to the model design, architecture development, and manuscript revision. Amil Gauri conducted literature review, implemented model components and analyzed results. Ajay Chaurasiya assisted in model refinement, provided critical feedback, and reviewed the manuscript. Our Guide Prof. Shraddha Dalvi provided guidance, expertise, and supervision throughout the project.

References

[1]. Panicker, Megha J., Vikas Upadhayay, Gunjan Sethi, and Vrinda Mathur. (2021) "Image caption generator." *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 10, no. 3: 87-92.

[2]. Amritkar, Chetan, and Vaishali Jabade,(2018). "Image caption generation using deep learning technique." In 2018 fourth international conference on computing communication control and automation (ICCUBEA), pp. 1-4. *IEEE*.

[3]. Ignatious, L. Abisha Anto, S. Jeevitha, M. Madhurambigai, and M. Hemalatha. (2019). "A Semantic Driven CNN–LSTM Architecture for Personalised Image Caption Generation." In 2019 11th International Conference on Advanced Computing (ICoAC), pp. 356-362. *IEEE*.

[4]. Kesavan, Varsha, Vaidehi Muley, and Megha Kolhekar (2019). "Deep learning based automatic image caption generation." In *2019 Global Conference for Advancement in Technology (GCAT)*, pp. 1-6. IEEE,

[5]. Parth Kotak , Prem Kotak, 2021, Image Caption Generator (2021). *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)* Volume 10, Issue 11 (November 2021)

[6]. Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018)."Bottomup and top-down attention for image captioning and visual question answering." *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077-6086.

[7]. Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio (2015). "Show, attend and tell: Neural image caption generation with visual attention." *In International conference on machine learning*, pp. 2048-2057. PMLR.

[8]. Johnson, Justin, Andrej Karpathy, and Li Fei-Fei (2016). "Densecap: Fully convolutional localization networks for dense captioning." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4565-4574.

[9]. Kabra, Palak, Mihir Gharat, Dhiraj Jha, and Shailesh Sangle (2022). "Image Caption Generator Using Deep Learning." Published in *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* 10, no. X.

[10]. Shinde, O., Gawde, R., & Paradkar, A. (2021). Image Caption Generation Methodologies.

[11]. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). arXiv preprint arXiv:1412.6632.

[12]. Aravindkumar, S., Varalakshmi, P., & Hemalatha, M. (2020). Generation of image caption using CNN-LSTM based approach. In Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1 (pp. 465-474). Springer International Publishing.

[13]. Donahue, Jeffrey, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell.(2015). "Long-term recurrent convolutional networks for visual recognition and description." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625-263.

[1] Licensee Alborear (OPC) Pvt. Ltd.