https://doi.org/10.2352/EI.2025.37.8.IMAGE-266

Optimization of Image Captioning Networks Using Targeted Component Pruning Method

Jishu Sen Gupta[+a], Yogendra Rao Musunuri[+b], Ih-Man Seo[b], and Oh-Seol Kwon[*b]

[a] Department of Mathematical Sciences, Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, 221005, India.

[b]Department of Control and Instrumentation Engineering, Changwon National University, Changwon, Gyeongnam 644-731, Korea.

Abstract

Deep learning models have significantly advanced, leading to substantial improvements in image captioning performance over the past decade. However, these improvements have resulted in increased model complexity and higher computational costs. Contemporary captioning models typically consist of three components such as a pre-trained CNN encoder, a transformer encoder, and a decoder. Although research has extensively explored the network pruning for captioning models, it has not specifically addressed the pruning of these three individual components. As a result, existing methods lack the generalizability required for models that deviate from the traditional configuration of image captioning systems. In this study, we introduce a pruning technique designed to optimize each component of the captioning model individually, thus broadening its applicability to models that share similar components, such as encoders and decoder networks, even if their overall architectures differ from the conventional captioning models. Additionally, we implemented a novel modification during the pruning in the decoder through the cross-entropy loss, which significantly improved the performance of the image-captioning model. Furthermore, we trained and validated our approach on the Flicker8k dataset and evaluated its performance using the CIDEr and ROUGE-L metrics.

Introduction

In recent years, the field of image captioning has witnessed significant advancements, primarily driven by the development of deep learning models. These enhancements have improved performance metrics considerably; however, they have also led to increased model complexity and elevated computational demands. Over the past decade, research focused on deep neural networks for image captioning has significantly enhanced model performance. Notably, the CIDEr [1] scores for state-of-the-art models on the MS-COCO dataset have risen from 66 to over 130 points. However, these advancements have typically resulted in substantial increases in model size, exemplified by the growth in decoder size from 12 million to 55 million parameters.

Contemporary image-captioning models typically consist of three primary components: a pre-trained convolutional neural network (CNN) encoder, a transformer encoder, and a transformer decoder. To mitigate the increase in model size, various pruning techniques have been developed to remove non-essential weights from the network. These pruning methods offer multiple benefits, including enhanced speed, reduced storage requirements, and lower energy consumption, especially during deployment.

Despite these developments, current research on network pruning for these models has not thoroughly addressed the distinct components, resulting in methods that are not universally applicable to models with varying architectures. This limitation restricts the generalizability of pruning techniques, particularly for models that incorporate similar components but differ in their overall structure of the models. Our objective is to establish a generalized pruning strategy applicable to various models featuring encoder and decoder networks. The contribution of this paper as follows:

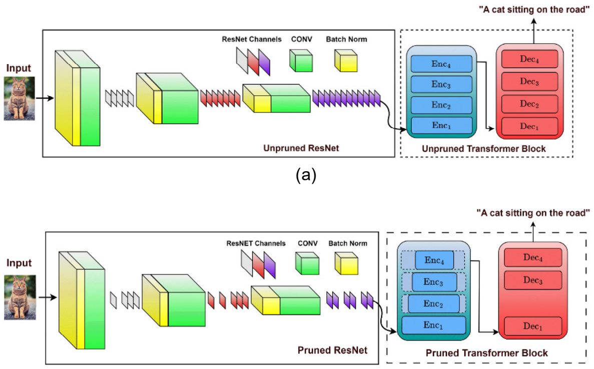
(a). We proposed a novel approach employing distinct pruning techniques tailored to each component of the ResNet-transformer-based model for image captioning (RTIC).

(b). Additionally, we implemented a novel modification to the pruning process in the decoder by incorporating the cross-entropy loss, which significantly improved the performance of the image-captioning model. Moreover, we trained and validated our approach on the Flickr8k dataset and evaluated its performance using the CIDEr and ROUGE-L metrics.

Related Work

Recent studies [2] [3] have conducted end-to-end pruning of image-captioning models. Tan et al. [2] developed a super-mask pruning technique that implements continuous and gradual sparsification during the training phase, based on parameter sensitivity in an end-to-end fashion. In [2], they noted the scarcity of their previous work on pruning the image captioning models due to two primary challenges: first, the presence of weights that are shared and reused across time steps, which complicates the application of variational pruning methods designed for feed-forward networks; second, the inherent complexity of the multi-modal task of image captioning, requiring any proposed method to perform effectively across both image and language domains.

Furthermore, methods [2] and [3] address the pruning of image captioning models within an end-to-end framework. However, their approaches are not easily generalizable to models with similar yet distinct architectures. For instance, image-captioning models often include vision transformer (ViT) encoders and language model (LM) decoders.



(a)

(b)

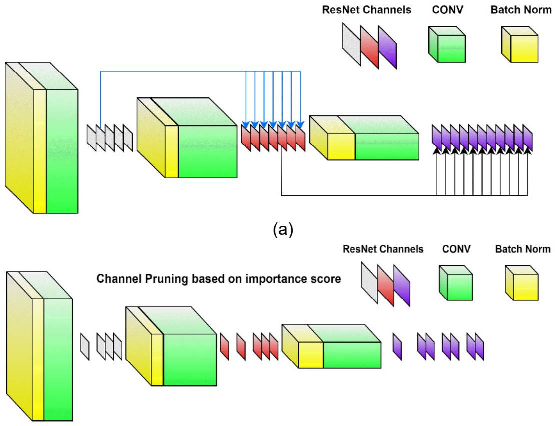
Figure 1. *Pipeline of the ResNet-Transformer based image captioning model (RTIC). (a). Unpruned model and (b). Pruned model with the proposed targeted component pruning method.*

[+] Equal contribution
[*] Corresponding author

Therefore, we propose a new method for pruning that treats each component—namely, the pre-trained ResNet encoder, the transformer encoder, and the transformer decoder networks—separately, as depicted in Figure 1. This method ensures that the pruning processes for each component are independent of one another. The unpruned model is shown in Figure 1(a), while the pruned model is shown in Figure 1(b).

Method

To implement the pruning techniques, we utilized a framework for the image-captioning model as depicted in Figure 1(a), which comprises three main components: a pre-trained ResNet serving as the backbone, a transformer encoder, and a transformer decoder. Figure 1(b) shows the pruned model after applying the targeted component pruning method to each of these components separately: channel pruning on ResNet, width pruning on the transformer encoder, and depth pruning on the transformer decoder.



(a)

(b)

Figure 2. Comparison of pruning and unpruning ResNet (a). Unpruned Res-Net, and (b). Pruned ResNet.

Targeted channel pruning of the ResNet component

The ResNet serves as a backbone of the RTIC, primarily responsible for feature extraction. Prior to targeted pruning, we trained the ResNet model on Tiny Image-Net dataset [4]. In order to implement the pruning on the pre-trained model, traditional pruning methods, which involve removing redundant channels through the use of a sparsity-inducing term in a pretrained network followed by fine-tuning. These approaches face several challenges; for example, group lasso technique employed in these methods is computationally demanding, challenging to converge, and often leads to diminished performance due to the simplified model architecture. Kethan et al. [5] introduced a channel pruning method applicable across all layers of a network, allowing for a varying number of channels to be pruned across different layers. This method is designed for a standard ResNet-101 architecture incorporating convolution-batch normalization, and ReLU activation. Let B denote the current mini-batch, a standard BN layer performs the following affine transformation for each of the i-th feature map $z_i = \mathbb{R}^{r\times r}$, for $i \in \{1, 2, . . ., n\}$ as shown in (1).

$$\hat{z}_i = \frac{z_i^{(in)} - \mu_{B_i}}{\sqrt{\sigma_{B_i}^2 + \epsilon}} \; ; \; z_i^{(out)} = \gamma_i \hat{z}_i + \beta_i \qquad (1)$$

In this context, $\hat{z}_i$ represents the normalized i-th feature map, $z_i^{(out)}$ denotes the i-th output feature map, μ_{B_i} represents the mean of the i-th feature map over the batch B, γ_i is the standard deviation of the i-th output channel $z_i^{(out)}$ and β_i denotes the mean of the i-th output channel $z_i^{(out)}$. The term γ_i^2 controls the variance of the i-th output channel $z_i^{(out)}$ and $\in$ is a small positive number. Neglecting the effect of activations, the i-th input channel of l-th convolution layer is has a variance of $\gamma_{\ell-1,i}^2$. For the entire ResNet, $W \equiv \{W_l\}_{\{1,2,\ldots,L\}}$ denotes the set of all convolution parameters, and $\gamma = \{\gamma_{l,i}, \beta_{l,i}\}_{l,i}$ represents the parameters of the batch normalization layers. Thus, the contribution of the i-th input to the variance of the j-th output in the l-th convolution layer is described by Eq. (2):

$$\gamma_i^2 = \gamma_{\ell-1,i}^2 \left\| W_{\ell,i,j} \right\|_2^2 \qquad (2)$$

Figure 2(a) illustrates the influence of each input channel on the variance of the output channels. When all outputs are considered simultaneously, the importance criteria (I_c) are established in Eq. (3).

$$I_c = \gamma_{\ell-1,i}^2 \sum_{j=1}^{n_\ell} \left\| W_{\ell,i,j} \right\|_2^2 \qquad (3)$$

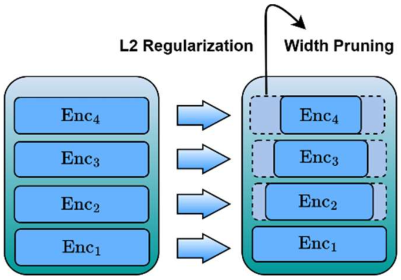
The summation can simply be modified with a scalar that it sums as one: $\sum_{j=1}^{n_\ell} \left\| W_{\ell,i,j} \right\|_2^2 = 1$. Consequently, the final global importance score is denoted by $\gamma_{(L-1,i)}$, which quantifies the extent to which i-th input channel contributes to the variance of the l-th convolution layer. To achieve the desired pruning ratio (η) over T iterations, the following steps need to be followed.

(a). Train the ResNet backbone on a large dataset, applying appropriate regularization on the batch normalization variance γ_i^2.

(b). Rank the channels according to their global importance score γ_i.

(c). Prune η/T channels based on to their importance score and fine tune the pruned model on a downstream target dataset.

(d). Repeat the process starting from step 2 for T iterations to attain the desired level of sparsity. The ResNet backbone is pruned independently. During the training of the image captioning model, the pruned ResNet is loaded while its weights are frozen. Figure 2(b) displays the final pruned ResNet component.



(a)

(b)

Figure 3. Comparison of the pruned and unpruned encoder of the RTIC. (a). Unpruned encoder, and (b). Pruned encoder.

Targeted width pruning of the Encoder component

The encoder, depicted as a second component of the RTIC in Figure 3. Ko et al. [6] demonstrated that the sparsity of the encoder network significantly influences the output quality of encoder-decoder LMs, whereas the number of encoder layers does not significantly affect inference time. Given that, the encoder-decoder network utilized in our image-captioning model mirrors the same architecture of traditional LMs, similar trends are anticipated. Width pruning is applied to the encoder network, as depicted in Figure 3. Unlike Ref. [6], which employed ℓ_0 regularization by enforcing an equality constraint between the target and current sparsity, we do not specify a target sparsity and instead apply regular weight regularization across all encoder weights. Additionally, ℓ_2 regularization is employed to maintain appropriate gradient flows throughout the model, with λ_1 set at 0.01. Thus, the loss contribution from the encoder (L_{enc}) is expressed by Eq. (4).

$$L_{enc} = \lambda_1 \sum_{\{i,j\}} \left\| {W^2}^{enc}_{i,j} \right\| \tag{4}$$

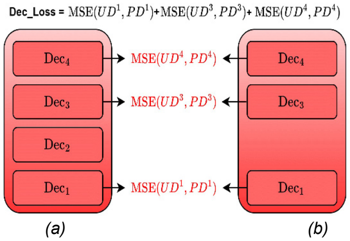


Figure 4. Comparison of pruned and unpruned decoder of the RTIC. (a). Unpruned decoder (UD) and (b). Pruned decoder (PD).

Targeted depth pruning of Decoder component

The decoder, illustrated as the third component of the RTIC in the Figure 4(a). Ko et al. [6] observed that the number of decoder layers was directly proportional to both the inference time and the model size. Accordingly, depth pruning was applied to the decoder network, as depicted in Figure 4(b). For a specified number of selected layers d_s, L_s represents the index of the selected layer, and a decoder subnetwork is generated through uniform sampling, as described in Eq. (5).

$$L_s = \left\{ \left\lfloor \frac{L-1}{d_s-1} \right\rfloor . \ \ell + 1 \mid \ell \in \{ 0 , \ldots , d_s - 1 \} \right\} \tag{5}$$

Ko et al. [6] utilized hidden state distillation to align the hidden states of the decoder subnetwork ($H^{\ell}_{dec,s}$) with those of the original decoder network. The mean square error (MSE), $H_{dec\,,\ell} \left\lfloor \frac{L-1}{d_s-1} . \ell + 1 \right\rfloor$ is the selected state from the original decoder network, and the hidden state distillation loss (L_h^{dec}) is illustrated in Eq. (6). This equation represents the loss contribution from the decoder network, referred to as the pruned RTIC [6].

$$L_h^{dec} = \sum_{\ell \in \{1,2,\ldots,d_s\}} MSE \left(H^{\ell}_{dec,s} \, , H_{dec\,,\ell} \left\lfloor \frac{L-1}{d_s-1} . \ell + 1 \right\rfloor \right) \tag{6}$$

Novel approach of the targeted pruning on Decoder

Our approach to depth pruning in the decoder is depicted in the Figure 5. We adopted a slightly different approach.

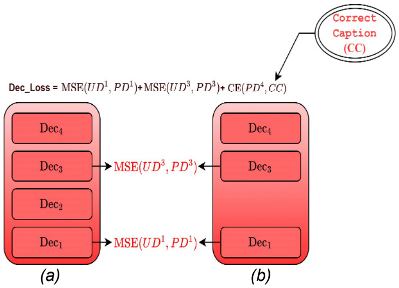


Figure 5. Comparison of pruned and unpruned decoder of the RTIC. (a). Unpruned decoder (UD) and (b). Pruned decoder (PD) with the novel change in decoder final layer.

We matched all hidden states of the decoder subnetwork to those of the original decoder network as shown in the Figure 4, except for the final layer, as stated in Eq. (7).

$$L_h^{dec} = \sum_{\ell=1}^{d_s-1} MSE \left(H^{\ell}_{dec,s} \, , H_{dec\,,\ell} \left\lfloor \frac{L-1}{d_s-1} . \ell + 1 \right\rfloor \right) \tag{7}$$

The output of the last layer is aligned with the true caption of the image, where CC denotes the correct captions, and CE represents the cross-entropy loss as expressed in Eq. (8).

$$L_{dec}^{total} = L_h^{dec} + CE\left(CC \, , H_{dec\,,d_s} \right) \tag{8}$$

This loss described in Eq. (8) as the loss contribution from the decoder network, this is referred to as the proposed pruned-novel change in the decoder. The rationale for this approach is to ensure that the output closely mirrors the original caption. Therefore, rather than aligning the last layer outputs of the pruned and unpruned decoders, we aligned the final output of the pruned decoder with the true caption of the image. Additionally, the proposed decoder pruning method demonstrated superior performance compared to the original decoder pruning technique. The final loss optimized during training is outlined in Eq. (9):

$$L^{total} = \lambda_1 \sum_{\{i,j\}} \left\| {W^2}^{enc}_{i,j} \right\| + \ \lambda_2 L_h^{dec} + \ CE\left(CC \, , H_{dec\,,d_s} \right) \tag{9}$$

where L^{total} represents the total loss optimized for the pruned network during the training, and λ_2 represents the decoder loss contribution coefficient set at 0.01. The final pruning model for the image-captioning model is displayed in Figure 1(b). This model integrates individual network components, including ResNet pruning, encoder pruning, and decoder pruning. The validation of the proposed method is demonstrated in the experiments and results section.

Experiments and Results

We conducted extensive numerical experiments to validate the effectiveness of our proposed method. This section introduces the datasets and evaluation metrics used, specifies the experimental settings, and compares the results of our method with those of state-of-the-art approaches. To validate the proposed method, we conducted the extensive experiments using the Flickr8k dataset [8] [9]. This dataset, curated specifically for image captioning tasks, comprises 8,000 images, each accompanied by five distinct captions

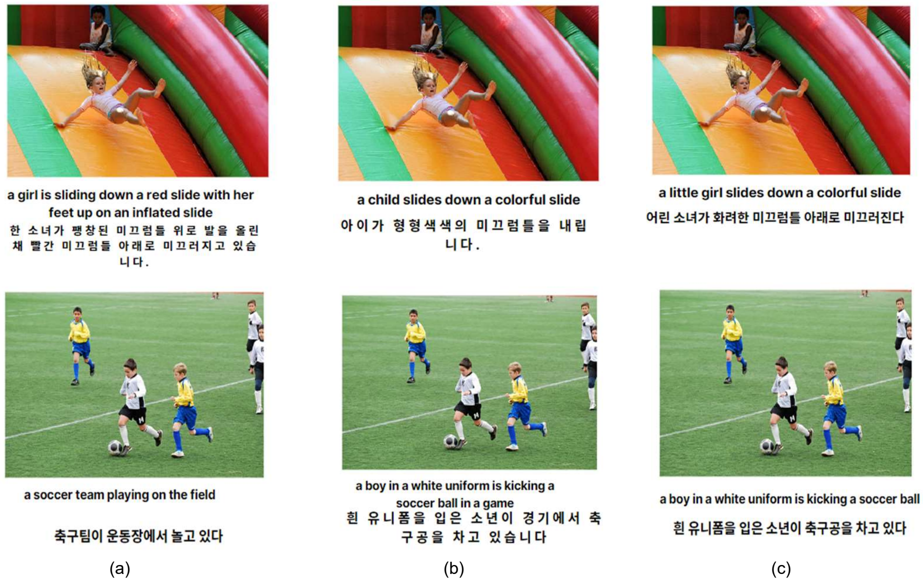

a girl is sliding down a red slide with her feet up on an inflated slide
한 소녀가 팽창된 미끄럼들 위로 발을 올린 채 빨간 미끄럼들 아래로 미끄러지고 있습니다.
a child slides down a colorful slide
아이가 형형색색의 미끄럼틀을 내립니다.
a little girl slides down a colorful slide
어린 소녀가 화려한 미끄럼틀 아래로 미끄러진다
a soccer team playing on the field
축구팀이 운동장에서 놀고 있다
a boy in a white uniform is kicking a soccer ball in a game
흰 유니폼을 입은 소년이 경기에서 축구공을 차고 있습니다
a boy in a white uniform is kicking a soccer ball
흰 유니폼을 입은 소년이 축구공을 차고 있다
(a)
(b)
(c)

Figure 6. *The quantitative results of the image captioning model, (a). Unpruned RTIC Model, (b). Pruned RTIC* [6], *and (c). Proposed Method.*

provided by human annotators, totaling 40,000 captions. The Tiny-ImageNet dataset [4], a subset of the larger ImageNet dataset, is designed for efficient experimentation in machine learning. This dataset includes 100,000 images across 200 classes. Each image is downsized to a 64 × 64 color image. Each class contains 500 training images, 50 validation images, and 50 test images. We used this dataset to train the ResNet-101 pre-trained architecture for feature extraction. To evaluate the performance of our proposed method and the quality of the generated captions, we employed two metrics: recall-oriented understudy for gisting evaluation (ROUGE-1, ROUGE-L) [7], and consensus-based image description evaluation (CIDEr) [1]. We conducted experiments on a single deep learning computer equipped with an NVIDIA RTX A6000 graphics card and CUDA, using PyTorch version 2.3.1.

The system was configured with the following hyperparameters: a pre-trained CNN based on ResNet, an input image size of 256 × 256, a batch size of 32, and an embedding size of 512. The dropout rates were set at 0.02, 0.1, and 0.5. A vocabulary was initially created from words appearing at least five times across the dataset and was tokenized using the Spacy tokenizer. The learning rate was set to 0.0003, referred to as “Karpathy’s learning rate,” for all experiments. The maximum caption length was 5000, Adam was used as the optimizer, and the weight decay for both the encoder and decoder was set at 0.01.

Table 1. Comparative analysis of performance scores between pruned and unpruned models.

Name of the Model	ROUGE-1	ROUGE-L	CIDEr
CNN+LSTM [8]	-	0.2180	0.2890
Merge_RNN [9]	-	0.4430	0.4690
Unpruned RTIC Model	0.3740	0.3478	0.7980
Pruned RTIC [6] Model	0.3104	0.2880	0.4320
Proposed method	0.3110	0.2894	0.4377

Table 2. Size comparisons of pruned and unpruned models.

Component	Size Ratio (Pruned/Unpruned)
ResNet	0.5
Decoder	0.5
Encoder	0.5-0.7

The number of attention heads in both the self-attention and cross-attention layers was fixed at 8. All models are trained with 20 epochs. To evaluate the proposed method, we used the ROUGE-L and CIDEr scores as shown in Table 1. The CNN+LSTM has the lowest reported performance with the modest ROUGE-L, and CIDEr scores, indicating its limited effectiveness in generating captions. The Merge_RNN demonstrates a significant improvement that shows a better alignment with the human-written sentences. The unpruned RTIC model achieves the better CIDEr score and similar performance with ROUGE-L. When pruned, the RTIC model drops a performance in CIDER metric due to prune the unwanted layers to help the model to reduce its complexity. Additionally, our proposed method outperforms the results achieved by relying solely on hidden state distillation for the decoder network.

Table 2 shows the pruning ratios for three components: ResNet, Decoder, and Encoder. Both ResNet and decoder are reduced to half their original size (0.5 ratio), indicating uniform pruning for these components. The encoder’s pruning ratio ranges from 0.5 to 0.7, suggesting some flexibility to retain more capacity in this critical component. This strategy aims to optimize resource usage while preserving the Encoder’s capacity to balance efficiency and quality. A comparison of the sizes of the pruned and unpruned components in terms of Megabytes (MB)is presented in Table 3. The unpruned RTIC model has a size of 346.5 MB that indicates it needs larger memory requirements compared to pruned models. Both the pruned RTIC model and the proposed method are reduced to 240.0 MB, showing identical memory efficiency after pruning. The pruning process results in a size reduction of approximately 31% (from

Table 3. Comparison of model size of pruned and unpruned models in terms of memory (MB).

Name of the Model	Unpruned	Pruned
Unpruned RTIC Model	346.5	-
Pruned RTIC [6] Model	-	240.0
Proposed Method	-	240.0

346.5 MB to 240.0 MB), making these models more lightweight. Despite the significant size reduction, the proposed method achieves better performance over the pruned RTIC model, demonstrating effective optimization. This emphasizes that pruning strategies can significantly reduce the model size without sacrificing performance and can even improve overall performance. Furthermore, the qualitative results are presented in Figure 6. Figure 6(a) shows the results of the unpruned RTIC model, Figure 6(b) demonstrates the predicted captions of the pruned RTIC model, and Figure 6(c) illustrates the predicted image captions generated by the proposed method. The model has a few false cases, as shown in Figure 7. In the first image, the model predicts: "A girl in a purple dress is laying on a red carpeted floor" instead of "A young girl climbs a rock wall in a purple dress." In the second image, the model predicts: "Two people are hiking up a steep grassy hill" instead of "A hiker climbing a rocky hill with fog surrounding him." In both cases, the model's predictions are accurate in general but fail to correctly identify the color or the number of people.

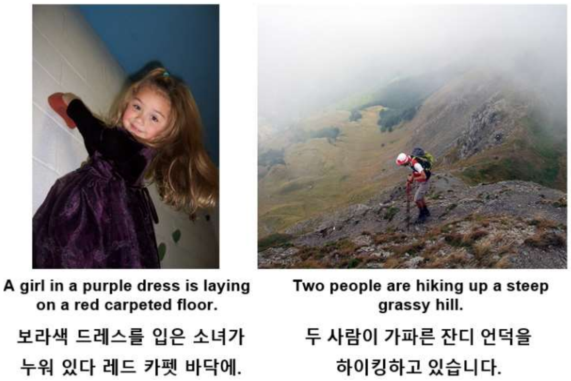

A girl in a purple dress is laying on a red carpeted floor.
보라색 드레스를 입은 소녀가 누워 있다 레드 카펫 바닥에.
Two people are hiking up a steep grassy hill.
두 사람이 가파른 잔디 언덕을 하이킹하고 있습니다.

Figure 7. *The wrong prediction cases of the image captioning model.*

Conclusion

In conclusion, the advancement of deep learning models has significantly enhanced image captioning performance, though this improvement often comes at the cost of increased model complexity and computational demands. Contemporary image-captioning models typically comprise a pre-trained CNN encoder, a transformer encoder, and a transformer decoder. Previous efforts in network pruning of these models have not addressed these components individually, thereby limiting their applicability to diverse model architectures. This study introduces specific pruning techniques designed for each part of the captioning model, thereby enhancing their generalizability to models with similar components, such as encoder and decoder networks. Additionally, this study proposed a novel approach in the decoder, which shows considerable promise in enhancing performance. This methodology not only builds upon existing research but also advances the model efficiency, and adaptability in the field of image captioning.

Acknowledgments

This work was supported in part by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003; in part by the Development Project of Industry Academic-Research Platform Collaboration Technology in 2022 under Grant S3311946; and in part by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program under Grant RS-2024-00438409 supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP).

References

[1] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015.

[2] J.H. Tan, C. Chan, and J.J. Chuah, "End-to-End Super mask Pruning: Learning to Prune Image Captioning Models," Pattern Recognition, vol. 122, no. 1, pp. 1-12, 2022.

[3] X. Dai, H. Yin, "Grow and Prune Compact, Fast, and Accurate LSTMs," IEEE Transactions on Computers, vol. 69, no. 3, pp. 441-452, 2020.

[4] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," CS 231N, vol. 7, no. 7, pp. 1-6, 2015.

[5] A. Khetan, and Z. Karnin, "PruneNet: Channel Pruning via Global Importance," 2020, arXiv:2005.11282 [cs. LG].

[6] J. Ko, S. Park, Y. Kim, S. Ahn, D. Chang, E. Ahn, and S. Yun, "NASH: A Simple Unified Framework of Structured Pruning for Accelerating Encoder-Decoder Language Models," in Findings of the Association for Computational Linguistics: EMNLP 2023, Sentosa Gateway, Singapore, 2023.

[7] C. Lin, "ROUGE: A package for automatic evaluation of summaries," in Proceeding of the Workshop Annual Meeting of the Association for Computational Linguistics., vol. 8, 2004, pp. 74–81.

[8] S.T. Santhanalakshmi, and R. Khilar, "Image Captioning Using Deep Learning," Journal of Harbin Engineering University, vol.44, no.7, July 2023, pp.1-16.

[9] M. Tanti, A. Gatt, and K. P. Camilleri, "Where to put the image in an image caption generator," Natural Language Engineering, vol. 24, no. 3, pp. 467-489, May 2018.

Author Biography

Jishu Sen Gupta is currently pursuing an Integrated Dual Degree (B-Tech + M-Tech) in Mathematics and Computing from Indian Institute of Technology (IIT BHU) Varanasi. His primary research interests in computer vision, multi modal research, and diffusion modelling.

Yogendra Rao Musunuri received his B. Tech degree in electronics and communication engineering from the affiliated college (DPREC) of JNTU, Hyderabad, India in 2012, and M.E degree in image processing and computer vision from the BUFS, Busan, South Korea in 2017. Currently, he is a doctoral student, pursuing a major in the Control and Instrumentation Engineering, from Changwon National University, South Korea. His work has focused on computer vision, NLP, and multi modal Generative AI.

Oh-seol Kwon received his B.S. and M.S. degrees in Electrical Engineering & Computer Science from Kyungpook National University, Republic of Korea in 2002 and 2004, respectively and Ph. D. degree in Electronics from the same university in 2008. From 2010 to 2011, he was a senior researcher with the Visual Display Division, Samsung Electronics, Korea. He joined Changwon National University in 2011, and is currently a Professor. He focused on signal processing, network pruning, language, and multi modal Generative AI.