# Deep Neural Models for Image Captioning: An Empirical Comparison

Abhinav Munagala
amunagal@mail.yu.edu; srikrishnamulagala@gmail.com
ORCID: 0009-0003-8141-8969

***Abstract*—Image captioning aims to generate natural-language descriptions for visual content and lies at the intersection of computer vision and natural language processing. In this paper we present a comparative evaluation of three neural-network architectures for image captioning CNN-LSTM, BERT-Transformer (ViT + BERT), and TransCapNet (ResNet-18 + Transformer) benchmarked on the Flickr8k dataset using five-fold cross-validation. Performance is measured with corpus level BLEU-4 (via `sacrebleu`) and ROUGE-L F scores. TransCapNet, which employs a five-model ensemble of ResNet-18 encoders paired with four-layer Transformer decoders, achieves the strongest results (BLEU-4 $= 0.8638$, ROUGE-L F $= 0.9412$), demonstrating that combining deep residual feature extraction with multi-head attention yields nuanced, human-like descriptions. The BERT-Transformer attains moderate scores (BLEU-4 $= 0.5416$, ROUGE-L F $= 0.2902$), while the CNN-LSTM baseline reaches a BLEU-4 of $0.0333$ and ROUGE-L F of $0.6430$. We provide a structured review of the underlying techniques, discuss challenges including object hallucination and contextual understanding, and identify future directions such as vision language pre-training and improved evaluation metrics. All source code is publicly available.[1]**

***Index Terms*—image captioning, deep learning, encoder-decoder, neural networks, vision language models**

## I. Introduction

Image captioning, the automatic generation of textual descriptions for images, is an important problem at the intersection of computer vision and natural language processing [1], [2]. It has a wide array of applications, from assisting visually impaired individuals [3] to analysing medical images [4], [5] and supporting quality control in industry [6]. The field has undergone major advances with the emergence of deep learning techniques [7], [8], which significantly improved caption quality over earlier rule-based or template-matching approaches.

Several comprehensive surveys cover image captioning research up to 2018 [9]. However, given the rapid progress in recent years, an updated study focusing on the latest deep learning methods can provide valuable insights. Key developments include employing convolutional neural networks (CNNs) for image feature extraction [7], [11], recurrent networks such as Long Short-Term Memory (LSTM) units for caption generation [1], [10], and attention mechanisms for focusing on salient image regions [7], [8].

More recent works have explored reinforcement learning for optimising caption level metrics directly [12] and generative adversarial training for improving caption diversity [13]. Pre-trained vision language models such as Flamingo [14] and CLIP [15] have further advanced the state of the art. In this work we evaluate three representative deep learning architectures for whole image captioning CNN-LSTM, BERT-Transformer, and TransCapNet on a common benchmark. Through detailed analysis and performance comparisons, we highlight architectural trade-offs and discuss remaining challenges and future directions. All implementations, training scripts, and evaluation code are released at https://github.com/mvsakrishna/Image_Captioning.

[1]https://github.com/mvsakrishna/Image_Captioning

## II. Background and Techniques

Automatic image captioning is computationally intensive and structurally complex. This section reviews the standard frameworks and building blocks employed by the models evaluated in this paper.

### A. Encoder-Decoder Framework

Image captioning is primarily framed as a sequence-to-sequence problem analogous to machine translation [16]. The dominant approach is the *encoder-decoder* framework: a CNN encoder maps the input image to an intermediate representation, and an RNN or Transformer decoder converts this representation into a word sequence forming the caption. Despite its success, compressing all visual information into a single vector often yields generic captions. Extensions such as attention mechanisms [7], [8] and dense captioning [17] address this limitation.

### B. Region-Based CNNs

Standard CNN-based object detection using a fixed grid is insufficient because objects vary in shape, size, and position. Region-based CNNs (R-CNN) [18] apply selective search to generate approximately 2 000 region proposals and process each through a CNN. **Fast R-CNN** [19] improves efficiency by computing a shared feature map for the full image. **Faster R-CNN** [20] replaces selective search with a learned Region Proposal Network (RPN), enabling near-real-time detection and forming the basis of many bottom-up attention features used in captioning [7].

### C. LSTMs and GRUs

Recurrent neural networks maintain an internal state for processing sequential input but suffer from vanishing gradients

TABLE I
SUMMARY OF EVALUATED ARCHITECTURES

| Model | Framework | Encoder | Decoder |
|---|---|---|---|
| CNN-LSTM | TensorFlow/Keras | InceptionV3 | LSTM |
| BERT-Transformer | HuggingFace+PyTorch | ViT-base | BERT-base |
| TransCapNet | PyTorch | ResNet-18 | 4-layer Tr. |

for long sequences [21]. Long Short-Term Memory networks (LSTMs) mitigate this with an internal gating mechanism that controls information retention across time steps [21]. Gated Recurrent Units (GRUs) use fewer gates, reducing parameter count and enabling faster training [16], with competitive performance on smaller datasets.

### D. Residual Networks

Residual networks (ResNets) address the degradation problem in very deep networks by introducing skip connections that bypass one or more layers [22]. This architecture is widely used in image captioning for extracting discriminative visual features (e.g., ResNet-18, ResNet-50, ResNet-152).

### E. Transformers

The Transformer architecture [23] replaces recurrence with multi-head self-attention, enabling parallel processing and efficient modelling of long range dependencies. Transformers are now the dominant architecture for both text generation and, through the Vision Transformer (ViT) [24], image understanding.

## III. MODELS

We evaluate three architectures of increasing complexity, summarised in Table I. All models are trained and evaluated on the Flickr8k dataset [29].

### A. CNN-LSTM

*1) Image Feature Extraction:* Images are resized to $299 \times 299$ pixels with three colour channels and processed by an InceptionV3 CNN [28] pre-trained on ImageNet. The final pooling layer is removed, yielding a feature vector of dimension $1 \times 2048$ that captures a high-level abstraction of the image content.

*2) Sequence Generation with LSTM:* The image feature vector is projected to a 256-dimensional embedding space and fed into an LSTM decoder with 512 hidden units. At each time step $t$, the LSTM receives the embedding of the previously generated word and the image context, and outputs a hidden state from which the next-word probability distribution is computed:

$$p(w_t \mid w_{1:t-1}, \mathbf{v}) = \mathrm{Softmax}\big(\mathbf{W}_o\, \mathbf{h}_t + \mathbf{b}_o\big), \qquad (1)$$

where $\mathbf{v}$ is the image feature vector, $\mathbf{h}_t$ is the LSTM hidden state, and $\mathbf{W}_o$, $\mathbf{b}_o$ are learned parameters. Generation starts with a $\langle \mathrm{start} \rangle$ token and terminates at the $\langle \mathrm{end} \rangle$ token. Word embeddings of dimension 256, parameterised by $\mathbf{W}_{\mathrm{emb}}$, capture semantic relationships between vocabulary items. Tokens

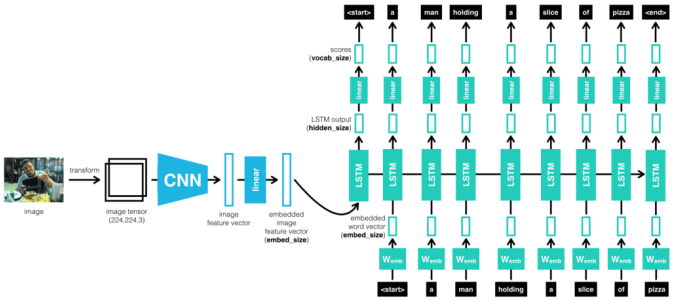


Fig. 1. CNN-LSTM architecture for image captioning.

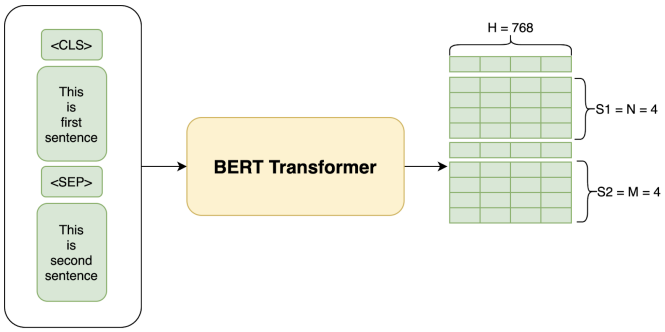


Fig. 2. BERT-Transformer (ViT + BERT) architecture and sample output.

appearing fewer than five times in the training set are mapped to an $\langle \mathrm{unk} \rangle$ token. The model is trained with teacher forcing using cross-entropy loss for 20 epochs with a batch size of 64.

### B. BERT-Transformer

*1) Model Architecture:* This model pairs a Vision Transformer (ViT-base-patch16) [24] encoder with a BERT-base decoder [25] using the HuggingFace `VisionEncoderDecoderModel` framework [26]. ViT divides the input image into fixed-size $16 \times 16$ patches, treats each patch as a sequential token, and processes the resulting sequence using self-attention. This allows the encoder to capture long-range dependencies between distant image regions without the spatial locality constraints of CNNs.

*2) Decoder and Tokenisation:* The BERT-based decoder processes tokenised text sequences produced by the `bert-base-uncased` tokeniser, capturing contextual linkages within and across tokens. The key configuration parameters vocabulary size, decoder start token ID, and pad token ID are aligned between the encoder output and decoder expectations.

*3) Training Configuration:* The model is fine-tuned on Flickr8k for 20 epochs with a batch size of 32 using the AdamW optimiser [27] at a learning rate of $5 \times 10^{-5}$ and a linear learning rate scheduler. Training is conducted on a single GPU.

### C. TransCapNet (ResNet-18 + Transformer)

*1) Image Feature Extraction:* Visual features are extracted from input images using ResNet-18 [22], a pre-trained 18-layer

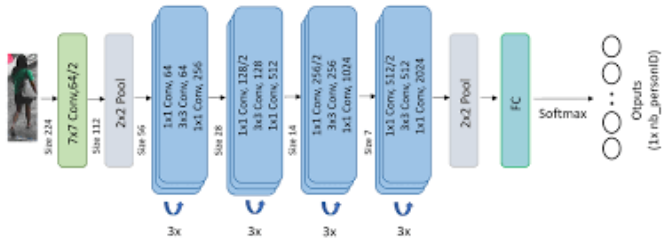


Fig. 3. TransCapNet (ResNet-18 + Transformer) processing pipeline.


Fig. 4. Sample TransCapNet output showing the predicted caption alongside ground-truth reference captions.

residual network. The final classification head is removed, producing a 512-dimensional feature vector for each image.

*2) Positional Encoding and Decoder:* Caption tokens are embedded into 512-dimensional vectors and combined with sinusoidal positional encodings [23], which encode each token's absolute position in the sequence. The Transformer decoder attends jointly to the image feature vector and the previously generated tokens via cross-attention and masked self-attention, respectively. A final linear projection maps the hidden states of the decoder to logits of vocabulary-size, from which the next token is selected auto regressively until the $\langle \text{end} \rangle$ token is produced.

*3) Architecture and Training Details:* The decoder comprises four Transformer layers, each with eight attention heads and a hidden dimension of 512. Tokens appearing fewer than three times are excluded from the vocabulary. The model is implemented in PyTorch and trained with the Adam optimiser (learning rate $10^{-5}$) and a `ReduceLR OnPlateau` scheduler for 30 epochs, minimising cross-entropy loss. The following optimisations are applied:

- *Teacher forcing:* Ground-truth tokens from the previous step are fed as decoder input during training, accelerating convergence.
- *Beam search:* A beam width of 5 is used at inference time to balance caption quality and computational cost.
- *Model ensembling:* Five identically structured models are trained with different random seeds and their logits are averaged at inference, reducing variance and improving robustness.

TABLE II
KEY HYPERPARAMETERS

| Parameter | CNN-LSTM | BERT-Tr. | TransCapNet |
|---|---|---|---|
| Epochs | 20 | 20 | 30 |
| Batch size | 64 | 32 | 64 |
| Learning rate | – | $5\times10^{-5}$ | $10^{-5}$ |
| Optimiser | Adam | AdamW | Adam |
| Embed. dim. | 256 | 768 | 512 |
| Decoder units | 512 (LSTM) | 12 layers | 4 layers |
| Attn. heads | – | 12 | 8 |
| Beam width | 1 (greedy) | 1 (greedy) | 5 |
| Ensemble size | 1 | 1 | 5 |
| Min. token freq. | 5 | – | 3 |

## IV. EXPERIMENTAL SETUP

### A. Dataset

All three models are trained and evaluated on the Flickr8k dataset [29], which contains 8 000 images each paired with five human-written reference captions. The dataset provides a diverse set of everyday scenes and activities, making it a widely used benchmark for image captioning research. Five-fold cross-validation is employed to obtain robust performance estimates.

### B. Evaluation Metrics

Performance is measured using two widely adopted automatic metrics:

- **BLEU-4** [30]: Measures $n$-gram precision (up to 4-grams) between generated and reference captions, with a brevity penalty for overly short outputs. Corpus-level scores are computed using the `sacrebleu` library [31], yielding values in the $[0, 1]$ range.
- **ROUGE-L** [32]: Computes the F-measure based on the longest common subsequence between generated and reference texts, rewarding both recall and fluency.

No post-processing layer (e.g., re-ranking or grammar correction) is applied to the generated captions.

### C. Implementation Details

Table II summarises the key hyperparameters for each model. All experiments are conducted on a single NVIDIA GPU. The CNN-LSTM model is implemented in TensorFlow/Keras, while the BERT-Transformer and TransCapNet are implemented in PyTorch. Pre-trained encoder weights (InceptionV3, ViT-base, ResNet-18) are obtained from their respective standard repositories and fine-tuned during training.

## V. RESULTS AND DISCUSSION

Table III presents the comparative results of the three models.

**TransCapNet** achieves the strongest performance with a BLEU-4 score of 0.8638 and a ROUGE-L F score of 0.9412 after 30 epochs, indicating high fluency and strong alignment with reference captions. The five-model ensemble and

TABLE III
FIVE-FOLD CROSS-VALIDATION RESULTS (NO POST-PROCESSING)

| Model | Epochs | BLEU-4 | ROUGE-L F |
|---|---|---|---|
| TransCapNet (ensemble) | 30 | **0.8638** | **0.9412** |
| BERT-Transformer | 20 | 0.5416 | 0.2902 |
| CNN-LSTM | 20 | 0.0333 | 0.6430 |

beam search decoding contribute significantly: averaging logits across models trained with different random seeds reduces prediction variance, while the beam width of 5 allows the decoder to explore a richer hypothesis space.

The **BERT-Transformer** achieves a BLEU-4 score of 0.5416 and a ROUGE-L F score of 0.2902 after 20 epochs. The low ROUGE-L F relative to BLEU suggests that the generated captions share $n$-gram overlap with references but diverge in sequential structure. The model's performance is likely constrained by the relatively small Flickr8k training set: ViT and BERT are heavily parameterised models that benefit from larger scale fine-tuning data. Additional epochs, data augmentation, or a larger dataset (e.g., MS-COCO) are expected to improve results.

The **CNN-LSTM** achieves a BLEU-4 of 0.0333 and a ROUGE-L F of 0.6430. The low BLEU-4 reflects poor 4-gram precision the model tends to produce short, generic captions that rarely match exact reference phrasing at the four word level. However, its ROUGE-L F of 0.6430 shows reasonable recall at the subsequence level, indicating that the model captures salient content words even when phrase level precision is lacking. This pattern is consistent with the known limitations of LSTM decoders without attention: they struggle to produce diverse, detailed descriptions but can still identify the primary subject of an image.

### A. Analysis

The results highlight several architectural trade-offs:

- *Attention is critical.* Both Transformer-based models significantly outperform the attention-free CNN-LSTM on BLEU-4, confirming that attending to specific image regions or patches improves $n$-gram precision.
- *Ensembling adds robustness.* TransCapNet's five-model ensemble provides a substantial advantage over the single-model BERT-Transformer, suggesting that prediction averaging is an effective strategy for small datasets.
- *Model capacity vs. data size.* The BERT-Transformer's large parameter count ($\sim$200M) is a liability on Flickr8k's 8 000 images. TransCapNet ($\sim$30M parameters) is better matched to the data scale.

## VI. CONCLUSION

This paper has presented a comparative evaluation of three neural network architectures CNN-LSTM, BERT-Transformer, and TransCapNet for automatic image captioning on the Flickr8k dataset. TransCapNet emerged as the strongest performer, demonstrating that combining ResNet-based feature extraction with a Transformer decoder and model ensembling yields high-quality, human-like captions. The BERT-Transformer shows promise but is constrained by the limited training data. The CNN-LSTM provides a competitive ROUGE-L baseline but lacks the $n$-gram precision of attention-based models.

Future research should focus on: (i) evaluating on larger-scale datasets such as MS-COCO to better leverage high-capacity models; (ii) incorporating vision language pre-training (e.g., CLIP [15]) to enhance both encoder and decoder representations; (iii) integrating attention mechanisms into the CNN-LSTM pipeline; and (iv) developing evaluation metrics that better correlate with human judgement, such as CIDEr [33] and CLIPScore [34]. The progress demonstrated in this work has broad implications for accessibility, multimedia retrieval, and human computer interaction.

### Code Availability

All source code, training scripts, and evaluation pipelines are publicly available at https://github.com/mvsakrishna/Image_Captioning.

## REFERENCES


[1] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3156–3164.

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 382–398.

[3] D. Gurari *et al.*, "Captioning images taken by people who are blind," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, UK, Aug. 2020, pp. 417–434.

[4] H. Ayesha *et al.*, "Automatic medical image interpretation: State of the art and future directions," *Pattern Recognit.*, vol. 114, p. 107856, Jun. 2021.

[5] M. M. A. Monshi, J. Poon, and V. Chung, "Deep learning in generating radiology reports: A survey," *Artif. Intell. Med.*, vol. 106, p. 101878, Jun. 2020.

[6] R. Al Sobbahi and J. Tekli, "Comparing deep learning models for low-light natural scene image enhancement and their impact on object detection and classification," *Signal Process.: Image Commun.*, vol. 109, p. 116848, Nov. 2022.

[7] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 6077–6086.

[8] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 2048–2057.

[9] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–36, Feb. 2019.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015.

[11] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 5561–5570.

[12] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7008–7024.

[13] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional GAN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2970–2979.

[14] J.-B. Alayrac *et al.*, "Flamingo: A visual language model for few-shot learning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 23716–23736.

[15] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, Virtual, Jul. 2021, pp. 8748–8763.

[16] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[17] J. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: Fully convolutional localization networks for dense captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4565–4574.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, 2015, pp. 91–99.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[23] A. Vaswani *et al.*, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[24] A. Dosovitskiy *et al.*, "An image is worth $16\times16$ words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Virtual, May 2021.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. (NAACL-HLT)*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[26] Hugging Face, "VisionEncoderDecoderModel," 2020. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/vision-encoder-decoder

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.

[29] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, Aug. 2013.

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Philadelphia, PA, USA, Jul. 2002, pp. 311–318.

[31] M. Post, "A call for clarity in reporting BLEU scores," in *Proc. 3rd Conf. Mach. Transl. (WMT)*, Brussels, Belgium, Oct. 2018, pp. 186–191.

[32] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, Jul. 2004, pp. 74–81.

[33] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4566–4575.

[34] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIPScore: A reference-free evaluation metric for image captioning," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Virtual, Nov. 2021, pp. 7514–7528.