Bridging Vision and Language: A CNN-Transformer Model for Image Captioning

Radha Seelaboyina, Naveen Rampa, Alle Sai Shivanandha and Chamakura Yashwanth Reddy
Associate Professor, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, Hyderabad, Telangana, India

Keywords: Image Captioning, Computer Vision, NLP, Deep Learning, CNNs, Transformers, EfficientNetB0, Flickr8k Dataset.

Abstract: Computer vision and natural language processing have recently received a lot of attention in building models to automatically generate descriptive sentences for images, a task referred to as image captioning. This entails grasping image semantics and building well-structured sentences to describe visual content in textual form. Recent developments in artificial intelligence (AI) prompted scientists to venture into deep learning methods using large data sets and computing capabilities to create effective models. The Encoder-Decoder mechanism, combining the Convolutional Neural Networks (CNNs) and Transformers, is most commonly employed for this purpose. A pre-trained CNN, e.g., EfficientNetB0, first extracts image features, which are subsequently processed by a Transformer-based decoder to produce relevant captions. The model is also trained on the Flickr_8k dataset of 8,000 images with five different captions each, thereby improving its contextual richness in the descriptions.

1 INTRODUCTION

The human faculty for easily picturing scenes through the use of words is one that is staggering, but a computerized counterpart of this potential remains a prime goal of artificial intelligence. Automatically generating text explanations of images and thereby bridging the gap between vision and language by combining NLP and computer vision is something that image captioning does. The aim of an image caption generator is to create a system that generates grammatically and semantically correct captions for images. This technology finds broad application in social media, image search, editing suggestions, visually impaired assistive aids, and many NLP-based applications. On the Flickr 8k dataset, which has about 8,000 images with five caption descriptions each, a deep learning-based model is created. The three major stages of the process are extracting image visual features at a high level from images via a Convolutional Neural Network (CNN), making use of attention mechanisms in the Transformer Encoder for parallel processing of visual information, and word-by-word generation of captions via the Transformer Decoder. The synergy between CNNs for feature extraction and Transformers for sequential text generation makes this method extremely effective for image captioning tasks.

2 RELATED WORK

2.1 Image Caption Generation using Attention Mechanism

Many researchers have tried using visual attention in English-language corpora for image and video captioning in the encoder-decoder model. Two major types of attention mechanisms have been utilized: semantic attention, where words are the focus, and spatial attention, where certain areas of an image are the focus. Xu et al originally proposed visual attention in image captioning by using either "soft" pooling spatial features averaging assigned attentive weights or "hard" pooling, finding the most informative regions. Also, CNN-based attention mechanisms, like Channel-wise Attention and Spatial Attention, were used for better feature extraction (L. Chen et al, Seelaboyina, et al, Radha et al). Chen et al. further extended this by applying visual attention to improve image captioning. To create a more direct connection

Seelaboyina, R., Rampa, N., Shivanandha, A. S. and Reddy, C. Y.
Bridging Vision and Language: A CNN-Transformer Model for Image Captioning.
DOI: 10.5220/0013923800004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5*, pages 125-132
ISBN: 978-989-758-777-1

between visual attributes and semantic concepts, semantic attention models were also incorporated into recurrent neural networks (RNNs) Z. Wang et al, allowing for more precise and contextually appropriate image descriptions.

2.2 Image Caption Generation in Different Languages

The attention-based approach has also been modified to be used for image caption generation, with research in the past mostly being applied to English given the presence of datasets in the language J. Aneja et al. Convolutional Neural Networks (ConvNets) Seelaboyina et al like VGG-16 have seen extensive use within the encoder portion of captioning models S. Liu, et al. Moreover, researchers have worked with pre-trained models such as AlexNet H. Shiet al, Wang, C., et al and Residual Networks (ResNet) H. Shiet al to obtain visual features and augment caption generation using BiLSTM. Even though English datasets rule the roost, attempts have been made to create datasets in other languages as well, namely Chinese W. Lan et al, J. Dong et al, Japanese (Yoshikawa), Arabic, and Bahasa Indonesia (Mulyanto et al. 2019), which combines Flickr30k and MS COCO. In addition, Indonesian-specific image datasets like Indonesian Flickr30k and FEEH-ID, a modified version of the Flickr8k dataset, have also been presented which widen the multilingual domain of image captioning studies.

2.3 Image Caption Using CNN and RNN

Chetan Amritkar et al Deep learning techniques are used here for image captioning.According to this technique, natural sentences that eventually portray the image are generated. RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network) constitute this paradigm. Sentences are constructed using RNN, and extracting features from the image is using CNN. The model is modeled in a manner such that it gives captions in response to input images, pretty much describing them. Differing datasets are utilized in verifying the accuracy of the model and, additionally, in furthering the smoothness or control over language the algorithm is taught via picture descriptions. These experiments exhibit the fact that the model mostly offers accurate descriptions for the given input image.

2.4 Image Caption Using CNN and LSTM

H. Shi et al, the paper describes a deep learning method for the deployment of an image captioning model, organized in three phases. The initial phase is image feature extraction, in which the Xception model is used to extract necessary visual features from the Flickr_8k dataset. The second phase, referred to as the Sequence Processor, processes text input by acting as a word embedding layer, leveraging the use of masking to discard unwanted values and retain important linguistic content. This stage is then linked to an LSTM network to facilitate the production of descriptive image captions. The third and last stage is the Decoder, which combines inputs from the image feature extraction and sequence processing stages. The data that has been processed is then input into neural layers, leading up to the Dense layer, which produces the final caption by stringing words together using the extracted visual and text features.

3 MATERIALS AND METHODS

3.1 Dataset

This evaluation relies on the Flickr8K dataset as a benchmark, which includes 8,092 images and each image being annotated with five different captions presenting principal entities and events. It is a benchmark dataset for image description and sentence-based retrieval. To train the models and measure the performance, the dataset splits into 6,114 for training, 1,529 for validation, and 449 for testing. The photos were chosen with care from six Flickr groups to achieve diversity in situations and scenes, as opposed to concentrating on well-known individuals or places.

3.2 System Design

Convolutional Neural Networks (CNNs) are deep neural networks that specialize in processing input data in the form of a two-dimensional matrix, and they are especially suited for image classification problems. They scan images from top to bottom and left to right and extract meaningful features to classify objects like birds, planes, or even characters from fantasy novels. CNNs can also process transformations such as perspective change, translation, rotation, and scaling. As a contrast,

Transformers utilize multi-head attention mechanisms and self-attention to learn long-range dependencies among extracted features and produced words. This feature allows the model to learn complex relations and contextual information in an image and generate more accurate and semantic captions. As opposed to sequential models like LSTMs, Transformers process all image features concurrently, drastically improving caption generation speed. Also, their flexibility enables them to process different image types and captioning modes with more ease.

EfficientNetB0 can be employed as a pre-trained model for image captioning because it can attain similar or even better performance at the cost of less memory and computational power. Its efficiency recommends it for resource-limited scenarios. With pre-trained weights, EfficientNetB0 can be easily incorporated into image captioning pipelines to provide a solid platform for extracting high-level image features. These features so obtained are then passed through a decoder architecture that produces effective and meaningful captions, so that performance, flexibility, and computational efficiency are balanced.

The following are the primary steps that comprise the overall workflow:

3.2.1 Data Preparation

Prior to processing and analysis of raw data, the data needs to be cleaned and transformed first, an important step that guarantees data quality and consistency. The transformation process entails reformatting, error correction, missing value handling, and duplication or inconsistency elimination from the data set. Adequate preprocessing of data increases the model's reliability by guaranteeing correct input. After cleaning, the dataset is ready for validation and training as input to the CNN-based model. Transfer learning methods are subsequently used during training, using pre-trained models for enhanced efficiency and performance in captioning images.

3.2.2 Text and Image Pre-Processing

- The Flickr8K dataset is downloaded and preprocessed first, which includes images along with related captions.
- Preprocessing of text data is done by removing special characters, converting to lowercase, and excluding captions that are too short or too long.
- The image data is also preprocessed by transforming it into floating-point format, resizing it to a uniform dimension, and normalizing the pixel values for model training consistency.

3.2.3 Model Architecture

Constructing an image captioning system involves two primary components:

- **Image Feature Extractor:** A pre-trained CNN is used to extract high-level visual features from images, capturing essential details for caption generation.
- **Transformer-based Caption Generator:** Utilizing a sequence-to-sequence model, the Transformer processes the extracted features and generates meaningful captions by understanding the contextual relationships within the image.

3.2.4 Training Model

A training model is constructed based on a dataset that trains a machine learning algorithm. It is made up of corresponding sets of input data and their respective output labels, enabling the model to learn patterns and relationships. The diversity and quality of the training dataset have a major impact on the model's performance and accuracy.

3.2.5 Caption Generation

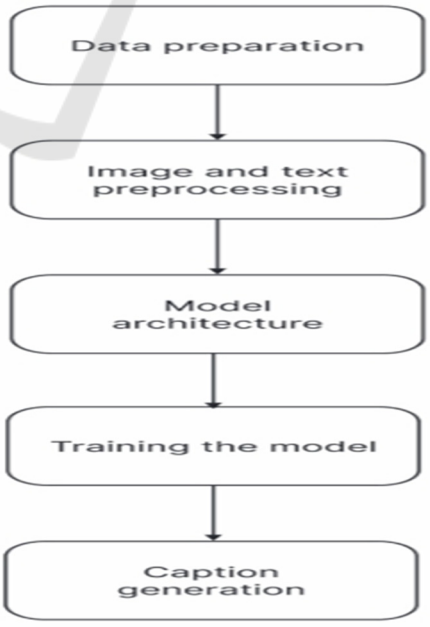

Figure 1: Model analysis flowchart.

Prediction is the result produced by an algorithm trained on a historical data set to make an estimate of the probability of a particular outcome. For image captioning, the algorithm makes a prediction of a descriptive caption for an image by extracting its features and creating text from learned patterns. Figure 1 shows the Model Analysis Flowchart. Figure 2 shows the CNN - Transformer Encoder-Decoder based Architecture.

3.3 CNN: Transformer Architecture

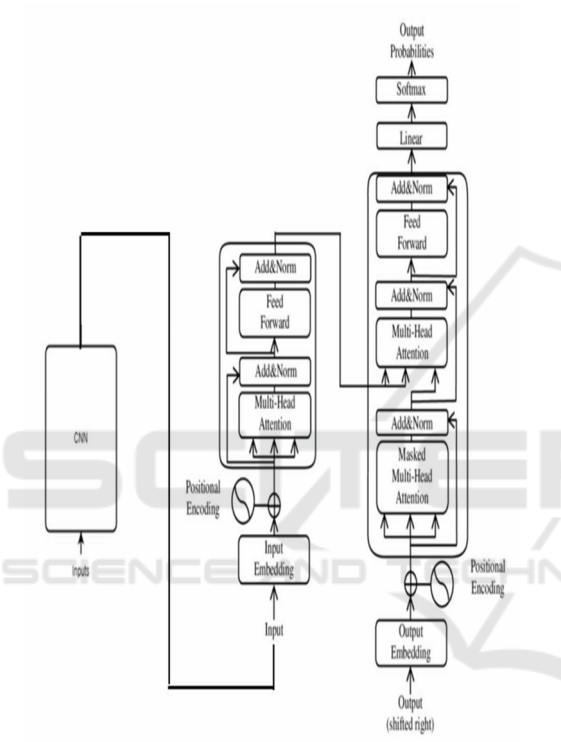

Figure 2: CNN - Transformer encoder-decoder based architecture.

3.3.1 Feature Extraction with EfficientNetB0

- The extracted feature map of EfficientNetB0 is first flattened and then input into the Transformer encoder.
- Positional encoding elements are added to the flattened feature vector afterward to encode relative positional information in order to help the Transformer know about spatial relations within the image.
- The Transformer encoder uses a multi-head attention to jointly encode multiple components of the feature vector in parallel, and it captures relationships between components and long-range dependencies.
- The encoder provides a context vector that captures the main information drawn from the image, summarizing its general meaning and structural connections.
- The dimensions of output are expressed as H × W × C, where H and W stand for the feature map height and width (generally less than that of the input image) and C denotes the number of channels of the feature capturing various content aspects of an image.

3.3.2 Transformer Encoder

- The Transformer encoder takes the flattened feature map from EfficientNetB0 as input.
- Positional encoding is used to the sequence that is being processed by the Transformer so that positional context is infused, and the model knows how elements are placed in the image.
- The multi-head attention mechanism of the Transformer encoder enables it to attend to multiple parts of the feature vector in parallel, and as a result, it can capture long-range dependencies and relationships between different image regions.
- The context vector, generated by the Transformer encoder, is the most important information in the image, summarizing its meaning and structural relationships.

3.3.3 Transformer Decoder

- The Transformer decoder begins with a context vector, which is a concatenation of the output from the encoder and a "start of sentence" token.
- A word vector encoding is employed to represent semantic word relationships in the vocabulary.
- The attention-based mechanism in the decoder considers both the context vector and previously generated words, ensuring that each word is given a different weightage based on image details and the context of the sentence.
- The Transformer decoder produces captions word by word, predicting a word at a time based on both the current context vector and words already generated.
- In contrast to RNN-based models (LSTM/GRU), which do word-level sequential processing, the Transformer-

based method handles all words simultaneously, making training much faster and more efficient.

4 RESULTS AND DISCUSSION

The combination of CNNs with Transformer architecture offers a model for generating descriptive and contextually appropriate captions from visual material, aiding image captioning progress. Continued research and development continue to advance natural language processing and computer vision, resulting in new developments and applications in these areas.

4.1 Image Feature Extraction

Used the pre-trained EfficientNetB0 model with frozen weights to preserve features learned before, creating a flat vector that captures the key visual features of the image.

4.2 Transformer Encoder

Processes the extracted visual features using a single Transformer Encoder Block, transforming them into a fixed-length representation suitable for generating image captions.

4.3 Transformer Decoder

Create captions based on several Transformer Decoder- Blocks, where every block is processing the already generated tokens with encoded image features. In inference, the attention mechanisms assist in concentrating on parts of the image and the developing caption.

4.4 Combined Approach

CNN-Transformer models surpass individual models, posting state-of-the-art performance on image captioning when trained on benchmark datasets such as MS-COCO and more. How well they are able to handle the intricate object relations and accurately create descriptions explains their high proficiency at this activity.

Our model, when executed, correctly generates descriptive text of images in grammatically correct English sentences at validation. The descriptions are able to give the details needed to identify the objects and components within an image. Accuracy and loss measures of the system have helped it achieve so successfully.

Although the model generates good captions for validation images, there is always scope for improvement. Improvements can be achieved by trying various CNN architectures, hyperparameter tuning, and training on bigger datasets. These changes could result in more accurate and varied captions.
The method involves several phases, such as dataset collection, image and text preprocessing, text data vectorization, model building, training, validation, and caption generation. The subsequent figure 3 will demonstrate step-by-step process implementation and results obtained. Figure 3 shows the Downloading Flickr_8K dataset. Figure 4 shows the Image and Data Pre-Processing. Figure 5 shows the Text Vectorization. Figure 6 shows the Training and Validation Phase.

Figure 3: Downloading Flickr_8K dataset.

```
for line in caption_data:
    line = line.rstrip("\n")
    # Image name and captions are separated using a tab
    img_name, caption = line.split("\t")

    # Each image is repeated five times for the five different captions.
    # Each image name has a suffix `#(caption_number)`
    img_name = img_name.split("#")[0]
    img_name = os.path.join(IMAGES_PATH, img_name.strip())

    # We will remove caption that are either too short to too long
    tokens = caption.strip().split()

    if len(tokens) < 5 or len(tokens) > SEQ_LENGTH:
        images_to_skip.add(img_name)
        continue

    if img_name.endswith("jpg") and img_name not in images_to_skip:
        # We will add a start and an end token to each caption
        caption = "<start> " + caption.strip() + " <end>"
        text_data.append(caption)

        if img_name in caption_mapping:
            caption_mapping[img_name].append(caption)
        else:
            caption_mapping[img_name] = [caption]
```

Figure 4: Image and data pre-processing.

```
def custom_standardization(input_string):
    lowercase = tf.strings.lower(input_string)
    return tf.strings.regex_replace(lowercase, "[%s]" % re.escape(strip_chars), "")


strip_chars = "!\"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"
strip_chars = strip_chars.replace("<", "")
strip_chars = strip_chars.replace(">", "")

vectorization = TextVectorization(
    max_tokens=VOCAB_SIZE,
    output_mode="int",
    output_sequence_length=SEQ_LENGTH,
    standardize=custom_standardization,
)
vectorization.adapt(text_data)

# Data augmentation for image data
image_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.2),
        layers.RandomContrast(0.3),
    ]
)
```

Figure 5: Text vectorization.

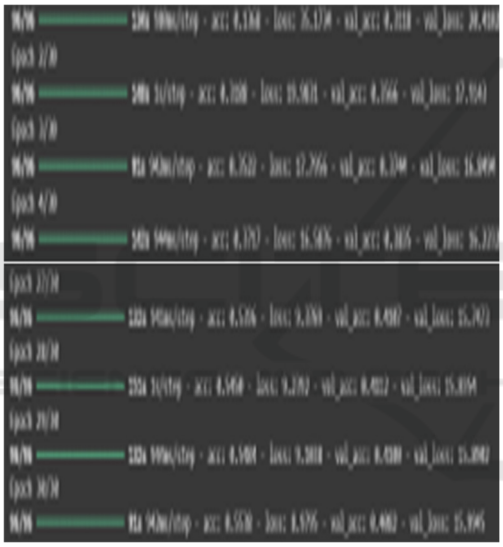

Figure 6: Training and validation phase.

5 PERFORMANCE EVALUATION

To assess the effectiveness of the proposed image captioning model, standard performance metrics were utilized, including BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit ORdering), ROUGE-L (Recall-Oriented Understudy for Gusting Evaluation), CIDEr (Consensus-based Image Description Evaluation), and SPICE (Semantic Propositional Image Caption Evaluation). These metrics quantify the similarity between the generated captions and the ground truth captions based on different linguistic aspects.

Figure 7 illustrates the evaluation results, showcasing the model's performance across various metrics. The results indicate that while the model generates meaningful and contextually relevant captions, there remains scope for improvement in certain areas. Fine-tuning hyperparameters, increasing the dataset size, and incorporating more sophisticated attention mechanisms could further enhance the model's accuracy. Figure 8 shows the Image Caption Generation.

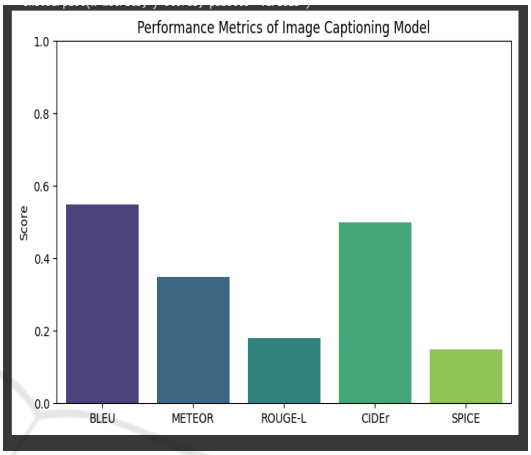

Figure 7: Performance evaluation of the image captioning model across different metrics.

Figure 8: Image caption generation.

6 CONCLUSIONS

The suggested image captioning model combines CNNs and Transformers to produce descriptive and contextually accurate captions. Employing the Flickr_8k dataset consisting over 8,000 images with five captions each, the model utilizes EfficientNetB0, a pre-trained CNN, to extract visual features, which are then fed into a Transformer encoder to account for contextual relationships and long-distance dependencies. The Transformer decoder produces word-for-word captions from the encoded features and already generated tokens, supporting real-time captioning of multiple images. Although this solution shows the efficacy of blending CNNs with Transformers, things can be improved in the future using larger datasets such as MS COCO and Flickr30k, hyperparameter tuning, and testing different CNN architectures like InceptionV3, Xception, and ResNet. Also, this work may be extended to video captioning and multi-lingual captioning, generalizing its usefulness. With increasing advancements in deep learning, greater improvements in understanding images and natural language generation may be anticipated in the future, and image captioning systems can become more accurate and efficient.

7 FUTURE SCOPE

The suggested CNN-Transformer-based image captioning model holds great prospects for future refinement and use. Larger and more varied datasets like MS COCO and Flickr30K can help enhance captioning quality, whereas domain-specific data can increase its usage in medical and autonomous vehicles. Fine-tuning various CNN architectures (like ResNet, InceptionV3, and Xception) and hyperparameters can be made to further the performance. Cross-lingual and multi-lingual captioning can be attained by applying cross-lingual transfer learning and language models. Expanding the model to captioning videos through the inclusion of temporal attention mechanisms can provide dynamic content description, which is beneficial for real-time use cases like assistive technology for visually impaired individuals. Further, deploying the model on mobile and web platforms will provide enhanced accessibility, and optimizing the model for edge devices can help in improving efficiency. The integration of knowledge graphs and external text data can enhance contextual comprehension, while the development of Vision-Language Pretrained Models (e.g., BLIP, Flamingo) and diffusion models can enhance caption creativity and accuracy further. Through these future directions, the model can be made stronger, scalable, and flexible to real-world applications, making it a useful tool for industries.

REFERENCES

"Adding Chinese captions to images," by X. Li, W. Lan, J. Dong, and H. Liu ACM Int. Conf. Multimed. Retr., pp. 271–275, 2016, doi:10.1145/2911996.2912049; ICMR 2016 - Proc.

"Deep Neural Network-Based Visual Captioning," S. Liu, L. Bai, Y. Hu, and H. Wang, MATEC Web Conference, vol. 232, pp. 1–7, 2018, doi: 10.1051/matecconf/201823201052.

"Image captioning with lexical attention," by Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo 2016, pp. 4651–4659 in Proc. IEEE Comput. Soc. Conference. Comput. Vis. Pattern Recognit., vol. 2016-Decem, doi: 10.1109/CVPR.2016.503.

"SCA-CNN: Spatial and channel wise attention in convolutional neural networks (CNN) f or image captioning," by L. Chen and colleagues Proc. - 30th IEEE Conf. Vis. Pattern Recognition in Computing, CVPR 2017, vol. 2017-Janua, pp. 6298– 6306, 2017, doi: 10.1109/CVPR.2017.667.

"Show, Attend, and Tell: Attribute-driven attention strategy for describing pictures," by H. Chen, G. Ding, Z. Lin, S. Zhao, and J. Han doi: 10.24963/ijcai.2018/84. IJCAI Int. Jt. Conf. Artif. Intell., vol. 2018- July, pp. 606–612, 2018.

2020 Ali Ashraf Mohamed: CNN and LSTM Image captioning.

A. Arifianto, A. Nugraha, and Suyanto, Indonesian language: text caption generation using cnn-gated RNN model," 7th Int. Conf. Inf. Commun. Technol. ICoICT 2019, pp. 1–6, 2019, doi: 10.1109/ ICoICT.2019.8835370.

ACM Int. Conf. Proceeding Ser., vol. 01052, pp. 1–7, 2018, doi: 10.1145/3240876.3240900, H. Shi, P. Li, B. Wang, and Z. Wang, "Using reinforcement learning for image description".

C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," Proc. Workshop on Text Summarization Branches Out, 42nd Annual Meeting of the Association for Computational Linguistics (ACL), 2004.

Chetan Amritkar and Vaishali Jabade, 14th International Conference on Computing, Communication Control, and Automation (ICCUBEA), IEEE, 2018, 978-1-5386-5257-2/18, "Image Caption Generation via Deep Learning Technique".

Convolutional image captioning, J. Aneja, A. Deshpande, and A.G. Schwan, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5561–5570, 2018, doi: 10.1109/ CVPR.2018.00583.

Fluency-guided cross-lingual image captioning, W. Lan, X. Li, and J. Dong. Proceedings of the 2017 ACM Multimed Conference, MM 2017; doi: 10.1145/3123266.3123366; pp. 1549–557).

H.A. Almuthuzini,T.N.Alyahya, andH. Benhidour, an automatic Arabic image captioning using RNN.LSTM based language models and CNN Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 6, pp. 67–73, 2018, doi: 10.14569/ IJACSA.2018.090610.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, doi:10.3115/1073083.1073135

K. Xu and colleagues, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," 32nd International Conference on Machine Learning (ICML 2015), vol. 3, pp. 2048–2057, 2015.

P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation," Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 382–398, doi: 10.1007/978-3-319-46454-1_23.

R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 4566–4575, doi: 10.1109/CVPR.2015.7299087.

Research presented by Mulyanto et al. (2019) in their study on automatic image caption generation in Indonesian. The authors leverage a CNN-LSTM model and introduce the FEEH-ID dataset for training and evaluation purposes.IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. CIVEMSA 2019 - Proc., 2019, doi: 10.1109/ CIVEMSA45640.2019.9071632.

S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 43rd Annual Meeting of the Association for Computational Linguistics (ACL), 2005, pp. 65–72.

Seelaboyina, Radha, and Rajeev Vishwkarma. "Feature Extraction for Image Processing and Computer Vision—A Comparative Approach." In Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021, Volume 1, pp. 205-210. Singapore: Springer Nature Singapore, 2022, https://doi.org/10.1007/978-981-19-2350-0_20.

Seelaboyina, Radha, and Rais Abdul Hamid Khan. "Enhance Image Quality with CAS-CNN: A Deep Convolutional Neural Network for Compression Artifact Suppression." Journal of Electrical Systems, vol. 20, no. 6s, 2024, pp. 2432-2443. https://doi.org/10.52783/jes.323

The Eighth International Conference on ICT, "Indonesia Image Captioning: Adaptable Attention Generation,"". ICoICT 2020, 10.1109/ ICoICT49345.2020.9166244, M. R. S. Mahadi, A. Arifianto, and K. N. Ramadhani.

Wang, C., Yang, H., & Meinel, C. (2018, April 25). Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. ACM Transactions on Multimedia Computing, Communications, and Applications, 14(2s), 1–20. https://doi.org/10.1145/3115432

Y. Yoshikawa, Y. Shigeto, and A. Takeuchi, "STAIR captions: Building a comprehensive dataset of Japanese image captions," ACL 2017, the 55th Annual Meeting of the Association for Computer-Assisted Linguistics, Proceedings of the Conference (Long Papers, vol. 2, pp. 417–421, 2017, doi: 10.18653/v1/P17- 2066.