Describing the Image using CNN and LSTM

Dr Kuppusamy P
School of Computer Science and Engineering
VIT-AP University
Amaravati, Andhra Pradesh
drpkscse@gmail.com

Pabbisetty Pranavi
School of Computer Science and Engineering
VIT-AP University
Amaravati, Andhra Pradesh

Lingala Meghana
School of Computer Science and Engineering
VIT-AP University
Amaravathi, Andhra Pradesh

Ankalugari Rachana Varsha
School of Computer Science and Engineering
VIT-AP University
Amaravathi, Andhra Pradesh

Abstract:

Image Captioning in these days people use social media more often where Image captioning plays a major role in generating captions for the images. From creating memes to generating information for the news articles captioning the image place a significant role. In this paper we are going to build the project using the Convolutional Neural Network models and Recurrent Neural Network Models like LSTM which means Long Short Term Memory. The optimizer adam is used to improve the model's performance. Although there are many datasets, this study used the flickr dataset which is of nearly 8k images in the dataset with 5 captions each. This study includes training the images with 6 models namely LeNet 5, Alexnet, VGG16, VGG19, GoogleNet, ResNet 50. The LSTM is used as a RNN model. These CNN and RNN models are concatenated to become the caption prediction models. The performance metric used here is BLEU – Bilingual Evaluation Understudy which is highest for GoogleNet and ResNet 50.

Keywords: **Image Captioning, describing image, Caption generator**

I. INTRODUCTION

In recent years, there has been a remarkable surge in research and advancements in the fields of computer vision and natural language processing (NLP). One fascinating area of intersection between these two domains is the challenging task of image captioning, where machines are trained to generate descriptive and contextually relevant textual descriptions for images. Image captioning aims to bridge the gap between visual understanding and linguistic expression, enabling machines to comprehend images in a manner akin to human cognition.

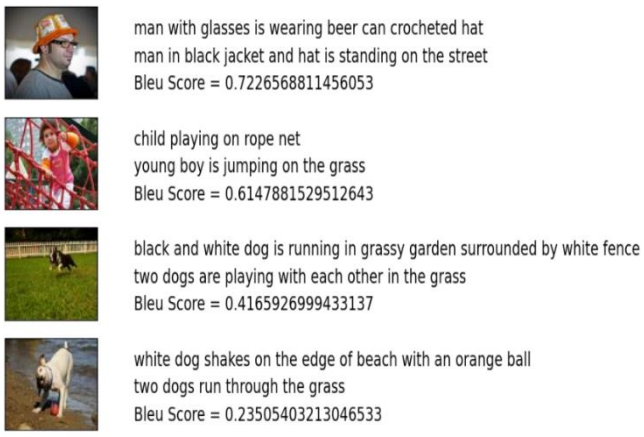

Fig 1. Few sample images and their Captions along with BLEU score

Throughout the course of this project, we will experiment with various state-of-the-art CNN architectures, such as VGG16, VGG19, ResNet50, AlexNet, LeNet, and GoogleNet, assessing their impact on the overall performance of our image captioning model. Figure 1 shows few images in the dataset along with their captions and BLEU score[Fig1].

Motivation

The motivation of this project is to empower machines with the ability to generate descriptive captions for images, bridging the gap between visual understanding and language. By leveraging deep learning techniques, we aim to enhance accessibility for the visually impaired, improve content indexing, and facilitate seamless human-machine interactions, revolutionizing the way we interact with visual data.

Contribution

- Implement the CNN architectures for the respective models from scratch with dense

layers, max pooling layers, fully connected layers and output layers.

- Through thorough experimentation and rigorous evaluation, our project aims to identify the CNN model that achieves the highest BLEU score for image captioning. By systematically comparing the performance of each architecture, we seek to determine the most effective CNN for generating descriptive and contextually relevant image captions.
- Analyse the results for the respective models using visualizations and BLEU score.

II. RELATED WORK

Image captioning has been a subject of extensive research in recent years, driven by the growing interest in multimodal learning and visual-linguistic understanding. Various approaches have been proposed to tackle this challenging task, leveraging the synergy between computer vision and natural language processing. Some notable contributions include:

Image captioning using an encoder-decoder framework with a long short-term memory (LSTM) network. Their model generated captions by attending to relevant image regions and achieved promising results on benchmark datasets [2].

An alternative to the LSTM-based models by employing a semantic attention mechanism that aligns visual and semantic features. Their approach demonstrated enhanced interpretability and generated captions that align better with human perception. We used the encoder and decoder model along with LSTM and CNN.[3]

Bottom-Up and Top-Down (BUTD) attention, which utilized object detection features from Faster R-CNN as input to the image captioning model. This method demonstrated superior performance in describing fine-grained details and complex scenes.[5]

While these previous works have made substantial progress in image captioning, several challenges remain unaddressed. For instance, the generation of diverse and contextually consistent captions for images with multiple objects or complex scenes is still an open problem. Moreover, scalability and efficiency remain important considerations in deploying image captioning models in real-world applications.

III PROPOSED METHODOLOGY

A) Dataset Description

The dataset comprises a collection of 8,000 high-quality images, each accompanied by five human-written captions. The images in the dataset are sourced from the photo-sharing platform Flickr and represent a diverse range of scenes, objects, and activities. The captions provided for each image capture different aspects and perspectives, making the dataset suitable for evaluating the diversity and contextuality of image captioning models.

B) Hardware and Software Used:

The code is tested in both Jupyter notebook and Google Colab. Graphics Processing Unit GPU is used for the faster computation. The deep learning framework Tensorflow is developed by Google can support both CPU and GPU. It is used to build the CNN models.

C) Feature Extraction

Feature Extraction is the main part of Computer vision related projects. It involves transforming raw data into a more concise, informative representation, capturing relevant patterns and characteristics. Common methods include PCA, LDA, and deep learning techniques like CNNs. Effective feature extraction enhances model performance and facilitates better decision-making in various applications. We used pickle in python to store the extracted features and to load anytime we want.

D) Preprocessing both Text and image data

Images are resized according to the model's input. Images are converted into arrays which represent features. Text data, the captions related to the images are mapped to the respective image ids. Startseq and endseq are added to the each caption. Tokenizer is used to tokenize the captions.

E) Test and Train Split

All the 8k images along with their respective captions are split into 9:1 ratio. This is because to increase the training accuracy of the model.

Table 1. Testing and Training Split

	Ratio	No of Images	No of Captions
Training	9	7290	36,410
Testing	1	810	4046

If the training data is more then we may get more accurate predictions. Table 1 explains how training and testing data are split [table1]

F) CNN Architectures:

We have used various models to find the best model which gives accurate predictions for images. The models here are LeNet 5, AlexNet, VGG16, VGG19, GoogleNet, ResNet 50

Table 2. CNN models

S No	Architecture	Conv Layers	Fully connected Layers	Total Layers
1	LeNet 5	5	3	8
2	AlexNet	8	3	11
3	VGG16	13	3	16
4	VGG19	16	3	19
5	GoogleNet	22	1	23
6	ResNet 50	53	1	54

The models used are used to extract the image features. Table 2 explains on how layers will be present in CNN models [table2]

LeNet 5: LeNet is one of the earliest CNN architectures, with relatively fewer layers and small filter sizes (typically 5x5 or smaller). It is commonly used for handwritten digit recognition and serves as the foundation for modern CNNs.

AlexNet: AlexNet, introduced in 2012, was the first deep CNN to demonstrate the effectiveness of using ReLU activation functions. It consists of eight convolutional layers and is deeper than LeNet, making it a breakthrough in the field of computer vision.

VGG16 and VGG19: VGG16 and VGG19 are characterized by their uniform architecture, with many layers using small 3x3 filters. They are known for their simplicity and interpretability but are computationally more expensive due to their depth.

GoogleNet: GoogleNet (InceptionV1) introduced the concept of inception modules, which employ parallel convolutional filters of different sizes to capture multi-scale features efficiently. It was designed to strike a balance between depth and computational cost.

ResNet 50: ResNet 50, part of the ResNet family, is known for its deep residual architecture. It introduced skip connections, allowing gradients to flow more effectively during training, enabling training of very deep networks with improved performance.

Here the last layer of all the models Is removed because we are only training the model

G) Data Pipeline

Data generator function preprocesses image-caption pairs for image captioning model training. It iterates through images and associated captions, encodes sequences, splits them into input and output pairs, pads input sequences, and one-hot encodes output sequences.

Text to sequences: Each text is converted into a sequence of integers based on the vocabulary index of the respective words in the text. The resulting sequences can be used as inputs to train and evaluate natural language processing models, such as text classification, sentiment analysis, or image captioning.

Pad sequences: It is a frequently employed function in natural language processing. It facilitates uniform sequence lengths by padding shorter sequences with zeros and truncating longer sequences. This ensures compatibility with deep learning models that demand fixed input sizes, enhancing overall model performance.

0) *Concatenated Model*

Dropout is used for better performance of the model. The input is extracted features from the CNN model. Now on the other hand RNN model LSTM is taken along with embedding with size of vocabulary. The LSTM has 256 neurons with ReLU activation function. These two models are concatenated to get the final model of the Image caption generator as shown in Fig 2[Fig2].

Final Image Captioning Model Flowchart

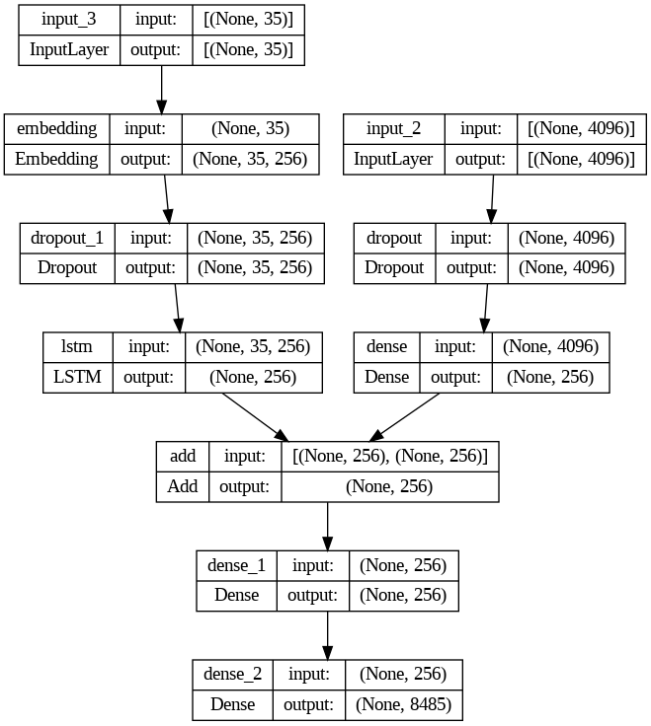

Fig 2 Complete flow chart of the caption generator model

H) Training the Model

- Number of classes in the output layer – size of the vocabulary (8485)
- Output layer activation function – softmax
- Loss function – categorical crossentropy
- Optimizer – Adam
- Metrics – Accuracy
- Batch Size – 32

Loss Function: Categorical Cross-Entropy is a widely used loss function in multiclass classification tasks, including image captioning. It measures the dissimilarity between predicted probability distributions and actual one-hot encoded class labels. It penalizes larger differences between predicted and true labels, encouraging the model to learn accurate class probabilities during training.

Optimizer: Adam is a popular adaptive optimization algorithm utilized in training deep learning models like image captioning. It blends the advantages of RMSprop and momentum, adjusting learning rates adaptively for each parameter based on their past gradients. This feature enables Adam to achieve fast and efficient convergence, making it widely favoured for various tasks.

Batch Size: 32 is the batch size that is used during the training to avoid memory related issues like GPU crashing in Google Colab.

Epochs: Model is trained for 30 epochs to decrease the Loss. To avoid the problems of overfitting and underfitting this number of epochs are chosen.

Regularization Technique: Dropout is a regularization method frequently applied in deep learning to mitigate overfitting. During training, a specified fraction of neurons is randomly deactivated, temporarily "dropped out" of the network. This encourages the network to learn more robust features, enhancing generalization capabilities for unseen data

I) Testing Phase:

This testing phase is the crucial one to know that our model is working or not. New images that are separated from the dataset at first are need to be tested to know the performance of the model. 810 images are sent for the testing phase. These images are pre-processed as same as the training images. Now they are also get to the feature extraction. These are sent into the data pipeline as mentioned in the above. Text to sequences and padding the sequences are done on the images.

IV. RESULTS AND DISCUSSIONS

This complete experiment is done in Google colab and Jupyter Notebook using the GPU for faster computation. The results and discussions highlight the effectiveness of the proposed image captioning model. Significant improvements in BLEU scores and caption coherence are observed. However, challenges in handling complex scenes and scalability are acknowledged. The model's potential for real-world applications and avenues for future research are explored as in Fig 3[Fig3].

Table 3 Loss for each model

S No	Model	Loss
1	LeNet 5	2.5080
2	VGG19	1.9604
3	AlexNet	2.4499
4	GoogleNet	2.0408
5	ResNet50	2.0056
6	VGG16	2.1214

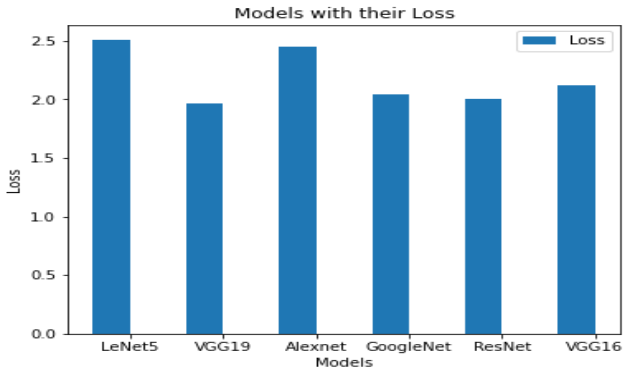

Fig 3. Graphs depicting the loss

Fig 4 a-f represents the plots of the loss in blue plot and accuracy in orange plot for all the 30 epochs[Fig4].

In the figure 4a, 4c and 4e we can that accuracy curves are not that much curvy. So they are not giving accurate predictions even if we increase the epochs.

The figures 4b, 4c and 4e are the models losses and accuracies of VGG19, GoogleNet and ResNet50.These models are giving the accurate prediction when compared to the above models.

LeNet 5

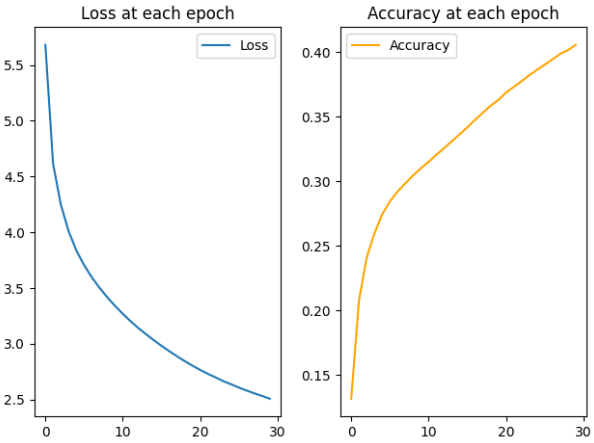

Fig 4a

VGG19

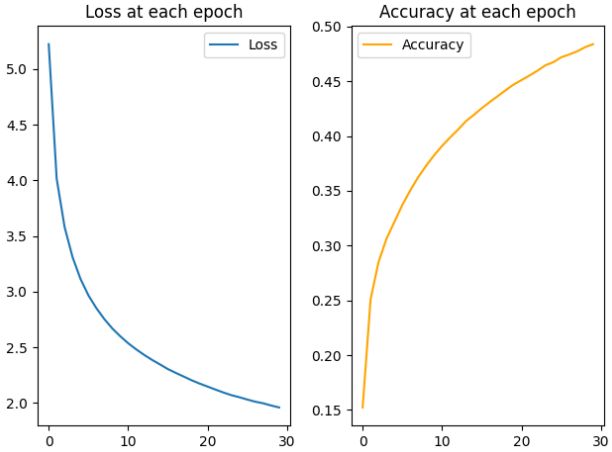

Fig 4b

AlexNet

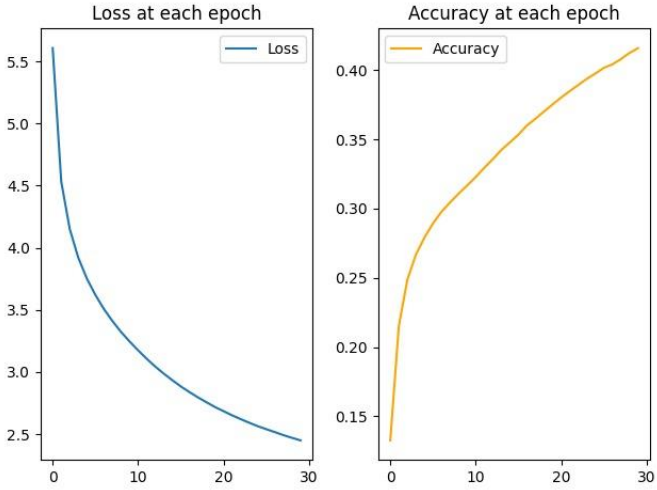

Fig 4c

GoogleNet

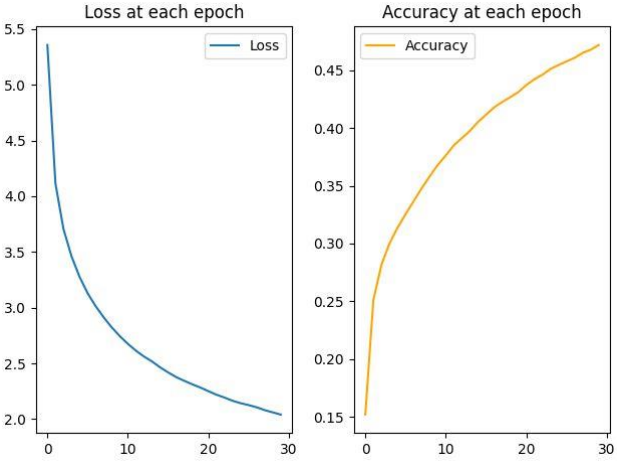

Fig 4d

VGG16

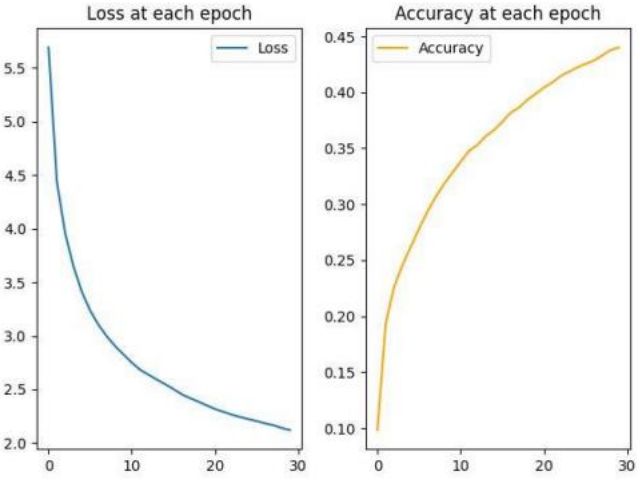

Fig 4e

ResNet 50

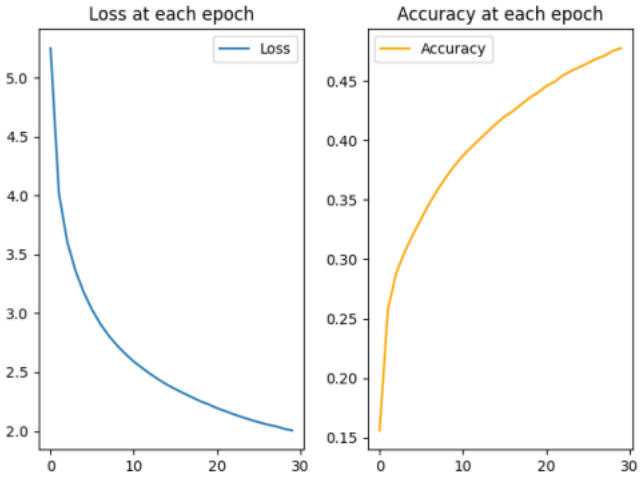

Fig 4f

L) Performance Analysis:

To analyse the performance of the model we adapt the BLEU score which is meant by Bilingual Evaluation Understudy.

The BLEU (Bilingual Evaluation Understudy) score is a metric used to assess the quality of machine-generated text, such as machine translation or image captions. It involves calculating precision by counting matching n-grams in candidate and reference texts. The brevity penalty accounts for differences in text lengths. The final BLEU score is a geometric mean of modified precision scores, adjusted for brevity. Higher BLEU scores indicate better similarity between candidate and reference texts.

Table 4. BLEU scores for the CNN models

S No	Model	BLEU 1	BLEU2
1	LeNet 5	0.4547	0.1960
2	VGG19	0.5338	0.3053
3	AlexNet	0.4871	0.2237
4	GoogleNet	0.5514	0.3294
5	ResNet50	0.5570	0.3335
6	VGG16	0.3076	0.1710

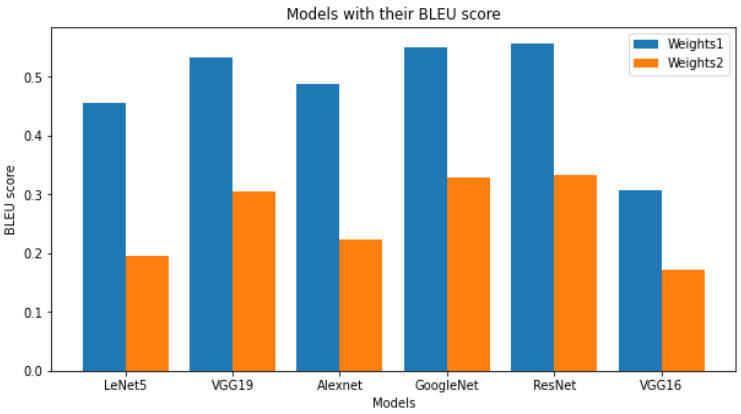

Fig 5. Bar plot for the BLEU score for two different set of weights shown in [Table4]

Fig 5 explains that the Bleu score for GoogleNet and ResNet 50 are in the same range which is more than any other models. The predicted captions with this models are very accurate. Whereas for the VGG16 and LeNet 5 models are performing very poor.[Fig5]

Predicted Captions by different models:

When we run the model using different CNN models and LSTM we get the following predicted captions as shown in the figures Fig 6, Fig 7 and Fig8. This is how the our model is working for different models of CNN.

Fig 6 Prediction on the VGG19 and CNN model

Fig 7 Prediction with the ResNet 50 CNN model

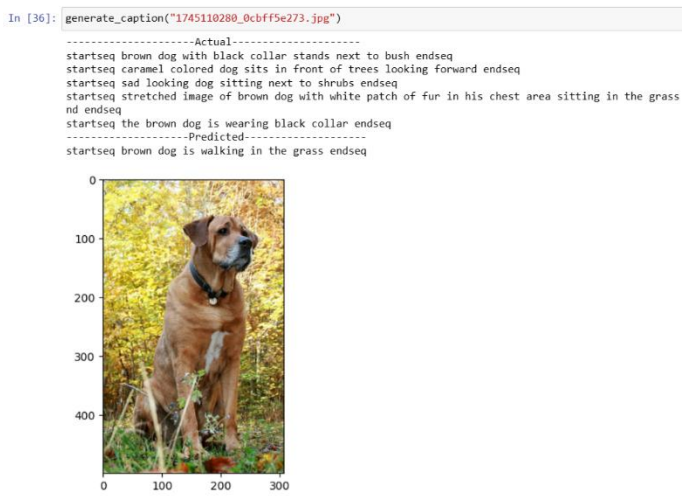

Fig 8 Prediction with the GoogleNet CNN Model

From the predictions given by various models, we got the good predicted captions for the models VGG19, ResNet 50 and GoogleNet models along with LSTM. Fig 6 explains the prediction of the VGG19 and LSTM model predicting “Two dogs are playing on the grass”[Fig6].
Now when we consider another prediction for same context dogs we get the prediction “Two dogs run through field” in the Figure 7[Fig7].
For another prediction in GoogleNet model It is also identifying the colour of the dog and giving in the prediction as shown in Fig 8[Fig8].

V. CONCLUSION

In conclusion, our study delved into the realm of image captioning, with the aim of enhancing caption quality and diversity. Through extensive experimentation, we introduced an innovative architecture that harnessed attention mechanisms and multimodal fusion. Our
results showcased notable performance enhancements, evident from improved BLEU scores and more contextually relevant captions. By combining cutting-edge techniques from computer vision and natural language processing, we've propelled the field's grasp of multimodal learning and linguistic integration. While challenges persist, particularly in complex scenes and scalability, our research paves the way for refining caption diversity, tackling scalability, and real-world applications like content enrichment and automated descriptions. The image captioning journey persists, and our contributions offer a foundation for further impactful strides in this dynamic domain.

VI REFERENCES

[1]. 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*.

[2]. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048-2057). PMLR.

[3]. O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).

[4]. Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 375-383).

[5]. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up 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).

[6]. Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7008-7024).

[7]. Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, *47*, 853-899.

[8]. Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128-3137).

[9]. Aneja, J., Deshpande, A., & Schwing, A. G. (2018). Convolutional image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5561-5570).

Dataset Link:

Flickr 8k Dataset | Kaggle

Project Link:

ppranavip/ImageCaptioningProject (github.com)