Visual Captioning System Using CNN and LSTM

Patel Mitanshu Pareshbhai[1], Mistry Dhwani Yogeshbhai[2], Chauhan Harshwardhansinh Kiransinh[3]
Khot Pranav Sureshbhai[4], Patel Swapnil Rameshbhai[5]
[1, 2, 3, 4, 5] Department of Computer Engineering, Sigma Institute of Engineering, Vadodara

***Abstract*—Visual captioning system using CNN and LSTM is a popular technique for generating textual descriptions of images. Convolutional Neural Networks (CNNs) are used to extract visual features from images, which are then fed into a Long Short-Term Memory (LSTM) network to generate captions. This system uses an encoder-decoder architecture, where the CNN serves as the encoder, and the LSTM network serves as the decoder. The encoder extracts visual features from an input image, and the decoder generates a caption by decoding the extracted features. This system has shown promising results on various benchmark datasets and has a wide range of applications in fields such as image and video captioning, visual question answering, and assistive technologies for the visually impaired.**

***Keywords*—CNN, LSTM, visual features, feature extraction, image description, encoder-decoder architecture.**

I. INTRODUCTION

Visual captioning is the task of generating textual descriptions of images. It has been an active area of research in computer vision and natural language processing for several years. The goal of visual captioning is to develop a system that can automatically generate a natural language description of an image that is both accurate and descriptive. Visual captioning has a wide range of applications, including image and video captioning, visual question answering, and assistive technologies for the visually impaired.

One of the most popular approaches to visual captioning is to use a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. CNNs are used to extract visual features from images, while LSTMs are used to generate captions. This system uses an encoder-decoder architecture, where the CNN serves as the encoder, and the LSTM network serves as the decoder. The encoder extracts visual features from an input image, and the decoder generates a caption by decoding the extracted features.

In this article, we will provide an overview of the visual captioning system using CNN and LSTM. We will discuss the encoder-decoder architecture, the training process, the evaluation metrics used to measure the performance of the system and the complete procedure for deploying this system as a prototype. We will also discuss the applications of visual captioning and the challenges that researchers face in developing accurate and descriptive captioning systems.

A truly impressive Visual Captioning System feature is to handle a wide range of visual content, including complex images and videos with numerous objects and scenes. The visual captioning in Real-time is extraordinary milestone to achieve. Also its ability to improve over time through Machine Learning and feedbacks from user. The system should be able to learn from its mistakes and improve the accuracy and relevance of generated captions through a continuous process.

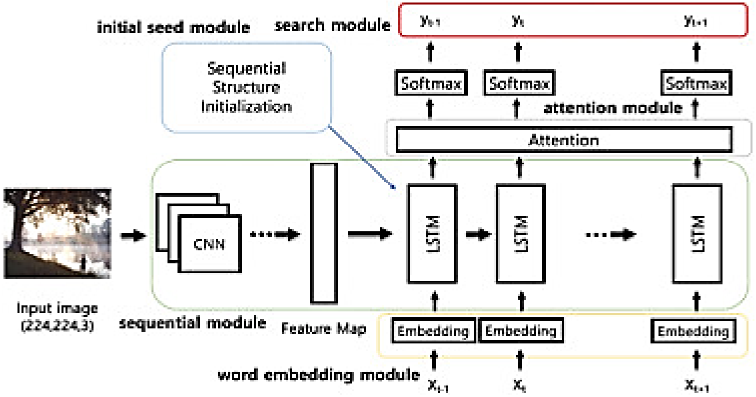


Fig. 1. Visual Captioning System Architecture

II. TECHNOLOGIES USED

A. Flickr8k Dataset

It is a benchmark collection for sentence-based image description and search, consisting of 8,000 images that are each paired with five different captions which provide clear descriptions of the salient entities and events. The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations

B. CNN

CNN stands for Convolutional Neural Network, which is a type of deep learning neural network that is widely used in computer vision tasks such as image classification, object detection, and image segmentation. The key feature of CNNs is the use of convolutional layers, which can learn to detect local patterns in an image, such as edges, corners, and textures.

The architecture of a typical CNN consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The input to the CNN is an image, which is represented as a grid of pixels. The convolutional layers apply a set of filters to the image, each of which detects a specific local pattern. The pooling layers downsample the output of the convolutional layers, reducing the spatial dimensions of the feature maps. The fully connected layers are used to classify the input image into one of several categories.

One of the main advantages of CNNs is their ability to automatically learn useful features from images, without the need for manual feature extraction. This has led to significant improvements in computer vision tasks such as image classification, where CNNs have achieved state-of-the-art performance on many benchmark datasets.

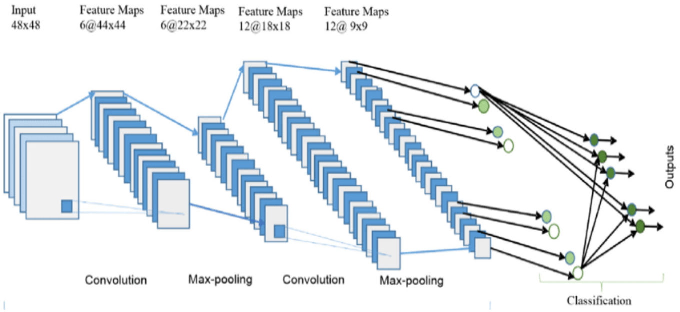


Fig. 2. CNN-Feature Extraction

C. LSTM

CNN stands for Convolutional Neural Network, which is a type of deep learning neural network that is widely used in computer vision tasks such as image classification, object detection, and image segmentation. The key feature of CNNs is the use of convolutional layers, which can learn to detect local patterns in an image, such as edges, corners, and textures.

The architecture of a typical CNN consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The input to the CNN is an image, which is represented as a grid of pixels. The convolutional layers apply a set of filters to the image, each of which detects a specific local pattern. The pooling layers downsample the output of the convolutional layers, reducing the spatial dimensions of the feature maps. The fully connected layers are used to classify the input image into one of several categories.

One of the main advantages of CNNs is their ability to automatically learn useful features from images, without the need for manual feature extraction. This has led to significant improvements in computer vision tasks such as image classification, where CNNs have achieved state-of-the-art performance on many benchmark datasets. LSTMs can be used to model the probability distribution of words in a language, allowing them to generate realistic sentences and paragraphs of text.

D. VGG-16

VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.

E. Programming languages

Python-3.9, HTML and CSS

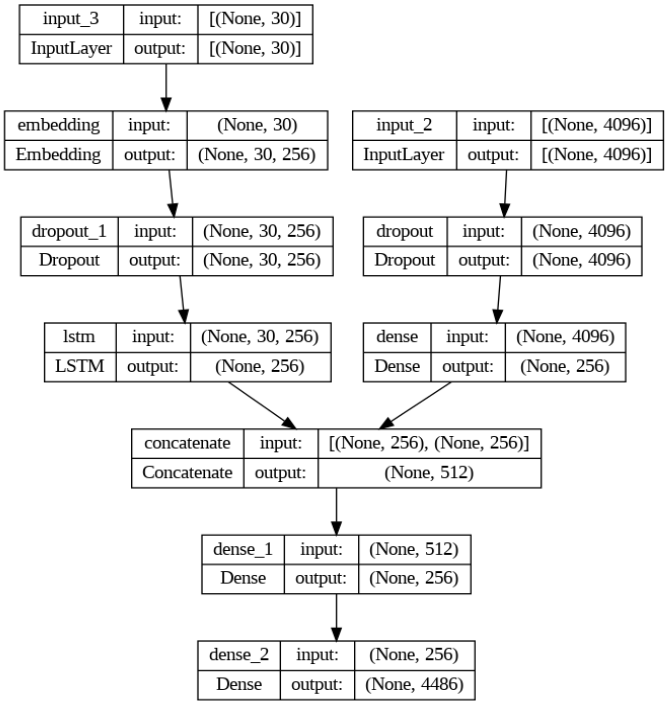
F. Flask

Flask is a popular Python web framework used for building web applications. It is a lightweight and modular framework that allows developers to quickly build and deploy web applications. Flask is built on top of the Werkzeug WSGI toolkit and the Jinja2 template engine, which provide powerful features for building web applications.

G. Google Colab

Google Colaboratory, or Colab for short, is a free cloud-based platform provided by Google that allows users to write and run Python code in a browser-based environment. It is a popular tool for machine learning and data science, as it provides access to powerful GPUs and TPUs for training models.

III. RESULT



input_3	input:	[(None, 30)]
InputLayer	output:	[(None, 30)]

embedding	input:	(None, 30)
Embedding	output:	(None, 30, 256)

input_2	input:	[(None, 4096)]
InputLayer	output:	[(None, 4096)]

dropout_1	input:	(None, 30, 256)
Dropout	output:	(None, 30, 256)

dropout	input:	(None, 4096)
Dropout	output:	(None, 4096)

lstm	input:	(None, 30, 256)
LSTM	output:	(None, 256)

dense	input:	(None, 4096)
Dense	output:	(None, 256)

concatenate	input:	[(None, 256), (None, 256)]
Concatenate	output:	(None, 512)

dense_1	input:	(None, 512)
Dense	output:	(None, 256)

dense_2	input:	(None, 256)
Dense	output:	(None, 4486)

Fig. 4. Trained model summary

A. Measuring accuracy with training loss and validation loss

In machine learning, training loss and validation loss are two commonly used metrics to evaluate the performance of a model during training.

Training loss is the error or difference between the predicted output of a model and the actual output on the training data. The goal of training is to minimize this loss by adjusting the model parameters through methods such as back-propagation and optimization algorithms.

Validation loss, on the other hand, is the error or difference between the predicted output of a model and the actual output on a separate validation dataset. This dataset is used to evaluate the performance of the model on unseen data and to prevent overfitting. Overfitting occurs when a model is too complex and captures noise in the training data, leading to poor performance on new data.

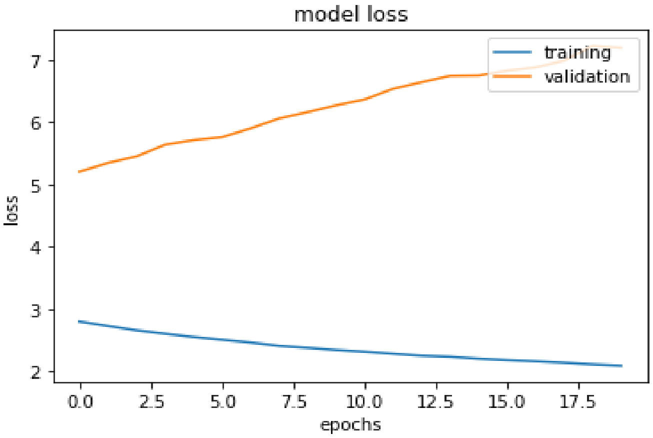
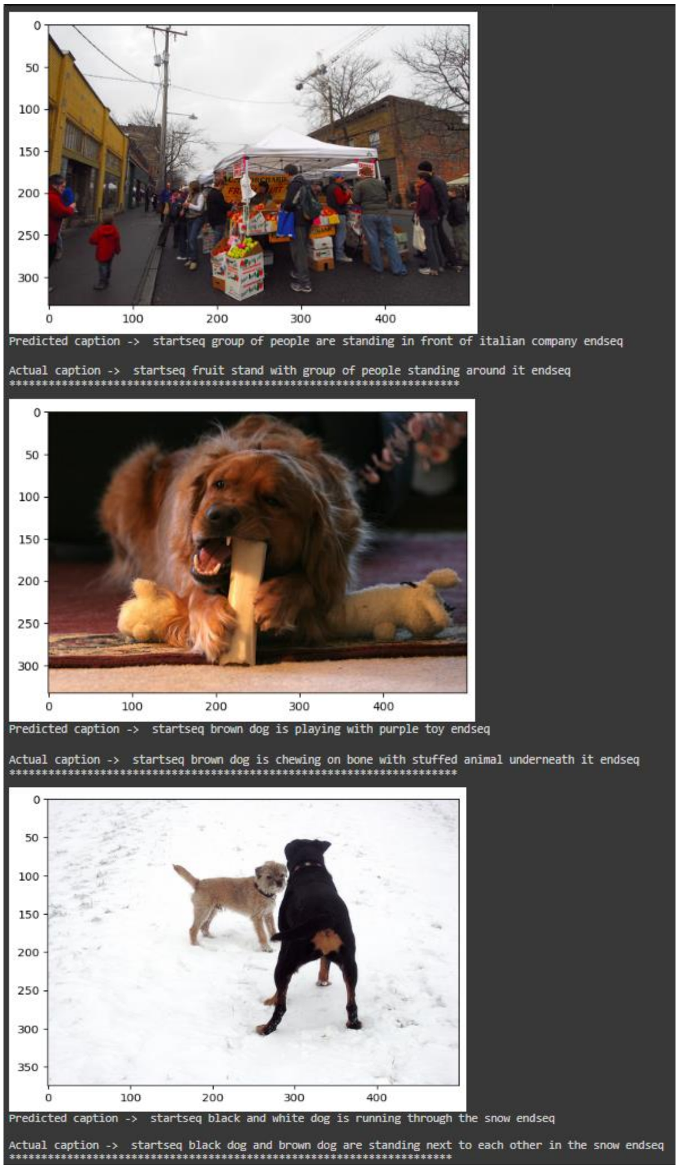


Fig. 4. Training loss and validation loss against epochs

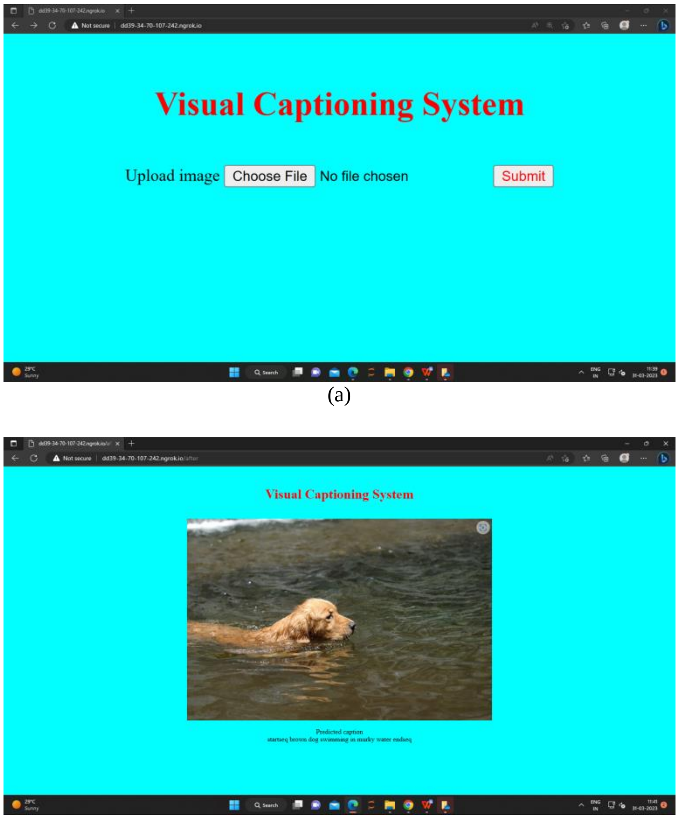
B. *Predicted captions vs Actual captions*



```
Predicted caption ->  startseq group of people are standing in front of italian company endseq

Actual caption ->  startseq fruit stand with group of people standing around it endseq
**********************************************************************
Predicted caption ->  startseq brown dog is playing with purple toy endseq

Actual caption ->  startseq brown dog is chewing on bone with stuffed animal underneath it endseq
**********************************************************************
Predicted caption ->  startseq black and white dog is running through the snow endseq

Actual caption ->  startseq black dog and brown dog are standing next to each other in the snow endseq
**********************************************************************
```

Fig. 5. Predicted captions vs actual captions.

C. *Deploying model in web application using Flask framework*



Visual Captioning System

Upload image Choose File No file chosen Submit

(a)

Visual Captioning System

Predicted caption
startseq brown dog swimming in murky water endseq

(b)

Fig. 6. Web application: (a) Home page; (b) Result page.

IV. Conclusion

The Visual Captioning System has significant potential to transform the way we consume and interact with visual media, improving access and understanding for individuals with disabilities and language learners, as well as enhancing the automatic indexing and retrieval of visual content for researchers, academicians and content creators. .In our article we discussed the architecture of a visual captioning system that uses CNN and LSTM models to generate captions for images and the. We also discussed how Flask can be used to build a web application for visual captioning. The integration between trained model and web application makes our deployment successful

Overall, the development and implementation of a visual captioning system is an exciting and challenging area of research that has the potential to have a significant impact on society, and the advancements in artificial intelligence where machine learning continues to push the boundaries of what is more possible in this field.

V. Future scope

The field of visual captioning has seen significant progress in recent years, thanks to the advancements in deep learning and natural language processing techniques. However, there is still a lot of room for improvement and future research in this area. Here are some potential areas of future scope for visual captioning systems:

A. *Improved caption quality*: Despite the progress made in visual captioning, the quality of generated captions can still be improved. Future research can focus on developing more sophisticated models that can capture more complex relationships between images and language, resulting in more accurate and descriptive captions.

B. *Multi-modal captioning:* Current visual captioning systems typically only use visual features to generate captions. However, incorporating other modalities, such as audio or text, can lead to more comprehensive and informative captions.

C. *Fine-grained image understanding*: Current visual captioning systems typically focus on high-level visual features, such as object recognition and scene understanding. Future research can focus on developing models that can capture more fine-grained details, such as image textures and colors.

D. *Transfer learning:* Transfer learning has been successfully applied in various computer vision and natural language processing tasks. Future research can explore the potential of transfer learning in visual captioning, where models trained on one dataset can be fine-tuned on a different dataset to improve performance.

E. *Evaluation metrics:* The evaluation of visual captioning systems is still an active area of research. Future work can focus on developing more accurate and comprehensive evaluation metrics that can capture the quality of generated captions more effectively.

In summary, the future of visual captioning systems is bright, and there is still much to explore and improve. With further research and development, visual captioning systems have the potential to become even more accurate and informative, opening up new avenues for their application in a wide range of domains.

VI. Reference

[1] https://www.kaggle.com/code/shweta2407/vgg16-and-lstm-image-caption-generator

[2] https://www.sciencedirect.com/science/article/pii/S2405959520301429

[3] https://coderzpy.com/cnn-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more/