

# Identification and Captioning using Deep Learning Techniques for Multimodal Images

Bhairavi D. Naik, Nikita R. Pardeshi, Kinjal S. Patil, Aarti S. Shinde

**MKSSS's Cummins College of Engineering for Women**

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

## Abstract

In recent years, with the development of deep learning, the combination of computer vision and natural language processing has aroused great attention in the past few years. With the rapid development of artificial intelligence, image caption has gradually attracted the attention of many researchers in the field of artificial intelligence and has become an interesting and arduous task. Captions have become one of the most needed tools in the modern world. Captions, which automatically generate natural language descriptions according to the content observed in an image, are an important part of scene understanding, combining computer vision knowledge and natural language processing. The process of generating image descriptions is called image Captioning. You need to recognize important objects, their attributes, and relationships between objects in the image. Generates sentences that are both syntactically and semantically correct. This article presents a deep learning model that uses computer vision and machine translation to describe images and generate captions. This paper aims at recognizing different objects in an image and recognizing relationships between these objects to generate captions. The dataset used was Flickr8k and the programming language used was Python3 to demonstrate the proposed project.

The application of image captions is extensive and significant, for example, the realization of human-computer interaction.

**Keywords:** Image, Caption, CNN, VGG16, RNN, LSTM.

## Introduction

Over the past few years, computer vision has made significant advances in image processing, such as image classification and object detection. Advances in image classification and object detection are making it possible to automatically generate one or more sentences to understand the visual content of an image, a problem known as image captioning.

Caption Generation is an interesting AI task that generates descriptive sentences for a given image. These include dual methods in computer vision to understand image content, and language models in natural language processing to translate image understanding into words in the correct order.

Automatic technology of image content using natural language is a basic and difficult task. The potential impact is huge. For example, it can help blind people better understand the content of images on the Internet. It can also provide more accurate and compressed image/video information in scenarios such as social media image sharing or CCTV systems.

Image captions have many applications, such as recommending editing applications, using virtual assistants, indexing images, visually impaired, social media, and other natural language processing applications.

It has been demonstrated that deep learning models can achieve optimal results in the field of caption generation tasks. Define a single end-to-end model to predict photo captions instead of complex data preparation or pipelines of custom models. To evaluate the model, we measure its performance on the Flickr 8K dataset using standard BLEU metrics. These results show that the proposed model outperforms the standard model in terms of image subtitles in performance evaluation.

## Related Work

We present a synthetic output generator that localizes and describes the objects, properties, and relationships of images in a natural language format. So, we will combine these.

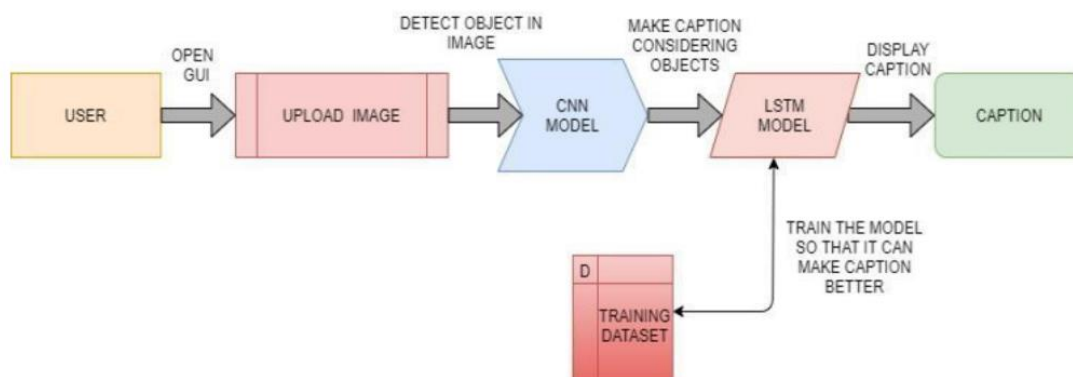
architectures to create an image caption generator model. This is also called a CNN-RNN model.

- CNNs are used to extract features from images.
- LSTM uses information from CNNs to help create image descriptions.

**CNN :** Convolutional Neural Networks are specialized deep neural networks that can process data with input shapes such as two-dimensional matrices. Images are easy to represent as 2D matrices and CNNs are very useful for working with images. CNNs are mainly used to classify images and determine whether an image is a bird, airplane, superman, etc. It scans the image from left to right and top to bottom to extract important features from the image and classify the image by combining the features. It can process images that have been translated, rotated, scaled and perspective changed.

**LSTM :** LSTM stands for Long Short Term Memory and is a type of Recurrent Neural Network (RNN) suitable for sequence prediction tasks. It can predict what the next word will be based on the previous text. It overcomes the limitations of short-term memory RNNs and proves to be superior to conventional RNNs. LSTMs can pass relevant information throughout input data processing and discard unnecessary information with the help of forget gates.

## DFD Diagram :



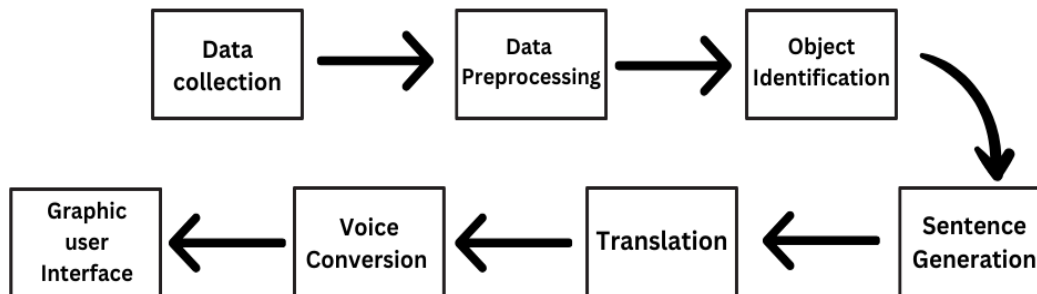
## Literature Review

Previously, extensive research has been done in the field of generating image captions and creating content for image captions. Authors suggested content selection methods for generating image captions. This document [1] mainly uses three categories of features to create content: geometric, conceptual, and visual. In addition, many methods have been proposed in the past to generate image captions. They are divided into three categories. i.e., template-based methods, search-based methods, and deep neural network (encoder-decoder) methods. These models are often built using CNNs to encode images and extract visual information, and RNNs are used to decode visual information into sentences. In this article [2], the authors have proposed a template slot for title creation that is populated with a predictable triple of visual components (object, action, scene). Again, in article [3], the authors used a conditional random field (CRF) based technique to obtain objects (such as people) attributes, and prepositions. This model is evaluated on the PASCAL dataset using the BLEU and ROUGE scores. The best BLEU score for this task was 0.18, the corresponding highest ROUGE was 0.25. The authors of the paper [4] suggested a way to create a title by selecting valuable phrases from existing titles and carefully combining them to create a new title. We used a dataset of one million caption images, of which 1000 images were set aside as a test set for calculating BLEU (0.189) and METEOR (0.101) scores. Most of these methods are difficult to develop and rely on predefined templates. Because of their dependencies on these patterns, these methods cannot generate variable-length sentences or signatures. A template-based approach allows you to create syntactically correct signatures. However, because of the predefined templates, the generated labels have a fixed length. In the paper [5], authors proposed a data-driven based approach for image description generation using retrieval based technique. They concluded that the proposed method provides efficient and relevant results to produce image captions. Although these strategies provide syntactically valid and generic sentences, they fail to produce image-specific and semantically correct sentences.

In the paper [6], the authors proposed a dual graph convolutional network with transformation and curriculum for image labeling. They assessed the results of the MS-COCO dataset with a BLEU-1 score of 82.2 and a BLEU-2 score of 67.6. The authors of this article [7] proposed a Neural Image Caption (NIC) model based on an encoder-decoder architecture. In this model, a CNN is used as an encoder, and the final layer of the CNN is connected with an RNN decoder that generates a text signature. In this model, LSTM is used as RNN. Again, in the paper [8], the authors used visual and semantic similarity scores to cluster similar images. They merge the images together, retrieving captions of the input image from captions of similar images in the same cluster. Some researchers have proposed a ranking-based framework to generate captions for each image by exposing it to sentence-based image captioning.

As a result of conducting research on generating captions for images mentioned above, several important research points were identified. First, the CNN models used in most state-of-the-art systems are pre-trained on object-specific ImageNet datasets rather than scene-specific. Therefore, these models provide per-subject results. Second, most articles only reported results in terms of one or two scoring metrics, such as accuracy and BLEU-1 score. In the proposed model to train a CNN, we use the VGG16 Hybrid Places1365 model as a CNN to obtain object- and scene-specific results. This model was pre-trained on ImageNet. In addition, to review most of the articles, we evaluated the results using several evaluation metrics: BLEU, METEOR, ROUGE, and GLEU measures.

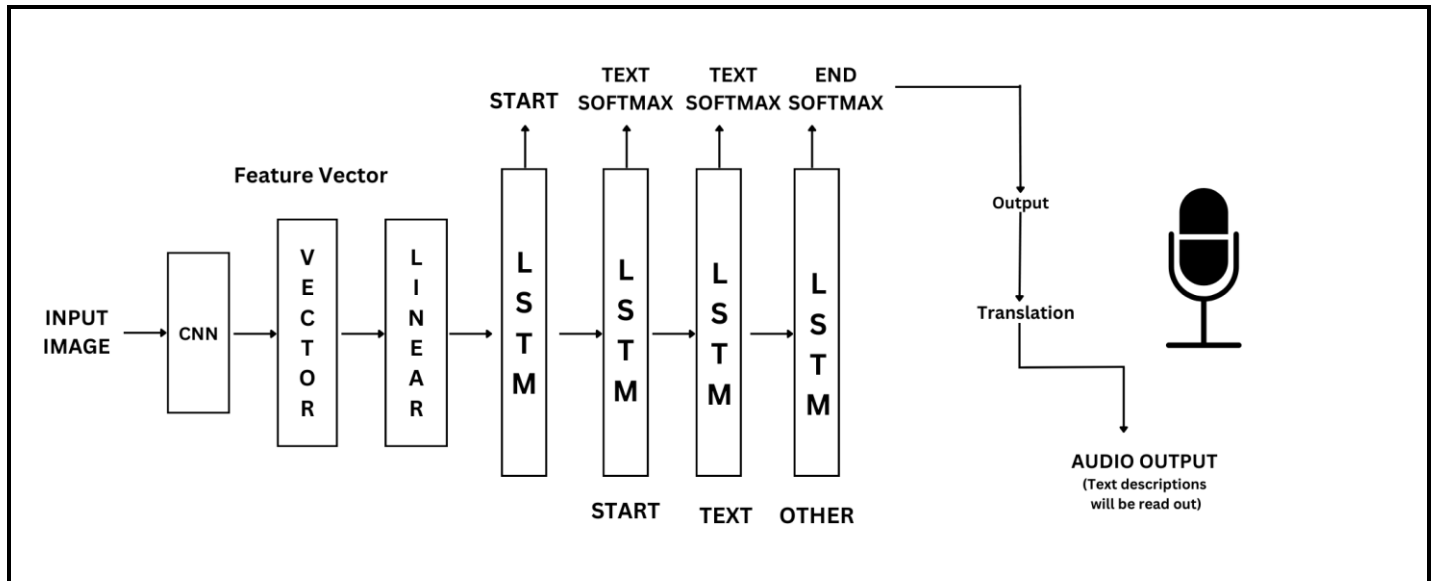
## Methodology



- 1) Data Gathered: To collect the necessary data, we gathered a substantial dataset of images paired with their corresponding captions. This can be done by manually creating captions for images or using existing datasets such as Flickr8k, Flickr30k, or MSCOCO.
- 2) Data Preprocessing:
  - a) The Dataset in which images are present are preprocessed to remove noise and convert to a suitable format for analysis. Data preprocessing may include several conversions to a suitable format for analysis. Data preprocessing may include several techniques to enhance the quality of the images, such as resizing, conversion to grayscale, or applying filters.
  - b) The captions in the dataset are preprocessed by cleaning, removing noise or unwanted characters, and converting the text into a standardized format.
- 3) Object Identification: It is the process of detecting and recognizing objects in an image. It is done by using pre-trained models that have already been trained on large datasets of images and can recognize specific objects in real-time. The visual features are extracted from the images using VGG16 pre-trained model of Convolutional Neural Network (CNN).
- 4) Sentence Generation: LSTM, part of RNN, is used to generate captions. The model takes input image features and word sequence is generated as output. This process can involve several sub-steps such as:
  - Data splitted into validation, training, test sets.
  - Encoding captions using word embedding.
  - Training set is used to train the model and tuning hyperparameters.
  - To measure performance, Model evaluation on the validation and test sets.
- 5) Translation: The output caption is translated into different languages using google translator. It translates text from one language to another.
- 6) Voice Conversion: The translated output is converted to voice using the gTTS

library. The gTTS helps to convert text into voice.

- 7) GUI : It is used for better user experience and more accessible for people with disabilities. It gives various options to the user for conversion of stored caption into different languages and also in voice.



- Image as input is taken from the Dataset.
- Objects that are present in the image are detected by Convolutional neural networks. It extracts the most relevant and discriminative features of an image and these features are then stored as a set of feature vector values, it predicts the features using pooling functions.
- After the feature extraction and classification process is complete, the output is passed to a LSTM layer, which uses the previously predicted words in a sentence to predict the upcoming word in the sequence.
- The softmax function takes the output and maps it to a probability distribution over the possible classes. This allows the network to predict the most accurate output. It also overcomes the overfitting problem.

## Training Phase

In the training phase, you feed a pair of input images and their corresponding captions to the image caption model. A VGG model is trained to identify all possible objects in an image. On the other hand, part of the LSTM model is trained to predict every word in a sentence and every word before it after seeing an image. For each signature, we add two extra characters to indicate the beginning and end of the sequence. Stop generating sentences and mark the end of a line whenever a stop word appears. The loss function of the model is computed with  $I$  being the input image and  $S$  being the generated header.  $N$  is the length of the generated sentence.  $p_t$  and  $S_t$  denote probabilities and predicted words at time  $t$ , respectively. During training, we tried to minimize this loss function.

$$L(I, S) = - \sum_{t=1}^N \log p_t(S_t)$$

## Implementation

Model implementation was performed using a Python environment. Keras 2.0 was used to implement the deep learning model due to the existence of the VGG network used to identify objects. The TensorFlow library is installed as a backend to the Keras framework for building and training deep neural networks. TensorFlow is a deep learning library developed by Google.

It provides a heterogeneous platform for algorithm execution. This means that it can run on low power devices such as mobile devices as well as large distributed systems with thousands of GPUs. The neural network was trained on an Nvidia Geforce 1050 GPU with 640 Cuda cores. To define the network structure, TensorFlow uses a graph definition.

Once the schedule is defined, it can run on any supported device. The photo features are precomputed and stored using a pretrained model. These features are then loaded into the model as interpretations for a given picture in the dataset to reduce the redundancy of running each picture across the network each time a new language model construct is tested. Image feature preloading is also performed to implement a real-time image captioning model. The architecture of the model is shown in Figure 2.

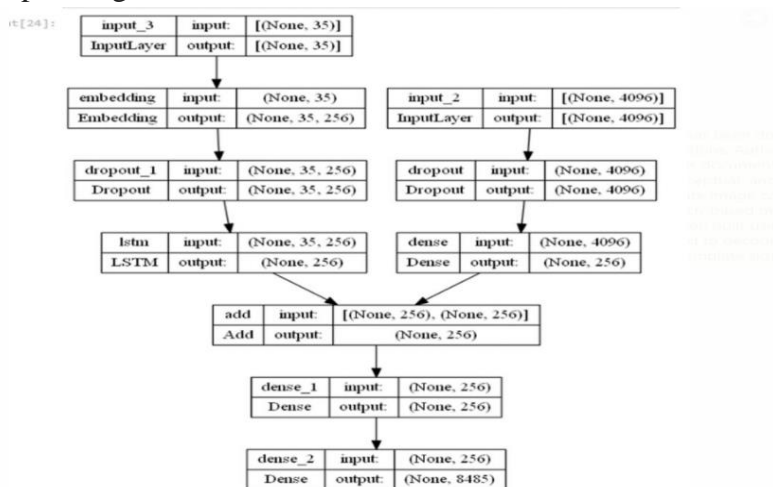



Figure 2: Image Captioning Model




## RESULTS

 generate\_caption("101669240\_b2d3e7f17b.jpg")

-----Actual-----  
 startseq man in hat is displaying pictures next to skier in blue hat endseq  
 startseq man skis past another man displaying paintings in the snow endseq  
 startseq person wearing skis looking at framed pictures set up in the snow endseq  
 startseq skier looks at framed pictures in the snow next to trees endseq  
 startseq man on skis looking at artwork for sale in the snow endseq  
 -----Predicted-----  
 startseq  
 two people displaying paintings in the snow endseq endseq endseq endseq endseq endseq endseq endseq endseq endseq endseq endseq



 generate\_caption('1001773457\_577c3a7d70.jpg')

-----Actual-----  
 startseq black dog and spotted dog are fighting endseq  
 startseq black dog and tri-colored dog playing with each other on the road endseq  
 startseq black dog and white dog with brown spots are staring at each other in the street endseq  
 startseq two dogs of different breeds looking at each other on the road endseq  
 startseq two dogs on pavement moving toward each other endseq  
 -----Predicted-----  
 startseq  
 two dogs playing with each other in the street endseq endseq endseq endseq endseq endseq endseq endseq endseq endseq endseq endseq



## FUTUREWORK

Future Scope of our project is to convert the text directly into braille language.

We can build an Android application for captions for social media uses.

We can compare different algorithms to increase the accuracy.

### **Benefits**

1. Intelligent monitoring
2. Human-computer interaction
3. Image and Video annotation

### **Major Challenges**

#### **SOFTWARE REQUIREMENT**

**Tensor-flow:** Tensor-Flow is an end-to-end open source platform for machine learning. Tensor-flow is developed by Google and incorporates the most common modules into a deep learning framework. It supports many modern networks such as CNNs and RNNs with different settings. Tensor-flow is designed for incredible flexibility, portability, and high-performance hardware.

**PyTorch:** PyTorch is a Python-based scientific computing package that serves two purposes. An alternative to NumPy to harness the power of GPUs and maximize flexibility and speed with a deep learning research platform.

**Keras:** Keras is a high-level neural network API written in Python and can be run on Tensor-flow, CNTK, or Theano.

It is designed with a focus on allowing you to experiment quickly. The ability to move from ideas to results with minimal delay is key to good research. Keras enables easy and fast prototyping (through its user-friendly line of modularity and extensibility). Keras supports both convolutional and recurrent networks, and a combination of both.

#### **HARDWARE REQUIREMENT**

**GPU-** Compared to CPU, the performance of matrix multiplication on Graphics Processing Unit is significantly better. With GPU computing resources, all the deep learning tools mentioned achieve much higher speedup when compared to their CPU-only versions GPUs have become the platform of choice for training large, complex Neural Network based systems because of their ability to accelerate the systems.

**TPU-** Tensor Processing Unit (Domain-Specific Architecture) is a custom chip that has been deployed in Google data centers since 2015. DNNs are dominated by tensors, so the architects created instructions that operate on tensors of data rather than one data element per instruction. To reduce the time of deployment, TPU was designed to be a coprocessor on the PCI Express I/O bus rather than be tightly integrated with a CPU, allowing it to plug into existing servers just as a GPU does.

The goal was to reduce the amount of I/O between TPU and CPU by running the entire inference model on the TPU. Minimalism is a virtue of domain processors.



## CONCLUSION

Based on the results obtained, it can be seen that the deep learning methodology used here has produced successful results. The CNN and LSTM, working together in the right synchronization, were able to find connections between objects in images. To compare the accuracy of our predictive signatures, we can use the BLEU score (Bilingual Studies) [5, 8] and compare them to the target signatures in the Flickr8k test dataset. The BLEU score is used in text translation to rate the translated text against one or more reference translations. Over the years, several different neural network techniques have been used to create hybrid image caption generators like the one proposed here.

For example, use the VGG16 model instead of the Xception model or the GRU model instead of the STM model. You can also compare these models using the BLEU score to see which model is most accurate. This article introduced various new developments in machine learning and artificial intelligence and how vast the field is. Indeed, several topics in this document are open to further research and development, and the document itself attempts to cover the basics of creating an image caption generator.

## References

- [1] Georgios Barlas, Christos Veinidis, and Avi Arampatzis. What we see in a photograph: content selection for image captioning. *The Visual Computer*, 37(6):1309–1326, 2021.
- [2] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010.
- [3] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. Baby talk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.
- [4] Yejin Choi, Tamara L Berg, U N C Chapel Hill, Chapel Hill, and Stony Brook. TREE TALK: Composition and Compression of Trees for Image Descriptions. 2:351–362, 2014
- [5] Vicente Ordonez, Xufeng Han, Polina Kuznetsova, Girish Kulkarni, Margaret Mitchell, Kota Yamaguchi, Karl Stratos, Amit Goyal, Jesse Dodge, Alyssa Mensch, et al. Large scale retrieval and generation of image descriptions. *International Journal of Computer Vision*, 119(1):46–59, 2016.
- [6] Xinzhi Dong, Chengjiang Long, Wenju Xu, and Chunxia Xiao. Dual graph convolutional networks with transformer and curriculum learning for image captioning. *arXiv preprint arXiv:2108.02366*, 2021
- [7] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:3156–3164, 2015.
- [8] Chen Sun, Chuang Gan, and Ram Nevatia. Automatic concept discovery from parallel text and visual corpora. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2596–2604, 2015.