IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

# IMAGE SCENARIO DESCRIPTION FOR THE BLIND USING DSP

**1.** A. V. D. N. Murthy, **2.** Simhachalam Madireddi,**3.** Koyya Jagadeesh,**4.** Law Bind Pandey, **5.** Relli Kushal Kumar

1 Assistant professor, 2 Student, 3 Student, 4 Student, 5 Student

Department of Computer Science and Engineering,
Lendi Institute of Engineering and Technology (A), Jonnada, Vizianagaram, Andhra Pradesh, India

---

**ABSTRACT -** In the modern era, image captioning has become a widely sought-after tool, with pre-built applications using advanced deep neural network models to generate captions for images. The process involves identifying the key objects, their characteristics, and their relationships in an image to create a descriptive caption. This caption is then transformed into relevant audio through computer vision and machine translation techniques. The purpose of this project is to develop an image-to-speech synthesizer capable of detecting various objects in an image, recognizing their relationships, and generating appropriate audio. To carry out this experiment, we will be using the Flickr8k dataset and Python3 programming language, employing Transfer Learning with the Xception model. This synthesizer software uses an Encoder-Decoder for natural language processing, followed by Digital Signal Processing (DSP) technology to convert the processed text into synthesized speech, making it an ideal tool for visually impaired individuals. Overall, this project aims to create a practical and useful image-to-speech synthesizer with significant potential for aiding those with visual impairments.

**KEYWORDS - Images, Captions, CNN, Xception, RNN, Encoder- Decoder, DSP, Neural Networks.**

---

## I. INTRODUCTION

Generating natural language descriptions for objects detected by computer systems has been a longstanding challenge in the field of artificial intelligence. Despite initial skepticism from computer vision researchers, recent breakthroughs in deep learning, abundant datasets, and increased computing power have allowed for the development of models capable of generating captions for images. This task involves both image processing and natural language processing techniques to identify the content of an image and communicate it in a human language, such as English or other languages.

After generating a natural language description of an image, it can be further improved by using digital signal processing techniques to convert the text into audio. This audio output can be particularly useful for visually impaired individuals, as it allows them to form mental images of the objects and scenes being described. By leveraging digital signal processing methods, the audio output can be optimized for clarity and intelligibility, making it easier for blind individuals to comprehend and visualize the content of the image.

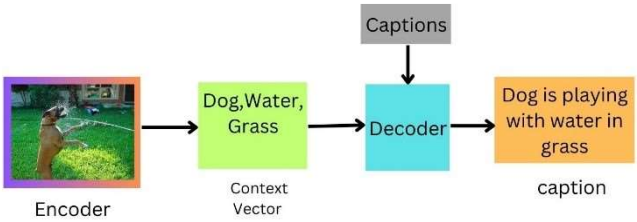


**Figure-1: Generating caption for image using Encoder-Decoder**

Our model is based on a deep learning neural network that consists of a vision CNN followed by a language generating RNN. It generates complete sentences as output captions or descriptive sentences. In recent years a lot of attention has been drawn towards the task of automatically generating captions for images. However, while new datasets often spur considerable innovation, benchmark datasets also require fast, accurate, and competitive evaluation metrics to encourage rapid progress. Being able to automatically describe the content of a picture using properly formed English sentences may be a very challenging task, but it could have an excellent impact, as an example by helping visually impaired people better understand the content of images online. This task is significantly harder, for instance than the well-studied image classification or visual perception tasks, which are a focus within the computer vision community Deep learning methods have demonstrated advanced results on caption generation problems. What is most impressive about these methods is that one end-to-end model is often defined to predict a caption, given a photograph, rather than requiring sophisticated data preparation or a pipeline of specifically designed models. Deep learning has attracted a lot of attention because it's particularly good at a kind of learning that has the potential to be very useful for real-world applications. The ability to find out from unlabeled or unstructured data is a huge benefit for those curious about real-world applications.

## II. PROBLEM STATEMENT

The challenge with image description development initially arose from the use of static object class libraries in images, which were modeled using statistical language models. However, this approach had limitations in accurately describing dynamic and complex images, such as those with multiple objects or actions taking place. Blind people are unable to access or comprehend visual content, such as images, which limits their ability to engage with and understand information that is presented in a visual format.

A. Making use of CNN: It's a Deep Learning algorithm that will intake in a 2D matrix input image, assign importance (learnable weights and biases) to different aspects/objects in the image, and be intelligent enough to be able to differentiate one from the other.
B. This model was advantageous in naming the objects in an image, but it could not tell us about the relationship among them (that's plain image classification).
C. In this paper, we present a generative model built on a deep recurrent architecture that unites recent advances in computer vision and machine translation and that can effectively generate meaningful sentences.
D. Making use of an RNN: They are networks with loops in them, allowing information to persist. LSTMs are a particular kind of RNN, capable of learning long-term dependencies.

## III. PROPOSED METHODOLOGY

A. **Task:** The task is to develop a system that can accept an image input represented as a multidimensional array and produce a textual output that accurately and grammatically describes the content of the image. This system will need to use techniques such as computer vision and natural language processing to analyze the image and generate a coherent and well-formed sentence that captures the key features and attributes of the image. The resulting output should be easily understandable by a human reader and should convey the relevant information contained within the image in a concise and effective manner. In addition to generating a textual output that describes the content of the image, the system should also convert the resulting caption into speech, allowing blind individuals to access the information conveyed in the image through an audio medium.

B. **Corpus**: Our approach to developing the image description system has involved using the Flickr 8K dataset as the corpus for training and evaluation. This dataset comprises 8,000 images, each of which has five captions, providing a rich and diverse set of annotations for each image. By leveraging the multiple captions for each image, our system is better able to capture the range of possible scenarios and accurately describe the content of the image. Additionally, we have made use of the predefined training dataset provided with the Flickr 8K dataset, to facilitate the training process and ensure consistency in our results. By leveraging this dataset and its associated resources, we aim to develop a robust and effective image description system that is capable of accurately and fluently describing a wide range of visual content.

C. **Pre-processing**: Our data pre-processing approach for the image description system involves two main steps: cleaning and pre-processing the images and captions separately. First, we utilize the Xception application of the Keras API, which is built on top of TensorFlow, to pre-process the images. This involves feeding the input data into Xception, which is pre-trained on the large-scale ImageNet dataset. By leveraging transfer learning, we can significantly reduce the amount of time and computational resources required to train our image description model, while also achieving state-of-the-art performance. In addition to image pre-processing, we also clean and pre-process the textual descriptions associated with each image. To accomplish this, we utilize the tokenizer class in Keras to vectorize the text corpus, creating a separate dictionary to store the resulting vectors. This allows us to represent the textual data in a format that can be easily ingested by our model during training. Furthermore, each word in the vocabulary is mapped to a unique index value, enabling us to efficiently generate text sequences during the training process.

D. **Model**: Deep learning is a subset of machine learning that utilizes artificial neural networks composed of several layers in a hierarchy. One of the most common architectures in deep learning is the Encoder-Decoder model. The model has an encoder that takes in the input data, which in image processing could be the pixel values of an image and learns to extract the important features that represent the input data. These encoded features are then passed on to the decoder, which generates the output based on these learned features.

**Image Dataset:**
https://github.com/jbrownlee/Datasets/releases/download/Flickr8k/Flickr8k_Dataset.zip
**Text Dataset:** https://www.kaggle.com/datasets/adityajn105/flickr8k

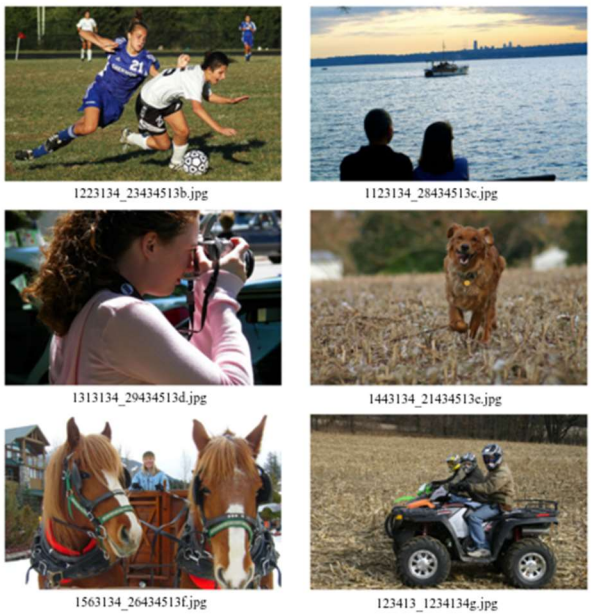


**Figure-2**: **Glimpse of the flickr8k image dataset**

122134_234345: Men are playing game in ground
11231434_284345: Men and women seeing boat
1313134_294345: Women using camera
1443134_21434513: Dog is walking
15613134_264345: Men in carriage
123413_1234134: Men riding in jeep

**Figure-3: Glimpse of flickr8k text file**

**Convolutional Neural Networks (CNN):**
Convolutional Neural Networks (CNNs) are a specialized type of deep neural network that has revolutionized computer vision applications. They are designed to process and analyze data that has a specific input shape, such as images that can be represented as 2D matrices of pixel values.

ResNet is a deep neural network architecture that uses residual connections to improve training performance by allowing information to bypass layers.

VGG16 is a deep convolutional neural network architecture that uses small filters and stacking to increase model depth, allowing for better feature extraction and classification.

Both VGG16 and ResNet are deep convolutional neural networks that are commonly used in image recognition and classification tasks. While they have some differences in architecture, both models work in a similar way to extract features from images.
In the context of image captioning, the VGG16 and ResNet models can be used as encoders in an encoder-decoder architecture. The encoder processes the input image and extracts a set of features, which are then used as input to the decoder to generate a natural language description of the image.

In this architecture, the VGG16 or ResNet model is used as the encoder to extract image features. Specifically, the output of the last convolutional layer of the encoder is used as the input to the decoder. The decoder is typically implemented as a Recurrent Neural Network (RNN) or a Transformer model, which takes the image features as input and generates a description of the image.

**Recurrent Neural Networks (RNN):**
LSTM (Long Short-Term Memory) is a type of Recurrent Neural Network (RNN) that is commonly used for sequence modeling tasks such as language translation and image captioning. In image captioning using encoder-decoder architecture, the LSTM is used as a decoder to generate a sequence of words given an image as input.

The LSTM in the decoder network is used to model the conditional probability distribution of the next word in the caption given the previously generated words and the image feature vector. The LSTM takes as input the concatenation of the image feature vector and the embedding of the previously generated word. The output of the LSTM is then fed into a fully connected layer with a SoftMax activation function to generate the probability distribution over the vocabulary.

**The Encoder-Decoder Algorithm:**

Encoder-decoder is a popular algorithm used in image caption generation. This algorithm involves two main components: an encoder and a decoder.

- The encoder component takes an input image and generates a feature vector that represents the image. This is usually done by passing the image through a pre-trained convolutional neural network (CNN), such as VGG16 or ResNet, which extracts high-level features from the image.

- The output of the encoder is a fixed-length vector that represents the image's content.
- The decoder component takes the feature vector generated by the encoder as input and generates a sequence of words that describe the image.
- This is usually done using a recurrent neural network (RNN), such as a long short-term memory (LSTM) network, which generates a sequence of words one at a time. The decoder starts by inputting a special token, such as <start>, and outputs the first word of the caption.
- The output word is then fed back into the decoder along with the feature vector and used to predict the next word in the sequence.
- This process is repeated until the decoder outputs an end-of-sequence token, such as <end>.
- Once the model is trained, it can be used to generate captions for new images. Given an input image, the encoder generates a feature vector, which is then fed into the decoder to generate a sequence of words that describe the image.
- The generated caption can then be evaluated using metrics such as BLEU or METEOR to assess its quality.
- The generated captions are converted into speech using Digital Signal Processing (DSP).

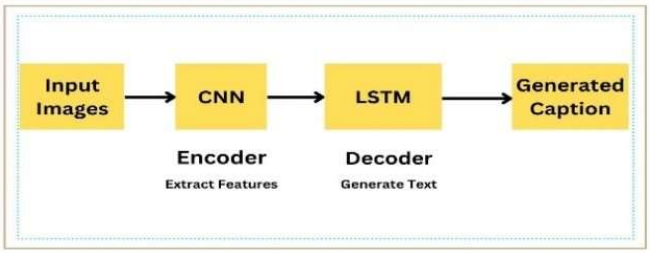


**Figure-4: Encoder-Decoder Architecture**

Digital signal processing (DSP) can be used in image captioning to convert the text generated by the decoder into an audio signal that can be played back to the user. This is typically done using a text-to-speech (TTS) system that converts the text into a sequence of speech sounds.

The process of converting text into speech involves several steps. First, the text is pre-processed to remove any unwanted characters or punctuation marks. Then, a language model is used to predict the phonetic pronunciation of each word in the text. This is typically done using a grapheme-to-phoneme (G2P) conversion algorithm that maps the written form of a word to its corresponding phonetic transcription.

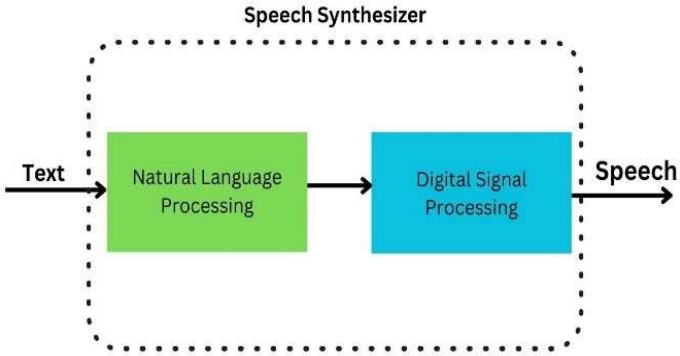


**Figure-5: Digital Signal Processing**

**Architecture:**

The Encoder-Decoder model in image captioning consists of two main components: the Encoder and the Decoder. The encoder takes in an image and generates a fixed-length feature vector that captures the important

visual features of the image. The decoder takes in this feature vector and generates a sequence of words that form a natural language description of the image.

The encoder is typically a convolutional neural network (CNN) that takes in the raw pixel values of the image and learns to extract its visual features. CNN typically consists of several convolutional and pooling layers followed by one or more fully connected layers. The output of the final fully connected layer is a feature vector that captures the important visual features of the image.

The decoder is typically a recurrent neural network (RNN) that takes in the feature vector generated by the encoder and generates a sequence of words that form a natural language description of the image. The RNN typically consists of one or more LSTM cells that learn to generate the sequence of words.

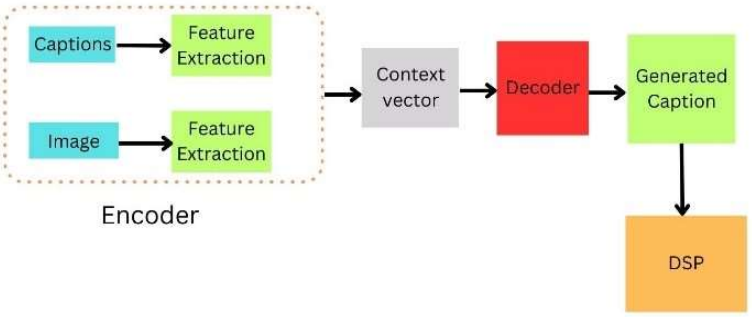


**Figure-6: System Architecture**

Once the model is trained, it can be used to generate captions for new images. During inference, the encoder generates the feature vector for the input image, which is then passed on to the decoder. The decoder generates a sequence of words one at a time, with each word being conditioned on the previously generated words. The generation process continues until an end-of-sequence token is generated or a maximum sequence length is reached.

## IV. EVALUATION

Here's how we can use an encoder-decoder model to generate a caption for this image:

1. **Data preprocessing:** The first step is to preprocess the image data. We can use a pre-trained convolutional neural network (CNN), such as VGG16 or ResNet, to extract features from the image. These features represent the important visual information in the image and will be used as input to the decoder.

2. **Building the encoder:** Now, we create an encoder network, which is a type of neural network designed to accept image features as input and generate a fixed-length vector of encoded information. This vector contains all the essential visual details from the image, which the decoder network will use to generate a caption. To build the encoder, we can employ a pretrained convolutional neural network, like VGG or ResNet, that is well-suited to extracting relevant features from images. By using a pretrained CNN, we can take advantage of its existing knowledge and expertise to create a more effective and efficient encoder, which in turn can lead to more accurate and meaningful image captions.

```
None
6000/6000 [==============================] - 613s 102ms/step - loss: 4.5162
6000/6000 [==============================] - 577s 96ms/step - loss: 3.6705
6000/6000 [==============================] - 591s 98ms/step - loss: 3.3801
6000/6000 [==============================] - 582s 97ms/step - loss: 3.2075
6000/6000 [==============================] - 571s 95ms/step - loss: 3.0898
6000/6000 [==============================] - 600s 100ms/step - loss: 2.9996
6000/6000 [==============================] - 617s 103ms/step - loss: 2.9267
6000/6000 [==============================] - 629s 105ms/step - loss: 2.8732
6000/6000 [==============================] - 636s 106ms/step - loss: 2.8224
6000/6000 [==============================] - 638s 106ms/step - loss: 2.7851
```

**Figure-7: Dataset Loading**

3. **Building the decoder:** Next, we build the decoder, which is another neural network that takes in the encoded information from the encoder and generates the caption word by word. The decoder is typically implemented using a recurrent neural network (RNN), such as a long short-term memory (LSTM) network or a gated recurrent unit (GRU) network. The decoder starts by taking in a special token (such as <start> or <bos>) as input and generates the first word of the caption. It then uses this word as input to generate the next word, and so on, until it generates a special end-of-sentence token (such as <end> or <eos>) to indicate the end of the caption.

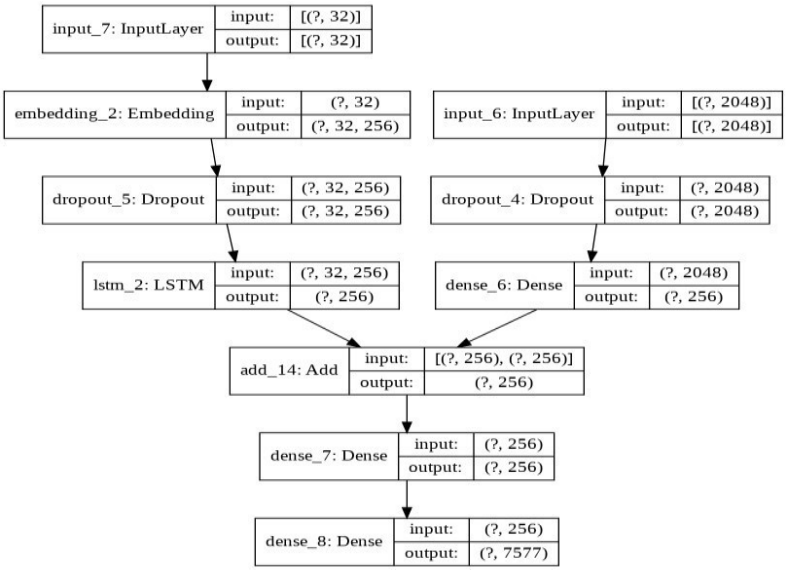


**Figure-8: The Architecture model of LSTM**

4. **Training the model**: Once we have built the encoder-decoder model, we can train it on a dataset of image-caption pairs. During training, the model learns to map the input image to the output caption by adjusting the weights of the neural network using backpropagation and gradient descent. The training data typically consists of images and their corresponding captions. and we use a loss function such as cross-entropy loss to measure the difference between the predicted captions and the ground truth captions.

5. **Generating captions:** Once the model is trained, we can use it to generate captions for new images. To generate a caption for a new image, we first extract its features using the pre-trained CNN and then pass the features through the encoder to get the encoded information. We then use the decoder to generate the caption word by word, starting with the special start token. At each step, the decoder generates the next word based on the encoded information and the previous words generated by the

decoder. We continue generating words until the decoder generates the special end-of-sentence token.

```
Dataset:  6000
Descriptions: train= 6000
Photos: train= 6000
Vocabulary Size: 7577
Description Length:  32
Model: "functional_5"
```

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_7 (InputLayer) | [(None, 32)] | 0 | |
| input_6 (InputLayer) | [(None, 2048)] | 0 | |
| embedding_2 (Embedding) | (None, 32, 256) | 1939712 | input_7[0][0] |
| dropout_4 (Dropout) | (None, 2048) | 0 | input_6[0][0] |
| dropout_5 (Dropout) | (None, 32, 256) | 0 | embedding_2[0][0] |
| dense_6 (Dense) | (None, 256) | 524544 | dropout_4[0][0] |
| lstm_2 (LSTM) | (None, 256) | 525312 | dropout_5[0][0] |
| add_14 (Add) | (None, 256) | 0 | dense_6[0][0]<br>lstm_2[0][0] |
| dense_7 (Dense) | (None, 256) | 65792 | add_14[0][0] |
| dense_8 (Dense) | (None, 7577) | 1947289 | dense_7[0][0] |

```
Total params: 5,002,649
Trainable params: 5,002,649
Non-trainable params: 0
```

**Figure-9: Training the model**

## V. RESULT / ANALYSIS:

For simplicity, a couple of images have been subjected to testing, and the results can be seen in the following images:

1. **Path of Image - 1:**
Flicker8k_Dataset/191737222_094ed5a30.jpg

**Output:**

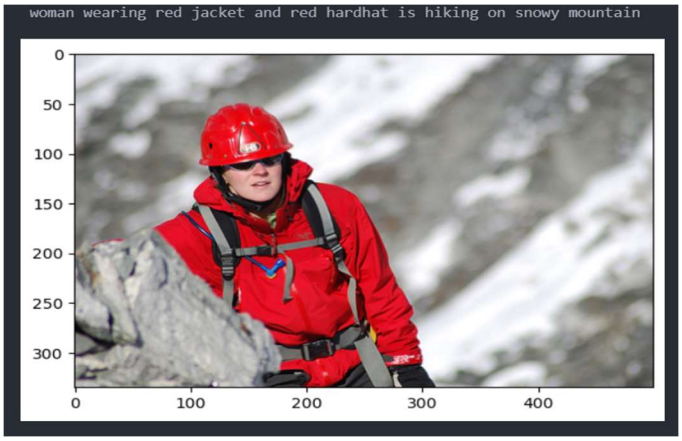


**Figure-10: Caption generated using Encoder-Decoder**

2. **Path of Image - 2:**
Flicker8k_Dataset/191885101_9a3217c5d0.jpg

**Output:**


**Figure-11: Caption generated using Encoder-Decoder**

## VI. CONCLUSION:

The deep learning approach used in this study demonstrated successful results. By combining Convolutional Neural Networks (CNN) and Recurrent Neural Network (RNN) models, the system was able to identify the relationship between objects in the images and generate relevant captions with the help of Encoder-Decoder then further converted into speech with the help of Digital Signal Processing (DSP).

To evaluate the accuracy of the generated captions, the BLEU score was used, which is commonly used to evaluate the quality of translated text. The test dataset used was Flickr8k, which provided reference captions to compare with the generated captions. The BLEU score can also be used to compare the performance of different image caption generators, such as using different models like VGG16 instead of Xception, or GRU instead of LSTM.

BLEU score (Bilingual Evaluation Understudy) is a metric used to evaluate the quality of machine-translated text by comparing it to one or more reference translations. It measures the similarity between the machine-generated output and the human-generated reference text using n-gram overlap.

This paper highlights the vast potential of machine learning and AI and introduced several new developments in the field of image captioning. While this paper covers the essential elements required to create an image caption generator.

In conclusion, the study demonstrated that deep learning models can be effectively used in image caption generation, and the BLEU score can be used to evaluate and compare the performance of different models. Further research and development in this field have the potential to enhance the accuracy and efficiency of image captioning systems.

## VII. REFERENCES:

[1] HaoranWang , Yue Zhang, and Xiaosheng Yu, “An Overview of Image Caption Generation Methods”, (CIN-2020)
[2] B.Krishnakumar, K.Kousalya, S.Gokul, R.Karthikeyan, and D.Kaviyarasu, “IMAGE CAPTION GENERATOR USING DEEP LEARNING”, (international Journal of Advanced Science and Technology- 2020 )
[3] MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga, “A Comprehensive Survey of Deep Learning for Image Captioning”, (ACM-2019)
[4] Rehab Alahmadi, Chung Hyuk Park, and James Hahn, “Sequence-tosequence image caption generator”, (ICMV-2018)
[5] Oriol Vinyals, Alexander Toshev, SamyBengio, and Dumitru Erhan, “Show and Tell: A Neural Image Caption Generator”, (CVPR 1, 2-2015)
[6] Priyanka Kalena, Nishi Malde, Aromal Nair, Saurabh Parkar, and Grishma Sharma, “Visual Image Caption Generator Using Deep learning”, (ICAST-2019)

[7] Pranay Mathur, Aman Gill, Aayush Yadav, Anurag Mishra, and Nand Kumar Bansode,"Camera2Caption: A Real-Time Image Caption Generator", International Conference on Computational Intelligence in Data Science(ICCIDS) - 2017
[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, et al., "Show, attend and tell: Neural image caption generation with visual attention", Proceedings of the International Conference on Machine Learning (ICML), 2015.
[9] J. Redmon, S. Divvala, Girshick and A. Farhadi, "You only look once: Unified real-time object detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
[10] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate.arXiv:1409.0473", 2014.
[11] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi, "Understanding of a convolutional neural network", IEEE – 2017.

## AUTHORS' PROFILES

**A.V.D.N. Murthy**, completed his MCA and M.Tech. in Computer Science and Engineering. He is currently working as Assistant Professor of Lendi Institute of Engineering and Technology, affiliated to JNTUGV University. He is having industrial experience of 1.2 years and teaching experience of 16 years. His areas of interest include Machine Learning, Data mining, Image Processing, Cryptography & Network security, Cloud Computing, Computer Networks and Operating Systems, Big Data.

**Simhachalam Madireddi**, Graduate in Computer Science and Engineering (2023) from Lendi Institute of Engineering and Technology, Jonnada, Vizianagaram, Completed intermediate education from Gayatri, Jr College, Gotlam, Vizianagaram. I have done a few projects like Spotify cloning, Human Scream Detection etc.

**Koyya Jagadeesh**, Graduate in Computer Science and Engineering (2023) from Lendi Institute of Engineering and Technology, Jonnada, Vizianagaram, Completed high school education from Sri Chaitanya, Jr College, Thotapalem, Vizianagaram. Done some web development projects in MERN Stack and a few in AI tech (jagadeesh10th@gmail.com).

**Law Bind Pandey**, Graduate in Computer Science and Engineering (2023) from Lendi Institute of Engineering and Technology, Jonnada, Vizianagaram, Completed high school education from B.V.K Jr College, Vizag. Has done few Mini projects in the field of Machine Learning and Artificial Intelligence. Some of the projects are Scream Detection, recommendation System etc.

.

**Relli Kushal Kumar**, Graduate in Computer Science and Engineering (2023) from Lendi Institute of Engineering and Technology, Jonnada, Vizianagaram, Completed high school education from Bhashyam , Bobbili. Has done few Mini projects in the field of Machine Learning and Artificial Intelligence. Some of the projects are Scream Detection, recommendation System etc.