International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

# Image Caption Generator Using VGG and LSTM For Visually Impaired

Ponnaganti Rama Devi[1], Mannam Thrushanth Deepak[2], Morampudi Lohitha[3], M.Surya Chandra Raju [4], K.Venkata Ramana[5]

*2,3,4,5 Student, Department of computer science engineering Gitam, Visakhapatnam, Andhra Pradesh, India*
*1 Assistant professor, Department of computer science engineering Gitam, Visakhapatnam, Andhra Pradesh, India*

Date of Submission: 05-04-2023 Date of Acceptance: 15-04-2023

**ABSTRACT:**The process of creating an appropriate word sequence for an image is known as image captioning. Implementing picture captioning has advanced due to recent developments in neural networks. Convolutional Neural Networks (CNNs) are used in neural network architecture to build captions that describe the contents of a picture. In this project, TensorFlow Keras was used to implement several different other architectures. The last 10 years have seen an increase in interest in automatic caption synthesis in natural language to describe the visual content of an image due to its possible applications. It is challenging to create captions that have the right linguistic properties because it requires a sophisticated level of image understanding that goes much beyond image categorization and object recognition. In this study, we suggest that the textual description of the image be transformed to speech and transmitted to the user after training photos using the VGG16 deep learning architecture. The user is then supplied a speech file that is a transcription of the written description of the image such that it provides them with a sense of what is around them. The generated caption's quality and accuracy are evaluated using the BLEU score.

**KEYWORDS**: Image captioning, Deep Learning, Neural Networks, BLEU Score.

## I. INTRODUCTION

A recent trend in computer vision and machine learning is teaching computers how to automatically create captions for pictures. Understanding image scenes, extracting features, and translating visual representations into plain languages are all components of this endeavor. The development of assistive technologies for people who are blind and assistance with captioning duties are two areas where this initiative holds out considerable promise. The goal of this project is to create appropriate captions for a given image. The captions will be chosen to reflect the context that the images offer. To create acceptable captions, current methods use convolutional neural networks (CNNs) and recurrent neural networks (RNNs) or their derivatives. These networks provide an encoder-decoder strategy to achieve this goal, using RNNs as decoders to provide linguistic descriptions and VGGs to encode the image into feature vectors. Models' architectures might differ greatly. For the encoder component, features from an image are extracted using a pre-trained CNN. These features are extracted from specific layers after the image has been put into the CNN. The feature vector is then directed into the architecture for picture captioning. The CNNs used for decoding most frequently include AlexNet, VGGNet, ResNet, and GoogleNet (Inception models) Several RNNs are utilised for the decoder portion. Long-short-term memories (LSTM) are especially helpful for handling long-term dependencies in sequences. Through embedding, words from predefined dictionaries are fed into LSTMs. In order to turn words into vectors, statistical techniques and algorithms are used. Images and captions from large, labelled data sets, such those in Microsoft COCO and Flickr, offer details about the events and objects. You can use these captions to write pertinent descriptions for brand-new photos. Metrics like BLUE can be used to gauge how appropriate and pertinent the generated captions are to the data set captions. The length of the caption and general similarity determine the BLEU score. Several relevant research publications have attempted to carry out this work over the past few decades, but they have run into several issues,

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

including language issues, cognitive absurdity, and irrelevant content. To overcome those issues, we have come up with this approach where we use computer vision and NLP techniques to get relevant content with a proper sentence formation which helps visually impaired people to use this model.

## II. PROBLEM IDENTIFICATION & OBJECTIVES

For many visually impaired people the new technological advancements have brought a solution to use their capability to listen what is around them by using Image captioning. The concept of image captioning evolved from deep learning concepts like CNN and also natural language processing (NLP). It is very difficult to understand what is happening around you when we are unable to use our eyes instead, we can use our ears to listen what is happening around us. It is not possible to assist people who are blind to assist them with providing information about their surroundings, but the concept of text generation captures the image for every instant of time and process it through best possible model we generate which provides high accuracy output and helps it to convert into a meaningful and efficient sentence so that whenever a blind person tries to know about the surroundings he can listen to that voice note. This job can be done with the help of making machines understand a 2D image and process it to predict the objects that are present in the image and able to form a meaningful sentence that is understandable for a human. To create the inscriptions, the system we suggested primarily uses two different brain organizations. The Convolutional Neural Network (CNN) is the primary neural network that is used to prepare images as well as to recognize objects in images with the help of various pre-built models like VGG, Inception, or YOLO. As our convolution neural organization model for this project, Network (RNN) based Long Short-Term Memory (LSTM), the second neural organization used, is used to generate subtitles from the constructed object watchwords. Also, we used the Flickr 8k informative collection for this project, where each image has five captions. This application will create a caption for any nature image based on the content of the image. Such a program could enable visually impaired persons to view the world's abundance of photos. This approach automatically creates natural language captions that can be used for image indexing and searching, social media tagging, assisting the blind, etc. Such an application has a wide range of applications. In this research, we will create such an application using Long Short-Term Memory (LSTM) for caption generation and Convolution Neural Networks (CNN) for feature extraction. Summarized calculations based on artificial intelligence won't be accurate because a lot of data is needed to create and certify the model. Recent developments in profound learning have been made to address the information requirements for machine learning calculations. The Deep Learning tasks must be carried out using GPU-based registration for maximum success.

## III. SYSTEM METHODOLOGY

The strategy for solving the problem of image captioning can be divided into five main categories. Collecting, cleaning, and processing data is the initial stage in the captioning process. Afterwards, features are extracted from pre-processed images. The network models that have the potential to yield excellent results are then set up for training. The preprocessed data is separated into training and testing and training data is used for training on the predefined network models. Finally, the evaluation of the model is done on separate test data. Google's TensorFlow library (as well as its deep learning equivalent Keras) were both utilised throughout the entire project. We selected Colab because it gave us a simple method to store our data and because it supported GPUs, which sped up network model training.

### A. Architecture

The basic flow of the projects is as follows:

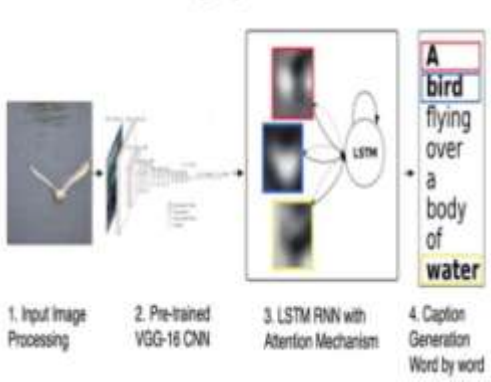


Fig. 1: Conversion of Image to Text

As can be seen, the initial stage is pre-processinga given image and then encoder processes the input image in the.jpg format using pre-built computer vision models like the VGG-16. The third part is using LSTM model we generate captions for the image processed by encoder which allows to label the objects present in the image and forming a meaningful sentence that is grammatically meaningful. The decoder does the

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

second part merging both the image and captions generated from LSTM model to generate a caption. Even though the number of stages for conversion are less each stage need equal amount of processing the data for predicting an efficient caption which is the main aim of this paper. Now we look into the stages involved in captioning a given image:

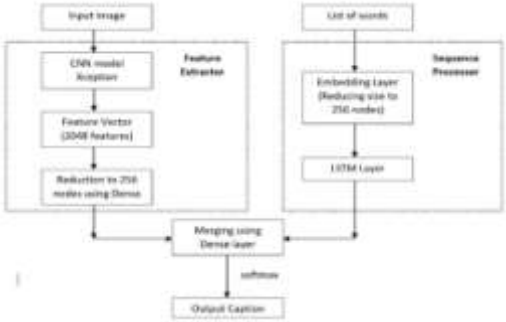

Fig. 2: Block Diagram

Initially, for every image to be captioned an input image is considered which is in a '.jpg' format and there will be a list of words associated to that image. The input image is fed into a feature extractor where CNN model like VGG16 is used for extracting features from the image. The features are stored in the form of vector and using layers like Dense we reduce the nodes to 256. On the other hand, the list of words is given to sequence processor which sends list of words associated to image as input where the embedding layer reduce the size to 256 nodes and then given LSTM layer to produce a more accurate and meaningful sentence to that image. These both outputs from feature extractor and Sequence processor are merged using Dense layer and SoftMax layer is used as activation function before output layer and then gives a meaningful and efficient caption for an image. We built two different models for comparing the efficiency if there is a change in accuracy of caption generated. So, both the models are as follows:

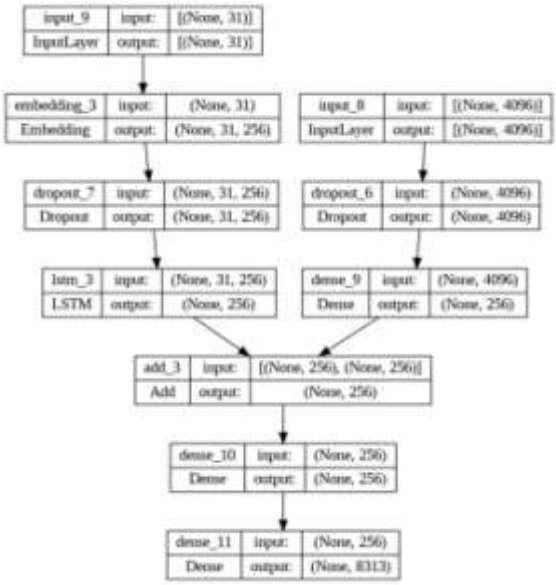

Fig. 3: Bi-Directional Multi-Model Diagram

The technical view of the block diagram above is shown in Figure 3. This Multi-Model Diagram describes the layers present in both image feature extraction and generating captions.

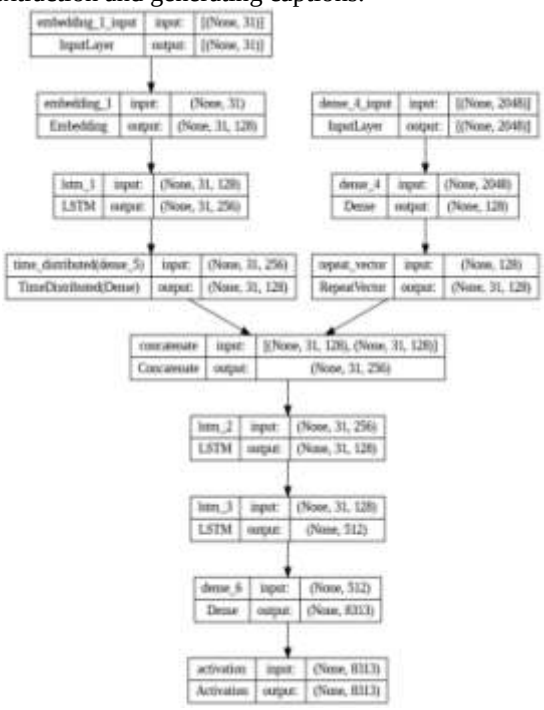

Fig. 4: Bi-Directional TimeDistributed Multi-Model

Diagram

In TimeDistributed Multi-Model we are having a TimeDistributed (Dense layer) in encoder which helps in building accurate captions to an image.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

## IV. IMPLEMENTATION

### A. Knowing about Dataset

The 8,000 images from Kaggle used in this new framework collection for caption-based image description and search have five different explanations, each of which highlights the key aspects and activities in the image. The dataset is known as flickr8k. The images are typically devoid of notable people or things and were hand-selected to get a variety of occasions and situations from six distinct Flickr groups. Images make up Flickr datasets, and the size of the dataset can be selected depending on the system. We have identified Flickr8k and Flickr30k as the two most popular picture caption training datasets in the Computer Vision research field. Each caption annotated image in these datasets, which each comprise 31,000 and 8,000 photos, is tagged with five distinct descriptions. Due to our limited storage and processing capability, we currently choose the Flickr8k dataset, which has the fewest photographs, as our main data source over the other two. Also, as part of our input dataset, we used sanitized Flickr8k data split that Andrej Karpathy open-sourced. This data split has transformed original Flickr8k text data to lowercase, eliminated non-alphanumerical characters and also separated data into train and test subsets.

### B. Importing Necessary Modules

At the beginning, we will load a few libraries, including those required for representing, displaying, and setting up data as well as those compatible with running models. All these modules are utilised at different phases of a project. In this project, each module is important for giving models and building the graphs that we utilize with tqdm. For a future purpose, we also imported models like keras. 1. Numpy: - For mathematical calculations

2. Tqdm: - To make a clever progress indicator for the loops which signifies progress, tqdm library is utilised.
3. Tensorflow: - A Python package containing TensorFlow optimizer implementations for building machine learning models.
4. Keras: - Having an emphasis on contemporary deep learning, a highly productive interface for addressing machine learning challenges.

### C. Extracting Features from an Image

In this part we create a dictionary of images, and we load the image from the file, and we resize the size of the image to 224,224. After resizing the image we convert image into a numpy array. This data should be re-shaped to extract the features to create a model. It is a RGB image so we will have three samples. We Preprocess the image for VGG model and extracted the features present in the image the features extracted for every image and each image is given an image id to store the feature. Every image in dataset undergoes this step and it takes some time to complete. For this we need to use GPU so that many images can be processed at a time instead of one image at a time. Once the features are extracted, we store those features in a pickle file to access the dictionary we created. These features can now be loaded whenever we need them which saves a lot of time. This step uses predefined CNN network model which is very accurate in deigning computer vision projects.

### D. Text Data Preprocessing

We read the text data present in captions.txt file and divide the captions into line by line. After dividing the captions every image is associated with five captions so using the image id, we make mapping dictionary in which keys consist of name of the image removing .jpg extension and value is the list of captions associated to that image. So, for 8091 images each of 5 captions we have 40455 captions which we process. Now the mapping dictionary consists of captions each caption is taken at a time and it is pre-processed. Initially, we convert the captions into lowercase and then replace all the special characters so that while speaking there is no need of using special characters. We also removed the additional spaces in between the words. We used to start and end tags so that it will be easy for the model to know where to start the sentence and where to end.

### E. Create Data Generator Function

The data is now split into train and test for this we used 90 percent of dataset for training the model and the remaining 10 percent for testing the model. We now create a data generator function so that we can load into the model and train it instead of memory. Each time it doesn't occupy memory and it avoids system crash. This caption generator employs an endless loop to analyse each caption one at a time before using Tokenizer to encrypt the sequence. Now we pad the input sequence, and then we encode the output sequence. We will store the output sequence. In this function we create a infinite loop and three lists and we encode the sequence and split the sequences into X, y pairs and pad input sequence and encode the output sequence. These sequences are stored to generate new captions in the model.

A data generator is utilised for both training and validation, and the model is trained utilising the results of these generators. A batch of

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

training/validation data is produced by the data generator function. We are using a generator function because we want to be able to alter the data that goes into the models. It was impractical to use our own data shape, thus a unique output sequence had to be constructed. This is a result of how predictions work. First, the data generator loops through the data batch size times. During each iteration, the data's captions are retrieved, and a new iteration window, this one for the number of captions, is created. Then a sequence variable is defined for each caption. The third and final iteration window is opened, and this time it navigates the 31 length (captions' maximum length) elements. This iteration window will consecutively fill the train input and the label across the LSTM's 31 necessary time steps. An output sequence (out seq), which is a one-hot-encoded vector representing the maximum argument that should be the output of the last layer softmax, is the label, and an input sequence (in seq), which represents the last k words of the caption pre-padded with zeros, are defined and added to a list at each iteration in this window.

**E. Model Creation**

Here we generate layers, initially we create image feature layers with input size of 4096 as suggested in VGG model.

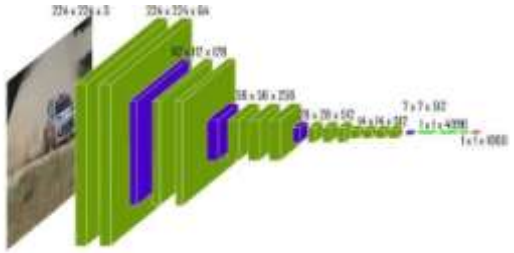

Fig. 5: Overview of VGG16 model layers.

We also create sequence feature layers. In sequence feature layers we use LSTM layer, Dropout layer, Dense layer and others. This is the encoder part coming to decoder in decoder we use dense layers, and the model is created. We set the loss and optimizer as follows:

loss = categorical_crossentropy

Optimizer = adam

$$E_{\text{entropy}} = -\sum_{n}^{N} t_k^n \ln p_k^n$$

Here we train our model by setting parameters like epochs, batch size and etc. As we already created a data generator it will be easy to train the model as it doesn't use much of our memory. Once we train the model, we can save the model. We have created another model by modifying some other layers. Batch normalization and layer normalization are comparable. Batch normalization is frequently used, particularly in CNNs. Even though the input images have already been processed, training causes parameters to change, necessitating a new normalization of the layer inputs. Models are more resistant to a high learning rate and poor parameter initialization with batch normalization. Saturating nonlinearities also gets more difficult. Layer normalization is demonstrated to be superior to batch In RNNs, normalization is simpler to put into practice. Further, it compares favourably to batch normalization in terms of performance RNNs. Layer normalization has been demonstrated to shorten training times while establishing hidden states in RNNs. Within RNN, layer normalization is applied at each time step in the models we use.

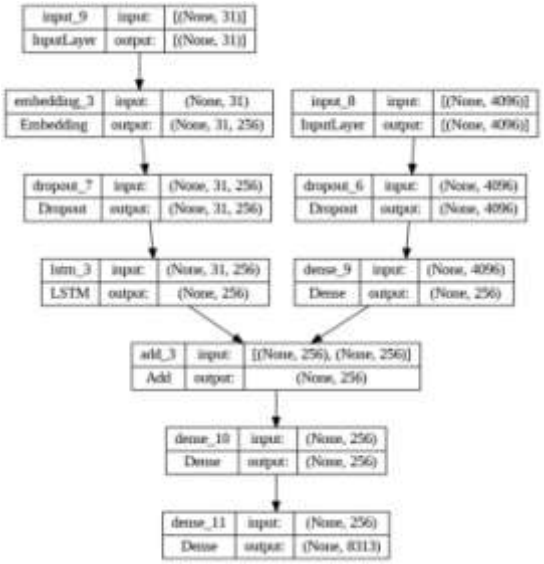


Fig. 6: Bi-Directional Multi-Model Diagram

Encoder:The encoder must extract visual characteristics of different sizes and encode them into vector space so that they can later be given to RNNs. Image encoders that are frequently recommended include VGG-16 and ResNet. We decided to alter the PyTorch library's pre-trained VGG-16 model. Instead of classifying the photos in this job, CNN is utilised to encode the features. As a result, we eliminated the max pool and fully connected layers at the network's end. The output of this new structure has dimensions N X 14X 14 X 512, while the input image matrix has dimensions N X 3 X 256 X 256. Additionally, we extended the CNN architecture by include an adaptive 2D layer to accept input images of different sizes.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

Decoder: The decoder must offer word-by-word visual descriptions using recurrent neural networks (LSTMs), which may output words in succession. The input for the decoder is made up of the encoded image feature vectors from CNN and the encoded picture captions produced during the data pre-processing stage. The decoder comprises of four entirely linked layers provided by the PyTorch library for initialising the states of the LSTMcell and word dictionary, an attention module that we designed and implemented, an LSTM cell module, and an LSTM cell module. When we get encoded photos and captions, we first group them according on how long the key was that was used to encrypt the image.We only wish to analyse the encoded photos whose caption lengths are more than or equal to the number of repetitions to maximize productivity and reduce training time. To construct the attention-masked images with a specific area highlighted, we first input the encoded images and the historical state of the network into the Attention module before the LSTM network iterates. All the captivating images are combined with the embedded captions of the prior words to create the next state of the LSTM. The likelihood of the present word embedding may then be predicted and elevated based on the existing circumstances.The probability of the current word embedding may then be predicted based on the present state and added to the word embedding prediction matrix via totally connected layers.

**F. Generating Captions for Image**

In this step, we create a function for generating a new caption and the steps involved are:

1. Add a start tag for the generation process.
2. Encode input sequence.
3. Pad the sequence.
4. Predict next word.
5. Get index with high probability.
6. Convert index to word.
7. Stop if not found.
8. Append word as input to generate next word.
9. Stop once we reach an end tag.

## V. EVALUATION

Although the pipeline has been put in place, it is not yet evident if it is 100% accurate and capable of providing the desired learning.The next step is to train our model and ensure that it will learn correctly. After the model is accurate, this project still has to finish another important part of it. The best approach to demonstrate the performance of our caption generator is through sentence generation and performance evaluation.

**A. Evaluation metrics**

The BLEU-1, metrics are the most often reported measures for evaluating the calibre of text production in natural language processing jobs (Bilingual Evaluation Understudy). Here, the 1-gram score and the BLEU-1 and BLEU-2 evaluation metrics were important. The next step is to think of more ways to strengthen our model and make it more appropriate for how the articles perform.

**B. Model Performance via BLEU-1 and BLEU-2 score.**

At now, 0.55 is the highest BLEU-1 score we've managed to get in 15 epochs (BLEU-1 is within range 0 to 1). As the BLEU2 score for the flickr8k dataset is 0.33, our model performs worse than that of a state-ofthe-art caption generator. As the criticism on the progress report implies, we opted against spending too much time and effort on simply increasing our BLEU1 score to match the findings in the article because these results may have been altered repeatedly by others and contained undocumented hacks.

Fig. 7: BLEU scores

## VI. RESULTS AND DISCUSSION

As previously indicated, a dictionary of words is used to encode data to integers. After the data has been consumed, the output from the pipeline will also be in the encoded format, which needs to be translated back into human-understandable English phrases. The output of the RNN network is a series of word likelihoods, which is another crucial aspect (likelihoods). Choosing the word with the highest likelihood at each decode stage in an RNN usually produces less-than-ideal results. Instead, we have used LSTM, a well-liked method for determining the best course for interpreting words in natural language. Below are a few captions that were generated for test photographs. Evidently, some of the automatically generated captions from the network exclude crucial details from the image, while others incorrectly identify certain visual elements.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 576-583 www.ijaem.net ISSN: 2395-5252

Fig. 8: Result 1

Object identification and image captioning testing results are shown at this level when the model was trained on a GPU host. An input image is initially fed to VGG16-no-FC, which is utilised to extract image features, in the image captioning step. These characteristics serve as the inputs for an attention mechanism, which extracts a relative range of the targeted area of the objects. Lastly, the LSTM network can be used to produce descriptive sentences. The texts generated can further be converted into speech using our text-to-speech convertors. The speech generated by the converter can be heard by bling people using headphones.

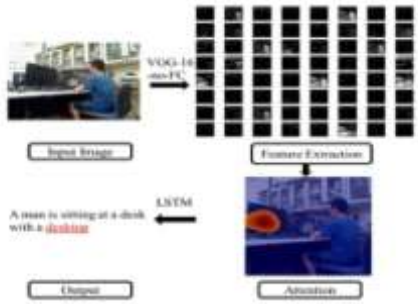


Fig. 9: Output Generation

The procedures of object segmentation and colour analysis make up the processing algorithm for object recognition. The object identification algorithm will first identify the object in the image, along with its range and position. The software then translates the colour data from each segmented pixel in the image into HSV values. By examining an object's colour composition, one can determine the principal object colours. The whole process of object recognition, including object segmentation and colour recognition, is shown in Figure 13. Also, the object identification and picture description algorithms are fed two preliminary detected photos, and the output results are displayed.

Fig. 10: Result 2

We also noticed that the network occasionally fails to separate objects. For instance, two dogs are fighting in the top portion of the photograph above. Nevertheless, two canines are too close together and have sections that appear related in the image for our network to correctly identify them. Thus, the network believes that they are playing with a toy. We found it interesting that captions for test photographs of dogs running were frequently produced extremely properly. This is probably because there are many training images of dogs jumping to catch Frisbees. Additionally, the pretrained CNN can also be very knowledgeable on what a dog looks like.

## V. CONCLUSION

An improved picture captioning model based on encoder-decoder technology is suggested in this work. Images with both distinctive objects and more common objects can use this paradigm. The model may explain the situation contained in an image as well as giving colour to the recognized object, which helps to provide better understanding of the scene. Moreover, colour recognition provides additional information to characterize the traffic light signal, helping those who are visually impaired. Using the already-existing data, we performed the following studies: training and loss performances, evaluation metrics on caption similarity, and qualitative assessments on representative image captions. To see how the network responds in a situation that is completely unrelated to the training and test set, we also tried samples from those sets. These investigations demonstrated to us how this task may be carried out with alternative architectures and models on TensorFlow Keras. Overall, we succeeded in experimenting with several models and learning how various models and alterations effect the network.

Given a smaller dataset, we could only deliver on the project proposal's promises (Flickr8k). Results will be more accurate if there are more epochs and larger datasets. There could be improvements if additional improved datasets are offered. Gains could also result from training on a mix of Flickr8k, Flickr30k, and MSCOCO. The output will generally be more accurate the more diverse the training dataset the network has seen. In the future, it may be thought of expanding the dataset with more data to boost recognition rates. The results of picture captioning might be improved by using a generative adversarial network (GAN) to fill in the recovered item image's backdrop and provide a more accurate description.Also, more focus must be placed on raising the quality of life for people who are blind or visually impaired if deep learning is to benefit everyone.

## REFERENCES

[1]. Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares, "Image Captioning: Transforming Objects into Words", In proceedings of 33rd Conference on Neural Information Processing Systems (NeurIPS 2019),2019.

[2]. Muhammad Abdelhadie Al-Malla, Assef Jafar and Nada Ghneim, "Image captioning model using attention and object features to mimic human image understanding" Al-Malla et al. Journal of Big Data (2022).

[3]. Yeong-Hwa Chang , Yen-Jen Chen , Ren-Hung Huang, "Enhanced Image Captioning with Color Recognition Using Deep Learning Methods" Department of Electrical Engineering, Chang Gung University, Taoyuan City 333, Taiwan, 26 December 2021.

[4]. Dr. A. M. Chandrashekhar, Akash Raj K, Preetham Jain, "Image Captioning using Deep Learning for the Visually Impaired" International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 9 Issue VII July 2021.

[5]. BerfinKav,sut, BesteAydemir, Ege Ozan Özyedek, "Implementation of Various Neural Network Models for Image Captioning" Electrical and Electronics Engineering Department, Bilkent University, EEE 443 Neural Networks Fall 2020-2021 Final Project.

[6]. H. Sharma, M. Agrahari, S. K. Singh, M. Firoj, and R. K. Mishra, "Image captioning: A comprehensive survey," in 2020 International Conference on Power Electronics IoT Applications in Renewable Energy and its Control (PARC), 2020.

[7]. H. Lamba, "Automatic image captioning," https://github.com/hlamba28/ Automatic-Image-Captioning, 2018.

[8]. M. Tanti, A. Gatt, and K. Camilleri, "What is the role of recurrent neural networks (rnns) in an image caption generator?" 08 2017.

[9]. L. Zhou, C. Xu, P. Koch, and J. J. Corso, "Image caption generation with text-conditional semantic attention," arXiv preprint arXiv:1606.04621, 2016.