International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

Image Recognition Using Text and Audio Translation for the Visually Challenged

Rishita Khurana[1], Preeti Manani[2], Nripendra Narayan Das[3], Manika[4], Madhulika[5], Ashish Grover[6], Richa Adlakha[7]

[1]Department of Computer Science and Engineering , Amity University, Noida,India
rishitaakhurana14@gmail.com
[2]Faculty of Education ,Dayalbagh Educational institute, (deemed to be university), Agra
preetimanani.1708@gmail.com
Department of Information Technology
[3]Corresponding Author, Department of Information Technology, Manipal University Jaipur, Rajasthan, India
nripendradas@gmail.com
[4]Department of Computer Science and Engineering , Amity University, Noida,India
manikachoudhary58@gmail.com
[5]Department of Computer Science and Engineering , Amity University, Noida,India
drmadhulikabhatia@gmail.com
[6]Department of Electrical and Electronics Engineering, MRIIRS,Faridabad
Ashi.21s@gmail.com
[7]Department of Electrical and Electronics Engineering, MRIIRS,Faridabad
Richaadlakaha.fet@mriu.edu.in

Abstract—WHO has expressed that out of the general populace on the planet there are 253 million individuals are outwardly impeded around the world. It comes to the standpoint that visually impaired individuals are finding burdensome to curve out their ordinary life. It is vital for take significant measure with the current innovations so they can experience the ongoing scene with next to no troubles. To lift the visually impaired people in the public, this project has been proposed, which can identify images and translates the description of image into text and then produce the audio. This can assist the individual with perusing any text and recognize the image and get the result in vocal structure. Motivated by late work in machine interpretation also, object recognition, a CNN-RNN based attention model is presented in this project. Through the proposed framework, an image is converted into text description first; then, utilizing a basic text-to-speech API, the extracted caption/subtitle is converted into speech which further assists the visually impaired to understand the image or visuals they are looking at. So, the focal part is centered on building the subtitle/text model while the subsequent part, which is changing the text-to-speech, is moderately simple with the text-to-speech API. When the model is fabricated, it is deployed on the local framework utilizing a Flask-based model to produce audio-based caption for any image fed to the model.

Keywords- image recognition; audio translation; blind assistive system; attention-based; artificial intelligence; neural networks; API.

I. INTRODUCTION

One of the main tasks of computer vision, which seeks to automatically provide accurate descriptions for pictures, is image captioning. It necessitates the ability to not only identify key elements in a picture and comprehend how they interact, but also to verbalize those elements in regular language, which is extremely difficult. The attention techniques employed often in today's encoder/decoder frameworks for visual captioning were inspired by the discovery of neural machine translation. For previous methods of captioning photographs, slotted caption templates were constructed. The outputs of object identification, attribute prediction, and scene recognition are utilized to fill in these templates. [1] [2] Deep encoder-decoder architecture, which is employed in current neural-based methods, was developed because of the development of neural machine translation. A CNN may be used in an end-to-end framework to encode the picture into a feature vector, and an LSTM may be used to decode it into a caption. The spatial attention mechanism is employed on the CNN feature map to take the visual environment into consideration. Because the output is directly conditioned on the attention result, the attention mechanism is essential in a system that must capture global dependencies, such as a model for the sequence-to-sequence learning job like image/video captioning. The decoder has only hazy knowledge of the relationship, if any, between the attentions result and the inquiry. When the attention result differs from what the decoder anticipates, the attention module works poorly, or there is no relevant information from the candidate vectors at all, the decoder may be forced to provide false results. Since mistakes inevitably occur, the former situation cannot be prevented. Some models introduce an adaptive attention technique to control the timing of visual

***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

attention. [3] Writing captions for pictures is obviously a chore that is very important to scene collecting, which is one of the main purposes of PC vision. The age models must not only be able to handle the PC vision challenges of identifying the things in a picture, but they must also be capable of comprehending and expressing their connections in common language. [4] Age of the subtitles has therefore been considered a problematic issue for a while. It is difficult for AI and artificial intelligence research to reproduce the amazing human capacity to pack a tone of notable visual information into illustrative language. Intermittent neural networks are then used to translate those representations into English language words. Convolutional neural networks (convnets) are utilized to produce vectorial representations of pictures. [5] [6] The availability of huge characterization datasets and advancements in the construction of deep neural networks has both been helpful in our study, which has mostly concentrated on the characteristics of inscription age. The existence of contemplation is one of the most intriguing aspects of the human visual framework. Consideration allows for stunning highlights to gradually move to the front when needed, as opposed to cramming the entire picture into a static depiction.

When there is a lot of clutter in an image, this is essential. An effective method that has been widely used in earlier work is the use of representations (such as those from the very top layer of a convnet) that condense the information in an image to its most striking aspects. Unfortunately, doing so can result in the loss of information that would make for richer, more interesting subtitles. Using a lower-level representation allows the preservation of this information. When there is a lot of clutter in an image, this is essential. An effective method that has been widely used in earlier work is the use of representations for example, those from a convnets very top tier that condense the information in an image to its most striking aspects. Numerous parameters, including the preprocessing approach, deep learning methodology, dropout use, and image classifier, were investigated to understand their effects on captioning.

1) Dataset: For this job, one public dataset based on Flickr8K was identified. The original Flickr8K, however, offers five captions for each picture. The two examples from this dataset are shown in Fig. 1.

2) Image feature extraction: For this job, building CNN is frequently used, although it needs a large dataset and powerful processing. Inception V3 is employed, which offers a trained model that is well-optimized and usable even without preprocessing, was also used in our study.

3) Data collection and preprocessing: It is necessary to pre-process the data before it is fed into the network since it may contain redundant data and irrelevant data. Excess of data also leads to Overfitting of the model. All occurrences of single characters should be removed to get good results. We also chose to correct the misspelling as a result.

4) Models: Two deep learning algorithms (CNN [8] and LSTM [9]), two image classifiers, and four preprocessing methods were used in tests to build the models. They were contrasted based on how they performed.

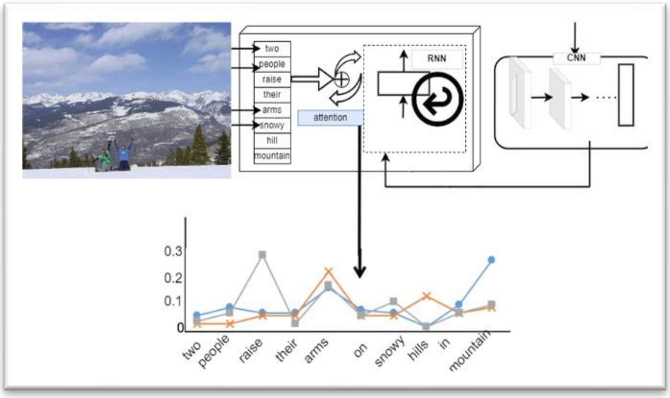


Figure 1. Proposed framework

5) Evaluation: The accuracy of photo captioning and translation in several languages is compared using the Bilingual Evaluation Understudy (BLEU) scale. To compare the effects of each, the understudy factors BLEU-1, BLEU-2, BLEU-3, and BLEU-4 are used.

The following are some of this project’s contributions:

- A CNN-RNN based attention image caption generator is proposed.
- By visualizing "where" and "what" the attention was focused on, it is demonstrated how understanding may be achieved and how the outcomes of this framework are perceived.
- On three benchmark datasets, including Flickr8k, the value of attention in caption production is confirmed.
- The generated captions are converted into speech using an API.
- The generated captions therefore benefit the visually impaired in understanding the image.
- The model is finally deployed on the local machine using Flask.

II. NEED OF THE STUDY

On our world, there are 7.4 billion individuals, of which 285 million are visually impaired (WHO, 2011). Of them, 39 million are totally blind, meaning they have no vision at all. According to estimates, there will be 200 million individuals who are visually impaired and 75 million blind people worldwide by the year 2020. Reading is essential to everyday life for most people, and text may be found in newspapers, consumer goods, signboards, computer displays, and other places. As a result, those who are blind or visually impaired have several challenges. Numerous improvements have been made in this section to make it easier for those who are blind to read. The main issue that visually impaired individuals now face is that they cannot do text recognition on their own, which forces them to rely on others for their daily activities like reading newspapers, responding to letters that are received through the mail, referring to books, etc. As they couldn't handle it on their own, this issue can make them less confident. The project's main objective is to assist these persons with visual impairments in text recognition. This goal is accomplished by creating a model that, after creating a caption from a picture, delivers the text aloud using the given speaker or headphones.

The development of technology for handheld devices and broadband wireless devices has several uses in image captioning. Instantaneous information may be found online. Most of this data is presented in integrated multimedia presentations or as text, graphics, and images that are meant to be read visually. Image captioning is the process of improving, manipulating, or analyzing digital image data algorithmically (which can also include comprehension or recognition). An image, such as an image or a frame of video, is used as the input for image captioning, which is a type of signal captioning. An image captioning could generate a visual, a set of measurements or features corresponding to the image, or both. Most image-captioning methods considers the images a two-dimensional signal and then processes it using common signal-captioning methods. Acquiring images is the process of imaging. The topic of image captioning is photographs that are two-dimensional objects that have been electronically collected, such as scanned office papers, satellite shots, x-ray images, etc. The use of video image captioning to resolve issues with real-time road traffic management systems is becoming more and more significant. This directly highlights the projected advancements in digital video camera technology in the future. Understanding the nuances of picture captioning and the range of applications that the technology will be utilized in the future will be helpful for planning in this important field. With increasing applications in all sectors of business, image captioning is one of the areas of information technology that is now advancing the fastest. This technology offers the promise of creating the ultimate device capable of taking on human visual functions in the future.

The discipline of image captioning serves as the foundation for all future forms of visual automation. A product is visually sorted using light sensors using sophisticated optical sorting systems, which employ image captioning to distinguish between an object's colors. The term "augmented reality" refers to a live, direct or indirect view of a physical, real-world environment whose elements are blended with (or improved by) virtual, computer-generated pictures. Thus, a confused reality is produced. The augmentation often takes place in real-time, such as when a sporting event is being shown on television. Research on augmented reality looks at the use of computer-generated images in real-time television broadcasts to enlarge the real-world. The term "feature detection" refers to techniques that compute abstract representations of image data and locally determine if a certain type of image feature is present at each image location. Many computer vision algorithms start with features as their building blocks. Repeatability is the desired characteristic for a feature detector. The ability to identify the same feature in two or more photographs of the same scene will be crucial. [10][11][12] The search for extraterrestrial intelligent life will be a component of image captioning in the future. New intelligent, digital species that were entirely generated by research professionals throughout the world will also arise because of the development of picture captioning technology. Millions of millions of robots will be present on the planet in a few decades, revolutionizing how society is organized. This will be the result of developments in image captioning and associated technologies. [13] Thanks to developments in image captioning and artificial intelligence, it is now possible to speak commands, anticipate government information needs, translate languages, recognize, and track people and things, diagnose medical conditions, perform surgery, reprogram human DNA flaws, and automate all types of transportation. With modern computing increasing complexity and power, the concept of computation can be expanded beyond its existing limitations. The ability to replicate the human visual system will be possible in the future as picture captioning technology advances. Graphics data is becoming more important in applications for picture captioning. However, the fully linked neural network continues to be an impractical substitute, but the cellular neural network has developed into a paradigm for future imaging approaches. This project offers the option to listen to the text and view the photos by listening to the audios, making it accessible to those who have mild to moderate visual impairment. For those with dyslexia or other learning problems that make it difficult to read or understand words and letters, it can also serve as a learning assistance. Since they won't require aid to interpret written language anymore, we want to help these folks become independent and self-sufficient. Such people would never feel disadvantaged since they will always have access to knowledge. The system's technical advancement and application will have a revolutionary positive influence on current civilization.

III. RESEARCH METHODOLOGY

The two attention-based model versions are discussed in this section by first discussing the data collection, pre-processing, and cleaning, followed by their shared framework. The definition of the function, which is covered in depth in the sections that follow, is the main distinction.

A. Data and sources of data

The dataset used in this project - Flickr8K, has been taken from Kaggle and is modified according to the requirement

***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

of the model proposed in this project. The steps used in modifying the dataset are mentioned below.

Apply train_test_split on both image path & captions to create the train & test list.

- Create a function which maps the image path to their feature.
- Create a builder function to create train & test dataset & transform the dataset
- Load the pretrained Imagenet weights of Inception net V3
- Shuffle and batch while building the dataset
- Ensure shape of each image in the dataset : (batch_size, 8*8, 2048)
- Ensure shape of each caption in the dataset : (batch_size, max_len)

The training and testing split ratio is 80:20 with random state = 42. Table 1 shows total number of data used for different category: - training images, testing images, training captions and testing captions.

Table 1. Total amount of data.

Category of data	*Total data*
Training images	32364
Testing images	8091
Training captions	32364
Testing captions	8091

It was necessary to provide vast amounts of data in a way that was straightforward to acquire and understand. Organizations produce data daily. As a result, there is currently a huge amount of data on the Internet. This enormous volume of data is difficult for users to navigate, comprehend, and use. Data visualization skills are essential for scientific research. Today, processing enormous volumes of data is possible thanks to computers. The design, creation, and use of computer-generated graphics for the display of data are all included in data visualization. [14] [15] Data efficiently represents information from several sources. Decision makers may therefore examine analytics in a visual way, making it simple for them to analyze the data. The terms scientific visualization and information visualization are frequently used to describe data visualization. Their ability to recognize patterns, understand information, and develop opinions is all aided by it. To guarantee that messages or information survive throughout time, humans have used visualization techniques for a very long period. Things that cannot be touched, smelled, or tasted can be graphically represented. In this project, the top 30 occurring words in the captions are visualized using a bar chart. Figure 2 shows the bar plot of top 30 occurring words.

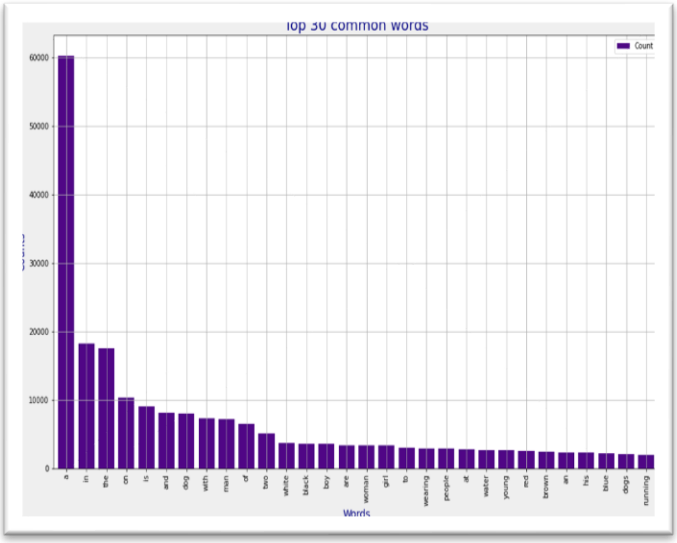


Figure 2. Bar plot showing the top 30 occurring words.

Through this bar plot, it is quite clear that stop words form a majority. Stop words are basically the set of most common words in a language. To get rid of phrases that are used so frequently that they don't really contain any useful information, they are commonly utilized in text mining and natural language processing (NLP). The top 30 stop words with the highest frequency are shown in Figure 3.

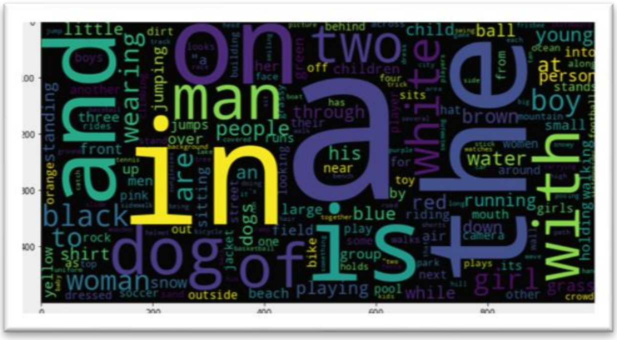
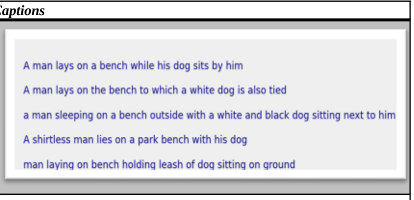
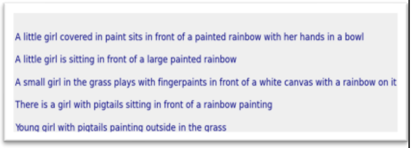
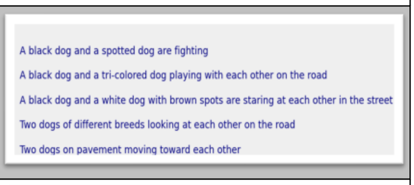
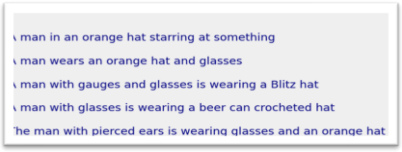


Figure 3. Frequency of top 30 words.

Data visualization uses computer visuals to display connections, trends, and patterns among various data objects. With only a few mouse clicks and pull-down menus, it can create pie charts, bar charts, scatter plots, and other forms of data visualization. When making sorts of visualization, colors are carefully chosen. The colors that are used to display data must make it obvious how the various data components vary from one another. Data is distilled and abstracted in data visualization. The data's spatial qualities, such as position, size, and shape, are crucial components. The original dataset should be modified, compressed, and shown on a screen using a visualization system. Results need to be communicated clearly and graphically using graphs and charts. [16] [17]

The captions and images are visualized together in this project to get a clear picture of how the model is going to work and helps in determining the areas to concentrate. Table 2 visualizes the images and captions that can be predicted.

Table 2. Visualization of images and captions.

Images	***Captions***
	 A man lays on a bench while his dog sits by him A man lays on the bench to which a white dog is also tied a man sleeping on a bench outside with a white and black dog sitting next to him A shirtless man lies on a park bench with his dog man laying on bench holding leash of dog sitting on ground
	 A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl A little girl is sitting in front of a large painted rainbow A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it There is a girl with pigtails sitting in front of a rainbow painting Young girl with pigtails painting outside in the grass
	 A black dog and a spotted dog are fighting A black dog and a tri-colored dog playing with each other on the road A black dog and a white dog with brown spots are staring at each other in the street Two dogs of different breeds looking at each other on the road Two dogs on pavement moving toward each other
	 A man in an orange hat starring at something A man wears an orange hat and glasses A man with gauges and glasses is wearing a Blitz hat A man with glasses is wearing a beer can crocheted hat The man with pierced ears is wearing glasses and an orange hat

Most visualization designs are intended to improve cognition and support decision-making. It's crucial to consider the intended application of data visualization while creating a prototype. Data visualization calls for the selection and reevaluation of the numerical data points in addition to just reporting the statistics. Data visualization is a crucial area in computer science with many potential applications. [18]

The collection of techniques used before deploying a data mining strategy is collectively referred to as "data pre-processing for data mining," and it is acknowledged as one of the most critical challenges within the well-known Knowledge Discovery from Data process. It is not immediately applicable to begin a data mining process with inconsistent or redundant data because flawed data will almost certainly exist. The velocity and size of data generation, which is expanding quickly in business, industrial, academic, and scientific applications, must also be mentioned. More complex techniques are needed to analyze the larger volumes of data that are collected. [19] Data pre-processing allows for the adaptation of the data to the specifications of each data mining technique, making it possible to handle data that would otherwise be infeasible. Even while information preparation is a valuable tool that can enable the client to accept and treat complicated information, it may take a long time to handle. It combines several disciplines, including information readiness and information reduction techniques. While the latter seeks to reduce the complexity of the information by includes choice, occurrence determination, or discretization, the former involves information modification, mix, cleansing, and standardization. [20] [21] The final informative collection obtained may be seen as a trustworthy and reasonable hotspot for any information mining computation conducted later with the use of an efficient information preprocessing step. [22] The captions and images, both are preprocessed before feeding them into the network. The steps used for preprocessing the captions are as follows: -

- By tokenizing the captions, for instance: - separating those using spaces & other filters, the tokenized vectors were created. This provided a lexicon of every distinct word in the data. Maintained a vocabulary of at least 5,000 words to preserve recall.
- The unknowable token "UNK" was used in lieu of all other words.
- Developed mappings from words to indexes and vice versa.
- All sequences were padded to match the length of the longest one.

Each individual image has five captions which are all different lengths**.** Hence, they can't be fed directly to the Decoder. Before continuing, padding is utilized to make sure that each caption is a specific length. It is a process that we apply at the start or the end of a sequence to make all the samples to have a common standard length. For text generation, the decoder needs to have an input at the start, and it can't be a padded input. Therefore, for this process, padding is done at the end of the caption sequence. All padded values will be masked and set to 'False' while the rest will be set to 'True'**.** All True values are assigned the value: 1, while padded, i.e., false values are set to 0**.** The danger of introducing penalty to the model increases with padding. Therefore, masking is used to fix the issue, and this reduces all extra penalties back to zero.

Data pre-processing isn't simply restricted to traditional information mining undertakings, as characterization or relapse. An ever-increasing number of scientists in clever information mining fields are giving progressively consideration to information data preprocessing as a device to work on their models. This more extensive reception of information preprocessing methods is bringing about variations of known models for related systems, or totally clever recommendations.

In the accompanying we will introduce the primary fields of information pre-handling, gathering them by their sorts and showing the ongoing open provokes comparative with everyone. In the first place, we will handle the preprocessing methods to manage blemished information, where missing qualities and commotion information are incorporated. Then, information decrease preprocessing approaches will be introduced, in which highlight choice and space change are shown. The accompanying segment will manage occurrence decrease calculations, including example determination and model age. The last three segments will be committed to discretization, resampling for imbalanced issues and information pre-handling in new fields of information mining separately.

The images are also preprocessed by using the following steps:

- Resized the images into the shape of (299, 299)
- The image was normalized between -1 and 1, putting it in the proper format for InceptionV3.

Since there was a list which contained the entire image path, it was needed to first convert them to a dataset using *tf.data.Dataset.from_tensor_slices*. Once this was done, a dataset consisting of image paths was created; and a function was applied to the dataset which applied all the necessary preprocessing to each image. This function resized them and did the necessary preprocessing that it converted the images into correct format for InceptionV3. Figure 4 shows the resized images. Shape of images after resizing is (299, 299, and 3)

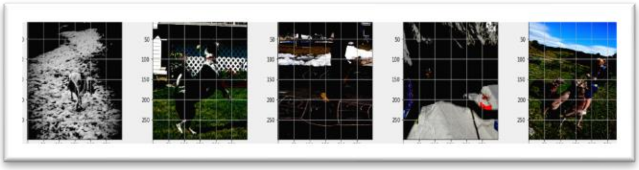


Figure 4. Resized images.

B. *Captions pre-processing*

For the past ten years, inscribing of images has been a significant problem, with a lot of efforts concentrated on English subtitling. The number of images uploaded to the internet has increased because to online entertainment. In June 2019, Facebook received 300 million photos daily, compared to 95 million for Instagram. A test for automated subtitling of photos to seek for photographs by pleased or by human language, as well as regarding video setting depictions, has also been created because of the arrival of dazzling devices and cameras out in the open places. [23] Since it begins by discriminating and recognizing things, connects these items, and then translates these recognized objects into text that is intelligible to people by utilizing their language's sentence structure and semantics, image captioning (IC) needs a lot of labor. Deep learning techniques were used to solve these challenges, and the work paid off with a satisfactory outcome. [24] [25] [26]

A four-part convolution structure for image subtitling was offered by the inventor. It starts with inserting a layer for the information message, moves on to inserting the information picture, and then adds the convolution model before installing the yield age. On the challenging MSCOCO dataset, an evaluation of the LSTM model is conducted. [27]

Another investigation was conducted with the goal of considering a feed-forward network that can operate across all words equally, and the results outperformed the benchmark LSTM model. An innovative method for picture subtitling was developed for the subtitle age, and the advantage in the model was to keep previous visual content. This method used visual location connections, chart brain organization, and establishing mindful consideration system. [28] The model was developed and tested using the MSCOCO and Flickr30K datasets. According to the model's developers, the findings showed how this model can outperform cutting-edge consideration-based techniques.

C. *Theoretical framework*

The proposed model incorporates CNN which serves the purpose of an encoder and RNN that works as a decoder. The quick ascent in the field of Profound Learning and PC Vision is changing the way in which we approach significant issues. This paper aims to support the visually disabled with image acknowledgment by utilizing a novel model that uses prevalent techniques for the image subtitling and makes an interpretation of the created inscriptions into discourse. The model was executed utilizing two methodologies - an encoder-decoder system. The exhibition of the two models was assessed utilizing the BLEU score metric.

The past few years have seen significant advancements in PC vision in the picture handling area, including object detection and picture organization. With the help of picture characterization and article recognition advancements, it is now possible to spontaneously generate at least one phrase to understand the visual content of a picture, a problem known as picture inscription. Making complete and consistent picture representations naturally opens a wide range of potential applications, such as titles for news images, illustrations for clinical images, text-based picture recovery, data access for clients who are blind, and human-robot collaboration. These picture subtitling applications offer a high theoretical and practical evaluation value. Thus, in the age of artificial thinking, image subtitling is a more complicated yet important task. When given another image, a picture-inscribing calculation should provide a semantic representation of that image. For instance, in Figure 1, the informational image includes people, sheets, and waves. [29] The picture's content is described in a line at the bottom of the page; this sentence unmistakably captures the setting, the action, and the objects that appear in the image. People can easily understand the content of a picture and translate it into sentences in regular language when it comes to picture subtitling, but computers must integrate the use of picture handling, PC vision, regular language handling, and other significant areas of exploration results. To put picture inscription to the test, create a model that can fully use image data and provide richer, more lifelike portrayals of images. The significant portrayal age cycle of undisputed level picture semantics demands not only the ability to understand objects or

***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

scenes in the picture but also the capacity to analyze their states, determine how they relate to one another, and create a semantically and linguistically sound sentence. How the mind creates an image and organizes the visual information into an inscription is yet unknown. A deep understanding of the universe and which items are outstanding components of the whole is required for picture inscription. Roused by AI's encoder-decoder engineering, late years most picture subtitling strategies utilize a Convolutional Neural Network (CNN) as the encoder and a Recurrent Neural Network (RNN) as the decoder, particularly Long-term short memory (LSTM) to create inscriptions, with the goal to boost the probability of a sentence given the visual elements of a picture. A few strategies are involving CNN as the decoder and the support advancing as the dynamic organization. As per these different encoding and disentangling strategies, in this paper, we partition the picture subtitling techniques with brain networks into three classifications: CNN-RNN based, CNN based and support-based system for picture inscribing. In the following part, we will discuss their primary thoughts. [30]

D. CNN-RNN Framework

This section elaborates the proper model framework which is being used to forward the study from data towards inferences. The detail of methodology is given as follows. An image comprises of various varieties to create the various scenes in the eyes of a human. Be that as it may, in the perspective on PC, most pictures are painted with pixels in three channels. In any case, in the brain organization, various modalities of information are moving to make a vector and do the accompanying procedure on these elements. It has been convincingly demonstrated the way that CNNs can deliver a rich portrayal of the information picture by implanting it into a fixed-length vector, with the end goal that this portrayal can be utilized for an assortment of vision undertakings like article acknowledgment, discovery, and division. Consequently, picture subtitling strategies considering encoder-decoder structures frequently utilize a CNN as a picture encoder. The RNN network gets verifiable data through nonstop flow of the secret layer, which has better preparation capacities and can perform better compared to mining further etymological information, for example, semantics and language structure data implied in the word arrangement. [31] For a reliance connection between various area words in verifiable data, a repetitive brain organization can be effectively addressed in the secret layer state. In picture subtitling task considering encoder-decoder structure, the encoder part is a CNN model for separating picture highlights. It can utilize models, for example, AlexNet, VGG, GoogleNet and ResNet.

CNNs are a subclass of deep learning models widely used for analyzing visual images. CNNs are able to recognize and rank certain components from images. Their uses include video and picture acknowledgement, picture ordering, clinical image analysis, computer vision, and handling of everyday language. In CNN, the word "convolution" refers to the numerical capacity of convolution, a special kind of direct activity in which two capabilities are combined to produce a third capability that conveys how the state of one capability is altered by the other. Simply put, two photos that may be used as grids are copied to get a result that is used to segregate items from the picture. CNN engineering consists of two main components:

- A component extraction cycle in a convolutional device that isolates and separates the various visual highlights for analysis.
- The organization of element extraction comprises of many sets of convolutional or pooling layers.
- A completely associated layer that uses the result from the convolution cycle and predicts the class of the picture considering the highlights separated in past stages.
- This CNN model of element extraction plans to diminish the quantity of highlights present in a dataset. It makes new elements which sums up the current highlights contained in a unique arrangement of elements. There are numerous CNN layers as displayed in the CNN design outline.

The model takes a solitary crude picture and creates an inscription y encoded as a grouping of 1-of-K encoded words.
y = {y1, . . ., yc}, y E Rk; where K is the size of the jargon and C is the length of the inscription. We utilize a convolutional brain network to extricate a bunch of element vectors which we allude to as comment vectors.
The extractor produces L vectors, every one of which is a D-layered portrayal comparing to a piece of the picture.
a = {a1. . . ac}, man-made intelligence E Rd;
To get a correspondence between the component vectors and bits of the 2-D picture, we separate elements from a lower convolutional layer not at all like past work which rather utilized a completely associated layer. This permits the decoder to zero in on specific pieces of a picture by weighting a subset of all the element vectors specifically.

The word vector articulation is fed into the RNN model in the decoder section of the system. Each word is initially addressed by a one-hot vector before becoming comparable to the picture's inclusion using the word inserting model. It is possible to think about the problem with picture inscription as a binary (I, S), where I refer to a diagram, S is a collection of target words (S = "S1," "S2"), and Si is a component of the informative collection extraction. Preparation's main goal is to increase the likelihood evaluation of the objective portrayal's (S|I) for the generated assertion's aim and its objective assertion matching even more closely. To address the picture inscribing issue, Mao presented a multimodal Intermittent Brain Network (m-RNN) model that creatively combines the CNN and RNN model. The LSTM model is a unique type of RNN model building that can address the challenges because standard RNN suffers from slope vanishing and limited memory problems. Three more control units (cells), including the information, yield, and ignored entryways, are added. The cells in the model will make decisions about the data as it is introduced. Nonconforming data will be ignored in favor of data that complies with the requirements. This standard allows for the resolution of the lengthy arrangement dependency problem in the organization of the brain. Vinyals introduced the NIC (Brain Picture Subtitle) model, which receives an image as input to the encoder section and generates the corresponding representations using LSTM networks in the decoder part. [32]

As a decoder, a recurrent neural network (RNN) is used. A single word is produced at each time step by the decoder and is modeled on a setting vector, the previous secret state, and the most recently formed words to build a subtitle. It is a type of fake brain network that makes use of time series data or sequential data. These sophisticated learning calculations are frequently applied to mundane or practical problems, such as language interpretation, normal language handling (NLP), discourse acknowledgment, and picture subtitling; they are built into well-known programmed like SIRI, voice search, and Google Decipher. Repetitive brain networks prepare data for learning, much as feed forward and convolutional neural networks (CNNs). As they use knowledge from prior contributions to influence the current information and outcome, they are identified by their "memory." The outcome of intermittent brain networks depends on the previous components in the succession, in contrast to conventional deep brain networks, which assume that information sources and results are independent of one another. Unidirectional repeating brain networks cannot capture future events in their predictions, even if they would be helpful in determining the outcome of a particular sequence. Recurrent influence back proliferation through time (BPTT) computation is used to determine angles, however it differs slightly from conventional back spread since sequencing data is explicit. The BPTT standards are analogous to traditional back spread, where the model learns itself by identifying errors from its result layer to its feedback layer. We can modify and fit the model's bounds using these computations. While feed forward networks don't need to aggregate errors since they don't share boundaries across each layer, BPTT differs from standard approach in that it collects errors at each time step. Detonating slopes and vanishing inclinations are two problems that RNNs frequently encounter because of this interaction. These problems are distinguished by the magnitude of the angle, which is the slope of the unluckiness capacity along the error bend. When the inclination is too small, it keeps on getting smaller, readjusting the weight limits until they are irrelevant, like 0. The computation is finished learning when that occurs. When the slope is too steep, an unstable model is created, detonating inclinations take place. In this case, the model loads will get overly large and will eventually be treated as NaN. Reduced hidden layers inside the brain organization, which eliminate some of the RNN model's complexity, is one way to address these problems. The model tackles the issue of vectorization of normal language sentences well indeed. It is of extraordinary importance to utilize PCs managing regular language, which makes the handling of PCs no longer stays at the straightforward degree of coordinating, yet further to the degree of semantic comprehension. Propelled by the brain network-based machine interpretation system, the consideration component in the field of PC vision is proposed to advance the arrangement among words and picture blocks. Accordingly, during the time spent sentence age, the "consideration" move cycle of reenacting human vision can be commonly advanced with the age cycle of the word arrangement, so the created sentence is more in accordance with individuals' appearance propensity. Rather than encoding the entire picture as a static vector, the consideration component adds the entire and spatial data relating to the picture to the extraction of the picture highlights, bringing about a more extravagant proclamation portrayal. [33] [34] The main consideration component was proposed in, it proposed the "delicate consideration" and that means to choose districts in view of various loads and the "hard consideration" which performs consideration on a specific visual idea. The exploratory outcomes got by utilizing consideration based profound brain networks have accomplished exceptional outcomes.

E. Evaluation metrics

To evaluate the benefits and drawbacks of the age findings, the continuing focus primarily makes use of the degree of correspondence between the inscription sentence and the reference sentence. BLEU, ROUGE, METEOR, SPICE, and CIDER are among the frequently used approaches. ROUGE is derived via text analysis, whereas CIDER and SPICE are explicit pointers considering picture subtitling. BLEU and METEOR are obtained by machine interpretation. [35] [36] [37]

- The evaluation of photo comment findings, which rely on n-gram accuracy, often uses BLEU. The BLEU rule states that the distance between the assessed and reference sentences must be determined. When the subtitle is closest to the length of the reference articulation, BLEU method will often provide a higher grade.
- Text outline calculations are the focus of the scheduled assessment standard ROUGE. ROUGE-N, ROUGE-L, and ROUGE-S are the three assessment rules.

***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

ROUGE-N computes a basic n-tuple review for all reference explanations, depending on the sentence that is being evaluated: ROUGE-L depends on LCS, the largest normal succession, to complete the evaluation. Considering co-occurrence measurements of the skip-bigram between the depiction of the reference text and the forecast text, ROUGE-S determines the review.

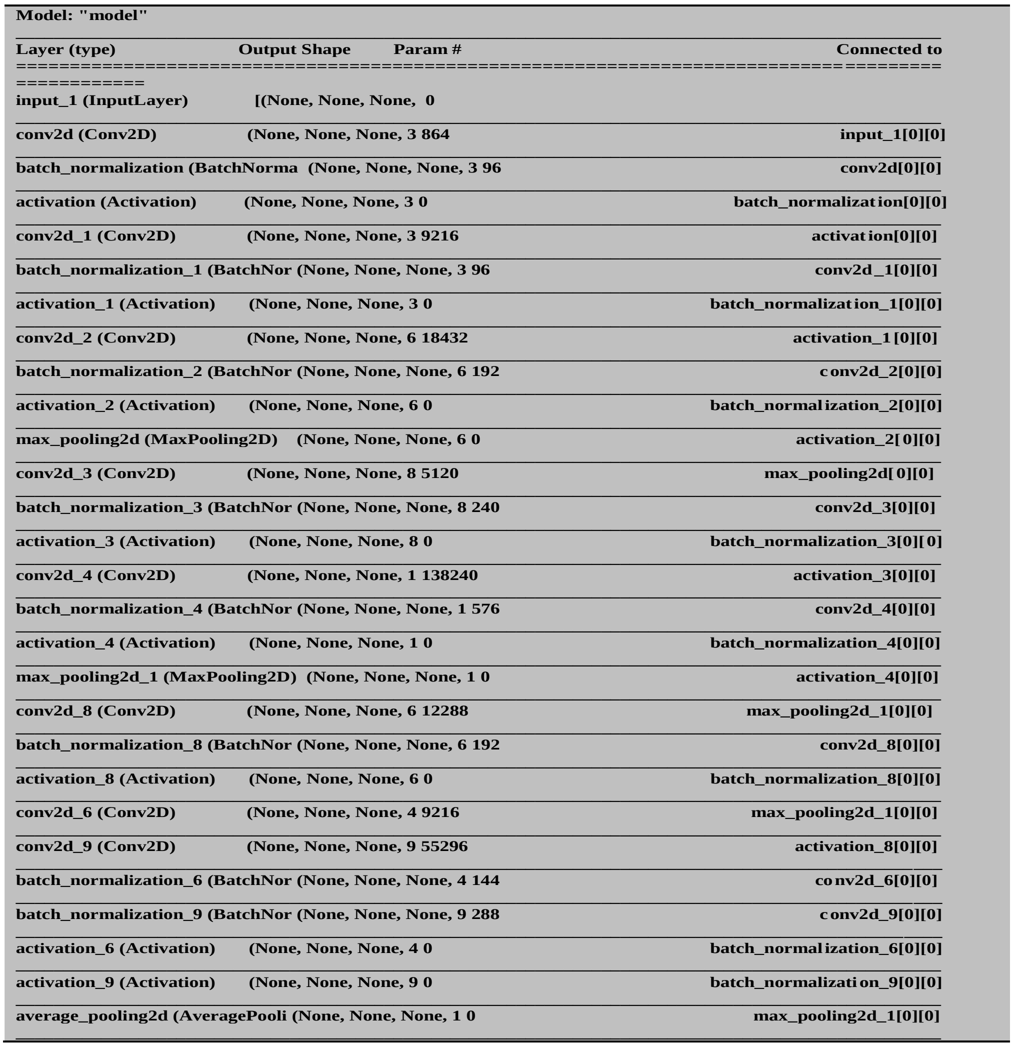
- CIDEr is a remarkable method that accommodates image inscription job. By using a term recurrence converse record recurrence (tf-idf) for each n-gram, it calculates the degree of agreement in image subtitling. According to studies, the CIDEr-human agreement match is more accurate than other assessment metrics.
- METEOR is reliant on the consonant mean of unigram exactness and review, although the review's weight is greater than its exactness. It differs from the BLEU in that it is present not only in the entire set but also at the sentence and division levels, and it has a strong correlation with human judgment. It is deeply relevant to human judgment.
- By converting the resultant representation sentences and reference sentences into chart-based semantic depictions, more specifically "scene diagrams," SPICE analyses the nature of picture captions. [38]

F. *Model building*

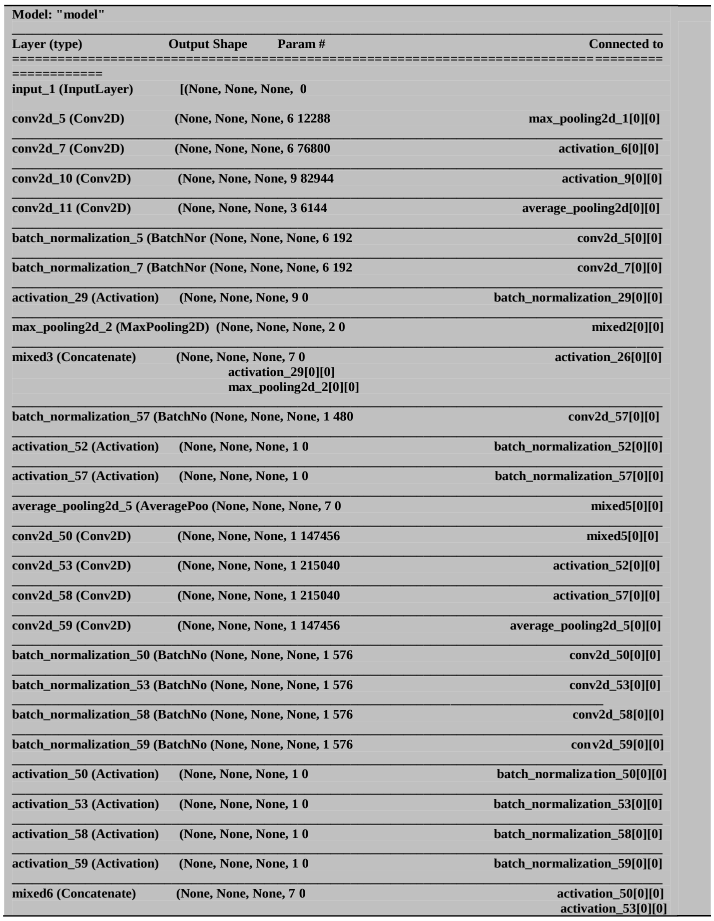
The basic steps used in building the model are as follows: -

- Set model parameters (units, embed_dim, vocab_size, train & test num steps, feature_shape, max_length)
- Build Encoder
- Build Attention Model
- Build Decoder

Figure 5 shows the summary of the model.

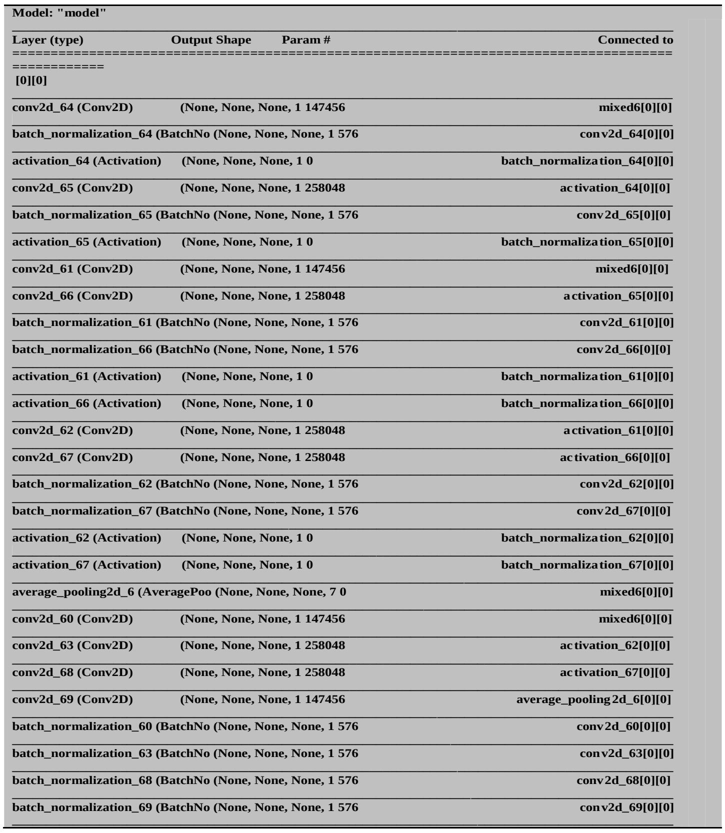


```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #                          Connected to
==================================================================================================
============
input_1 (InputLayer)            [(None, None, None,  0
__________________________________________________________________________________________________
conv2d (Conv2D)                 (None, None, None, 3 864                              input_1[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma  (None, None, None, 3 96                              conv2d[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, None, None, 3 0                  batch_normalization[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, None, None, 3 9216                             activation[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, None, None, 3 96                               conv2d_1[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, None, None, 3 0                  batch_normalization_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, None, None, 6 18432                            activation_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, None, None, 6 192                              conv2d_2[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, None, None, 6 0                  batch_normalization_2[0][0]
__________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)    (None, None, None, 6 0                            activation_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, None, None, 8 5120                         max_pooling2d[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, None, None, 8 240                              conv2d_3[0][0]
__________________________________________________________________________________________________
activation_3 (Activation)       (None, None, None, 8 0                  batch_normalization_3[0][0]
__________________________________________________________________________________________________
conv2d_4 (Conv2D)               (None, None, None, 1 138240                           activation_3[0][0]
__________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, None, None, 1 576                              conv2d_4[0][0]
__________________________________________________________________________________________________
activation_4 (Activation)       (None, None, None, 1 0                  batch_normalization_4[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, None, None, 1 0                            activation_4[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, None, None, 6 12288                      max_pooling2d_1[0][0]
__________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, None, None, 6 192                              conv2d_8[0][0]
__________________________________________________________________________________________________
activation_8 (Activation)       (None, None, None, 6 0                  batch_normalization_8[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, None, None, 4 9216                       max_pooling2d_1[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, None, None, 9 55296                            activation_8[0][0]
__________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, None, None, 4 144                              conv2d_6[0][0]
__________________________________________________________________________________________________
batch_normalization_9 (BatchNor (None, None, None, 9 288                              conv2d_9[0][0]
__________________________________________________________________________________________________
activation_6 (Activation)       (None, None, None, 4 0                  batch_normalization_6[0][0]
__________________________________________________________________________________________________
activation_9 (Activation)       (None, None, None, 9 0                  batch_normalization_9[0][0]
__________________________________________________________________________________________________
average_pooling2d (AveragePooli (None, None, None, 1 0                          max_pooling2d_1[0][0]
__________________________________________________________________________________________________
```

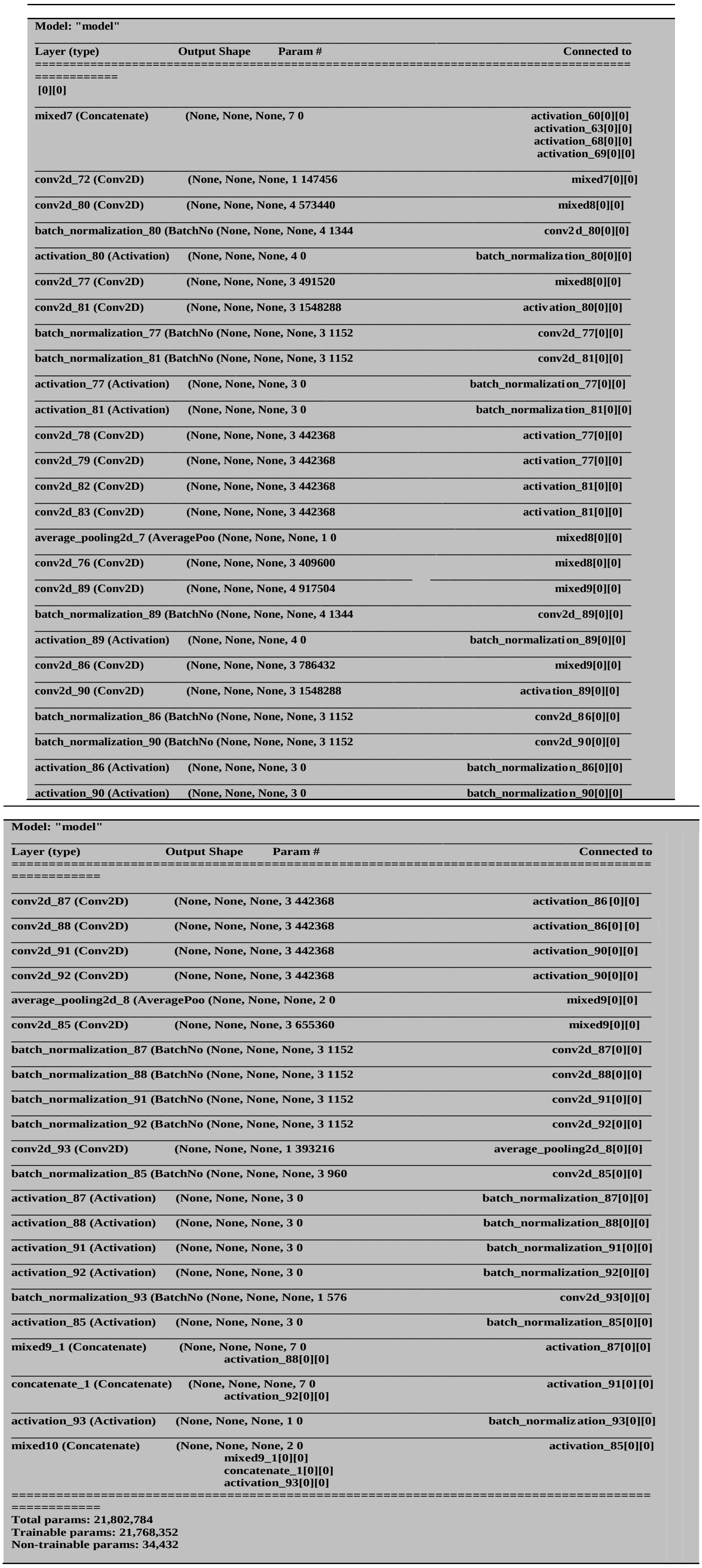


```
Model: "model"
_________________________________________________________________________________________
Layer (type)                Output Shape        Param #                      Connected to
=========================================================================================
============
input_1 (InputLayer)        [(None, None, None,  0

conv2d_5 (Conv2D)           (None, None, None, 6 12288            max_pooling2d_1[0][0]
_________________________________________________________________________________________
conv2d_7 (Conv2D)           (None, None, None, 6 76800                 activation_6[0][0]
_________________________________________________________________________________________
conv2d_10 (Conv2D)          (None, None, None, 9 82944                 activation_9[0][0]
_________________________________________________________________________________________
conv2d_11 (Conv2D)          (None, None, None, 3 6144            average_pooling2d[0][0]
_________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, None, None, 6 192                  conv2d_5[0][0]
_________________________________________________________________________________________
batch_normalization_7 (BatchNor (None, None, None, 6 192                  conv2d_7[0][0]
_________________________________________________________________________________________
activation_29 (Activation)  (None, None, None, 9 0          batch_normalization_29[0][0]
_________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, None, None, 2 0                    mixed2[0][0]
_________________________________________________________________________________________
mixed3 (Concatenate)        (None, None, None, 7 0                    activation_26[0][0]
                                      activation_29[0][0]
                                      max_pooling2d_2[0][0]
_________________________________________________________________________________________
batch_normalization_57 (BatchNo (None, None, None, 1 480                 conv2d_57[0][0]
_________________________________________________________________________________________
activation_52 (Activation)  (None, None, None, 1 0          batch_normalization_52[0][0]
_________________________________________________________________________________________
activation_57 (Activation)  (None, None, None, 1 0          batch_normalization_57[0][0]
_________________________________________________________________________________________
average_pooling2d_5 (AveragePoo (None, None, None, 7 0                    mixed5[0][0]
_________________________________________________________________________________________
conv2d_50 (Conv2D)          (None, None, None, 1 147456                   mixed5[0][0]
_________________________________________________________________________________________
conv2d_53 (Conv2D)          (None, None, None, 1 215040              activation_52[0][0]
_________________________________________________________________________________________
conv2d_58 (Conv2D)          (None, None, None, 1 215040              activation_57[0][0]
_________________________________________________________________________________________
conv2d_59 (Conv2D)          (None, None, None, 1 147456        average_pooling2d_5[0][0]
_________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, None, None, 1 576                 conv2d_50[0][0]
_________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, None, None, 1 576                 conv2d_53[0][0]
_________________________________________________________________________________________
batch_normalization_58 (BatchNo (None, None, None, 1 576                 conv2d_58[0][0]
_________________________________________________________________________________________
batch_normalization_59 (BatchNo (None, None, None, 1 576                 conv2d_59[0][0]
_________________________________________________________________________________________
activation_50 (Activation)  (None, None, None, 1 0          batch_normalization_50[0][0]
_________________________________________________________________________________________
activation_53 (Activation)  (None, None, None, 1 0          batch_normalization_53[0][0]
_________________________________________________________________________________________
activation_58 (Activation)  (None, None, None, 1 0          batch_normalization_58[0][0]
_________________________________________________________________________________________
activation_59 (Activation)  (None, None, None, 1 0          batch_normalization_59[0][0]
_________________________________________________________________________________________
mixed6 (Concatenate)        (None, None, None, 7 0                    activation_50[0][0]
                                                                      activation_53[0][0]
```



```
Model: "model"
_________________________________________________________________________________________
Layer (type)                Output Shape        Param #                      Connected to
=========================================================================================
============
 [0][0]
_________________________________________________________________________________________
conv2d_64 (Conv2D)          (None, None, None, 1 147456                   mixed6[0][0]
_________________________________________________________________________________________
batch_normalization_64 (BatchNo (None, None, None, 1 576                 conv2d_64[0][0]
_________________________________________________________________________________________
activation_64 (Activation)  (None, None, None, 1 0          batch_normalization_64[0][0]
_________________________________________________________________________________________
conv2d_65 (Conv2D)          (None, None, None, 1 258048              activation_64[0][0]
_________________________________________________________________________________________
batch_normalization_65 (BatchNo (None, None, None, 1 576                 conv2d_65[0][0]
_________________________________________________________________________________________
activation_65 (Activation)  (None, None, None, 1 0          batch_normalization_65[0][0]
_________________________________________________________________________________________
conv2d_61 (Conv2D)          (None, None, None, 1 147456                   mixed6[0][0]
_________________________________________________________________________________________
conv2d_66 (Conv2D)          (None, None, None, 1 258048              activation_65[0][0]
_________________________________________________________________________________________
batch_normalization_61 (BatchNo (None, None, None, 1 576                 conv2d_61[0][0]
_________________________________________________________________________________________
batch_normalization_66 (BatchNo (None, None, None, 1 576                 conv2d_66[0][0]
_________________________________________________________________________________________
activation_61 (Activation)  (None, None, None, 1 0          batch_normalization_61[0][0]
_________________________________________________________________________________________
activation_66 (Activation)  (None, None, None, 1 0          batch_normalization_66[0][0]
_________________________________________________________________________________________
conv2d_62 (Conv2D)          (None, None, None, 1 258048              activation_61[0][0]
_________________________________________________________________________________________
conv2d_67 (Conv2D)          (None, None, None, 1 258048              activation_66[0][0]
_________________________________________________________________________________________
batch_normalization_62 (BatchNo (None, None, None, 1 576                 conv2d_62[0][0]
_________________________________________________________________________________________
batch_normalization_67 (BatchNo (None, None, None, 1 576                 conv2d_67[0][0]
_________________________________________________________________________________________
activation_62 (Activation)  (None, None, None, 1 0          batch_normalization_62[0][0]
_________________________________________________________________________________________
activation_67 (Activation)  (None, None, None, 1 0          batch_normalization_67[0][0]
_________________________________________________________________________________________
average_pooling2d_6 (AveragePoo (None, None, None, 7 0                    mixed6[0][0]
_________________________________________________________________________________________
conv2d_60 (Conv2D)          (None, None, None, 1 147456                   mixed6[0][0]
_________________________________________________________________________________________
conv2d_63 (Conv2D)          (None, None, None, 1 258048              activation_62[0][0]
_________________________________________________________________________________________
conv2d_68 (Conv2D)          (None, None, None, 1 258048              activation_67[0][0]
_________________________________________________________________________________________
conv2d_69 (Conv2D)          (None, None, None, 1 147456        average_pooling2d_6[0][0]
_________________________________________________________________________________________
batch_normalization_60 (BatchNo (None, None, None, 1 576                 conv2d_60[0][0]
_________________________________________________________________________________________
batch_normalization_63 (BatchNo (None, None, None, 1 576                 conv2d_63[0][0]
_________________________________________________________________________________________
batch_normalization_68 (BatchNo (None, None, None, 1 576                 conv2d_68[0][0]
_________________________________________________________________________________________
batch_normalization_69 (BatchNo (None, None, None, 1 576                 conv2d_69[0][0]
_________________________________________________________________________________________
```



```
Model: "model"
Layer (type)                   Output Shape        Param #                                   Connected to
====================================================================================
============
 [0][0]
mixed7 (Concatenate)           (None, None, None, 7 0                              activation_60[0][0]
                                                                                   activation_63[0][0]
                                                                                   activation_68[0][0]
                                                                                   activation_69[0][0]
conv2d_72 (Conv2D)             (None, None, None, 1 147456                               mixed7[0][0]
conv2d_80 (Conv2D)             (None, None, None, 4 573440                               mixed8[0][0]
batch_normalization_80 (BatchNo (None, None, None, 4 1344                             conv2d_80[0][0]
activation_80 (Activation)     (None, None, None, 4 0                     batch_normalization_80[0][0]
conv2d_77 (Conv2D)             (None, None, None, 3 491520                               mixed8[0][0]
conv2d_81 (Conv2D)             (None, None, None, 3 1548288                       activation_80[0][0]
batch_normalization_77 (BatchNo (None, None, None, 3 1152                             conv2d_77[0][0]
batch_normalization_81 (BatchNo (None, None, None, 3 1152                             conv2d_81[0][0]
activation_77 (Activation)     (None, None, None, 3 0                     batch_normalization_77[0][0]
activation_81 (Activation)     (None, None, None, 3 0                     batch_normalization_81[0][0]
conv2d_78 (Conv2D)             (None, None, None, 3 442368                        activation_77[0][0]
conv2d_79 (Conv2D)             (None, None, None, 3 442368                        activation_77[0][0]
conv2d_82 (Conv2D)             (None, None, None, 3 442368                        activation_81[0][0]
conv2d_83 (Conv2D)             (None, None, None, 3 442368                        activation_81[0][0]
average_pooling2d_7 (AveragePoo (None, None, None, 1 0                                   mixed8[0][0]
conv2d_76 (Conv2D)             (None, None, None, 3 409600                               mixed8[0][0]
conv2d_89 (Conv2D)             (None, None, None, 4 917504                               mixed9[0][0]
batch_normalization_89 (BatchNo (None, None, None, 4 1344                             conv2d_89[0][0]
activation_89 (Activation)     (None, None, None, 4 0                     batch_normalization_89[0][0]
conv2d_86 (Conv2D)             (None, None, None, 3 786432                               mixed9[0][0]
conv2d_90 (Conv2D)             (None, None, None, 3 1548288                       activation_89[0][0]
batch_normalization_86 (BatchNo (None, None, None, 3 1152                             conv2d_86[0][0]
batch_normalization_90 (BatchNo (None, None, None, 3 1152                             conv2d_90[0][0]
activation_86 (Activation)     (None, None, None, 3 0                     batch_normalization_86[0][0]
activation_90 (Activation)     (None, None, None, 3 0                     batch_normalization_90[0][0]
```

```
Model: "model"
Layer (type)                   Output Shape        Param #                                   Connected to
====================================================================================
============
conv2d_87 (Conv2D)             (None, None, None, 3 442368                        activation_86[0][0]
conv2d_88 (Conv2D)             (None, None, None, 3 442368                        activation_86[0][0]
conv2d_91 (Conv2D)             (None, None, None, 3 442368                        activation_90[0][0]
conv2d_92 (Conv2D)             (None, None, None, 3 442368                        activation_90[0][0]
average_pooling2d_8 (AveragePoo (None, None, None, 2 0                                   mixed9[0][0]
conv2d_85 (Conv2D)             (None, None, None, 3 655360                               mixed9[0][0]
batch_normalization_87 (BatchNo (None, None, None, 3 1152                             conv2d_87[0][0]
batch_normalization_88 (BatchNo (None, None, None, 3 1152                             conv2d_88[0][0]
batch_normalization_91 (BatchNo (None, None, None, 3 1152                             conv2d_91[0][0]
batch_normalization_92 (BatchNo (None, None, None, 3 1152                             conv2d_92[0][0]
conv2d_93 (Conv2D)             (None, None, None, 1 393216                  average_pooling2d_8[0][0]
batch_normalization_85 (BatchNo (None, None, None, 3 960                              conv2d_85[0][0]
activation_87 (Activation)     (None, None, None, 3 0                     batch_normalization_87[0][0]
activation_88 (Activation)     (None, None, None, 3 0                     batch_normalization_88[0][0]
activation_91 (Activation)     (None, None, None, 3 0                     batch_normalization_91[0][0]
activation_92 (Activation)     (None, None, None, 3 0                     batch_normalization_92[0][0]
batch_normalization_93 (BatchNo (None, None, None, 1 576                              conv2d_93[0][0]
activation_85 (Activation)     (None, None, None, 3 0                     batch_normalization_85[0][0]
mixed9_1 (Concatenate)         (None, None, None, 7 0                             activation_87[0][0]
                                      activation_88[0][0]
concatenate_1 (Concatenate)    (None, None, None, 7 0                             activation_91[0][0]
                                      activation_92[0][0]
activation_93 (Activation)     (None, None, None, 1 0                     batch_normalization_93[0][0]
mixed10 (Concatenate)          (None, None, None, 2 0                             activation_85[0][0]
                                      mixed9_1[0][0]
                                      concatenate_1[0][0]
                                      activation_93[0][0]
====================================================================================
============
Total params: 21,802,784
Trainable params: 21,768,352
Non-trainable params: 34,432
```

Figure 5. Model summary.

G. Model training and optimization

Machine learning models are essentially centered on optimization, with algorithms trained to do jobs as quickly and effectively as possible. Machine learning models are used to foresee the outcomes of a function, whether categorizing an object or forecasting data patterns. Creating the most accurate model that can reliably transform inputs into predicted outputs is the aim. The objective of optimization is to improve the model's accuracy while decreasing the possibility of errors or losses brought on by these predictions. On offline or locally static datasets, machine learning models are routinely trained.

Optimization increases prediction and classification accuracy while lowering error. Hyper parameter optimization is necessary to build an accurate model. The best model configurations are selected with consideration for the model's accuracy and ability to perform specific tasks. Hyper parameter optimization might be difficult, though. Correctness is vital because models that are either over- or under-optimized face the risk of failure. The wrong hyper parameters might cause machine learning models to either fit the data incorrectly or too well. Machine learning models aim for some level of generalization to be useful in a dynamic environment with new datasets. Here are a few measures that were taken to optimize the model that was suggested for this project: -

- Set the optimizer & loss object
- Create the checkpoint path
- Create the training & testing step functions
- Create the loss function for the test dataset

H. Model evaluation

The steps used in evaluating the model implemented are mentioned below: -

- Set up evaluation function using Greedy Search
- Test on sample data images
- Evaluate based on BLEU Score

Greedy search -

One approach to problem-solving is the greedy strategy, which is like the divide-and-conquer approach. This solution addresses optimization-related issues. An optimization issue calls for maximum or minimal outcomes. The simplest and most direct strategy is gluttony. It is a method rather than an algorithm. Most decisions made using this technique must be supported by current information. Regardless of the current state of knowledge, decisions are taken without considering how they may impact the future. To construct the response in the best way feasible, this technique creates two sets, one having all the picked items and the other containing the rejected items. In anticipation of a successful or ideal conclusion, a greedy algorithm makes prudent local judgments. [39] [40]

Pseudo code of greedy search: -

```
Algorithm Greedy (a, n)
{
 Solution: = 0;
 for i = 0 to n do
  {
    x: = select(a);
   if feasible(solution, x)
   {
     Solution: = union(solution , x)
   }
    return solution;
   }  }
```

Beam search –

By expanding the most secure node in a small collection, a heuristic search technique known as beam search may traverse a network. The beam search technique, which requires less memory, optimizes the best-first search. All partial answers are arranged using a heuristic in the best-first search method of graph search. Beam search, however, only keeps a select few of the best partial answers as candidates. As a result, the algorithm is greedy. Beam search constructs its search tree using breadth-first search. All the states at each level of the tree are constructed and are put in increasing order of heuristic cost. The optimal states at each level, or beam width, are only stored in a finite number of them. With increasing beam width, fewer states are trimmed. No states are pruned in the case of infinite beam width, and beam search is equivalent to breadth-first search. It is common to provide the initial beam search solution. The method will assess the solutions discovered during a search at various depths and return the best one with the highest probability after it reaches the chosen maximum search depth (i.e., translation length).

Algorithm of beam search: -

```
beamSearch(problemSet, ruleSet, memorySize)
     openMemory = new memory of size memorySize
     nodeList = problemSet.listOfNodes
     node = root or initial search node
     Add node to openMemory;
     while
 Delete node from openMemory;
Expand the node to get its progeny, then assess them;
Delete a child node that has been pruned by a rule in the ruleSet;
Add the last group of unpruned kids to openMemory;
Remove the worst node found by ruleSet in openMemory if memory is full and cannot accommodate additional nodes;
node = the openMemory node with the lowest cost;
```

IV. RESULTS

The final step is to evaluate the model proposed in the project. This section describes the outputs of the model using various metrics and compares the accuracy of the test images. For training the model, masking is applied since Padding has the danger of penalizing the model further. Once the padding is done, masking is applied. Without masking, the model will consider the padded input at that timestamp, which will contribute to an increased loss. Through masking the model is informed to ignore whenever a padded input is passed at a timestamp**,** hinting that this part of the input is padded. To correct the situation, masking is used, which will reduce any further penalties to zero**.** While creating the training function of the model, teacher forcing is applied because there are multiple issues using recurrent neural networks for training when the input is the output from earlier time steps. Some of them are as follows: -

- Slow convergence
- Model instability
- Poor skill

The procedure will continue if the preceding output is inaccurate (by chance), which will cause inaccurate input for the following time stamp and lead to an unexpected output. As a result, the model will stray from its intended path and experience negative effects for each successive word it generates. Learning is slower as a result, and the model is unstable. To remedy this, teacher forcing is employed. The target/real word (i.e., ground truth) is provided as the next input to the decoder instead of the previous prediction or output in a method called teacher forcing, which is quick and efficient for training recurrent neural networks. Training from this method happens more quickly when forced. The model's predictions are incredibly poor during the initial phases of training. Without utilizing instructor forcing, the model's hidden states will be updated by a succession of inaccurate predictions, errors will arise, and the model will struggle to learn from those mistakes. Figure 6 shows the training vs. test loss. The final test loss achieved after 15 epochs is 0.482. Table 3 shows the test and training loss for first 7 epochs.

Table 3. Test and training loss for 7 epochs.

Epoch	*Training loss*	*Testing loss*
Epoch 1	1.437	1.203
Epoch 2	1.111	1.059
Epoch 3	1.007	0.975
Epoch 4	0.935	0.909
Epoch 5	0.875	0.853
Epoch 6	0.823	0.802
Epoch 7	0.774	0.758

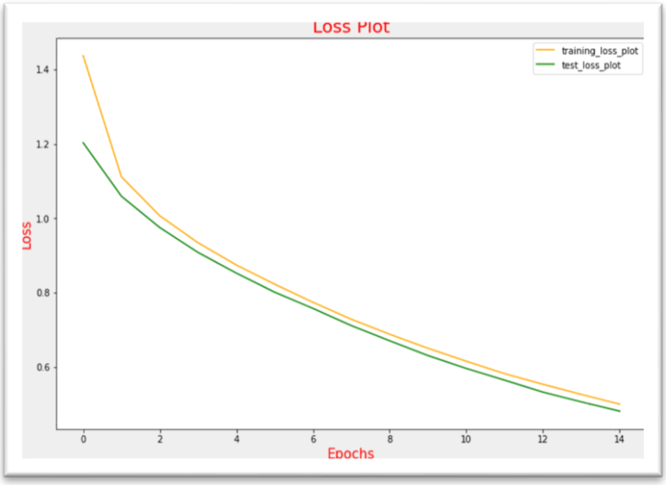
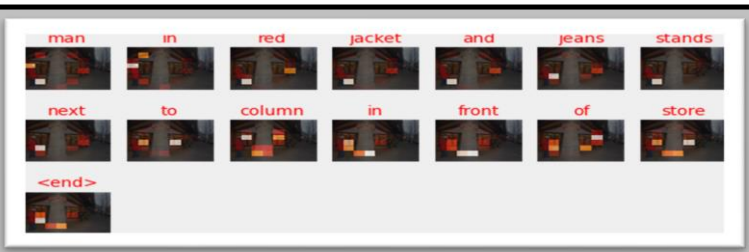
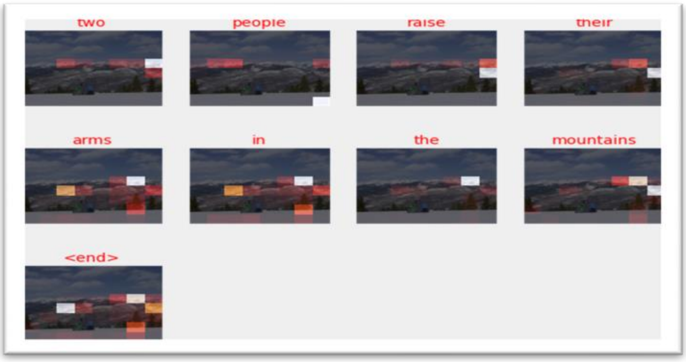
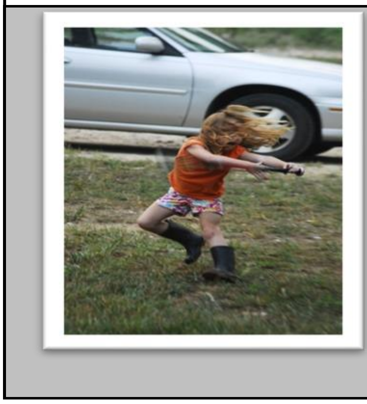
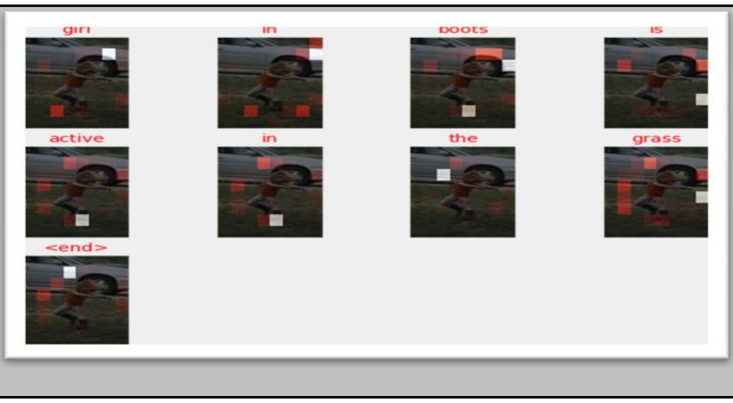


Figure 6. Loss plot

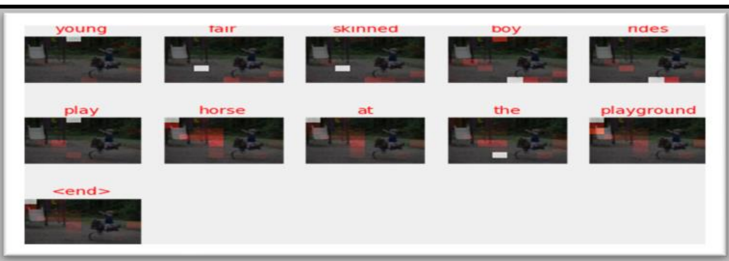
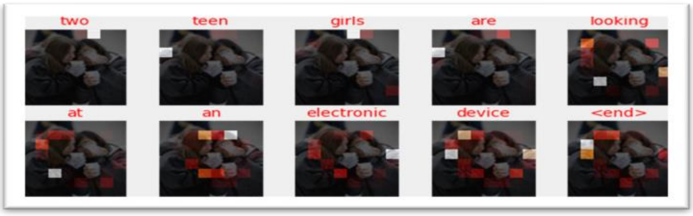
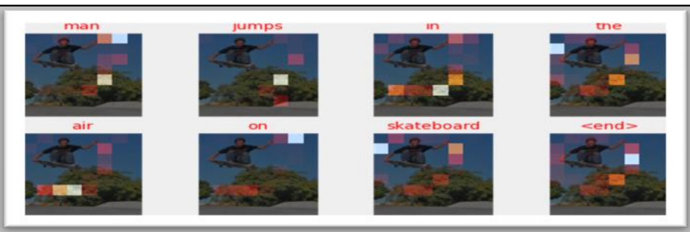
BLEU measure is used to evaluate the result of the test set generated captions. It is a widely used statistic to assess how closely two reference sentences and one hypothesis sentence resemble each other. It returns a number between 0 and 1 in response to a single hypothesis statement and several reference phrases. If the two are highly similar, the score will be close to 1. The test set generated captions is evaluated using the BLEU metric. The BLEU merely uses the portion of the anticipated sentence's n-grams that are present in the ground truth.

Table 4 shows the test images and their corresponding BLEU scores.

Table 4. BLEU scores.

Test image	***Predicted caption***	***BLEU score***
	 man in red jacket and jeans stands next to column in front of store <end>	**BELU score:** 100.0 **Real Caption:** man in red jacket and jeans stands next to column in front of store **Predicted Caption:** man in red jacket and jeans stands next to column in front of store
	 two people raise their arms in the mountains <end>	**BELU score:** 66.1306622519314 **Real Caption:** two people raise their arms on snowy hill in the mountains **Predicted Caption:** two people raise their arms in the mountains
	 girl in boots is active in the grass <end>	**BELU score:** 100.0 **Real Caption:** girl in boots is active in the grass **Predicted Caption:** girl in boots is active in the grass

Article Received: *26 August 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *02 November 2023*

Test image	***Predicted caption***	***BLEU score***
	 young fair skinned boy rides play horse at the playground <end>	**BELU score:** 90.51045767389155 **Real Caption:** young fair skinned boy rides play horse at playground **Predicted Caption:** young fair skinned boy rides play horse at the playground
	 two teen girls are looking at an electronic device <end>	**BELU score:** 52.35215949109693 **Real Caption:** two teen girls are looking at small electronic device while wearing winter coats **Predicted Caption:** two teen girls are looking at an electronic device
	 man jumps in the air on skateboard <end>	**BELU score:** 79.13476338264609 **Real Caption:** man jumps in the air on his skateboard **Predicted Caption:** man jumps in the air on skateboard

V. CONCLUSION

The challenging problem of computer vision and natural language processing is image captioning. The main difficulty is to use machines to extract semantic information from photos and translate it into human language. The difficulty of an image captioning is further increased by the interaction between computer vision and natural language processing. The issue is intriguing not only because it has significant practical implications, such as improving the vision of the blind, but also because image interpretation, a key issue in computer vision, is seen as a big task. Image captioning must go beyond object identification and image categorization in order to produce a meaningful natural language description of an image. The connection between Computer Vision and Natural Language Processing, two of the main subfields of Artificial Intelligence, makes the challenge much more intriguing. Notably, studies have been done on how to successfully use deep learning approaches to close the semantic gap in image captioning. Deep learning methods are adept at handling the difficulties of picture captioning. A thorough investigation is conducted to determine the numerous cutting-edge methods for image captioning. Existing techniques to picture captioning fall into one of two categories: top-down or bottom-up. The bottom-up paradigm first creates words describing various characteristics of an image and then combines them, in contrast to the top-down paradigm which starts with the "gist" of an image and translates it into words. Both paradigms use language models to create intelligible statements. The top-down paradigm, which uses recurrent neural networks to formulate an end-to-end process from an image to a phrase, is at the cutting edge. All the parameters of the recurrent network may be learnt from training data.

In this project, a CNN-RNN attention-based model is proposed to address the problem faced by visually impaired in visualizing the images. The top-down and bottom-up techniq

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 26 August 2023 **Revised:** 20 October 2023 **Accepted:** 02 November 2023*

ues to image captioning are combined in this project using a semantic attention model. The capacity to offer a thorough, coherent description of semantically significant items that are required precisely when they are required is the definition of semantic attention in picture captioning. The semantic attention model proposed in this project includes the following characteristics:

- Being able to focus on a semantically significant idea or region of interest in an image
- Being able to weigh the relative strength of attention given to other concepts
- Being able to dynamically transition between concepts depending on the job at hand

A bottom-up strategy is used to choose semantic ideas or traits as candidates for attention, and a top-down visual feature is used to designate where and when attention should be triggered. Our model, which is based on a recurrent neural network (RNN), initially collects global data via a top-down feature. Through an attention mechanism imposed on both network state and output nodes, the bottom-up qualities provide feedback and interaction to the RNN state as it transitions. This input enables the computer to more effectively anticipate new words while also enabling more reliable inference of the semantic gap between prior predictions and the content of the image. The approach used in this project consequently makes extensive use of fine-grain visual semantic elements in addition to an overall knowledge of the input image. Our model's actual strength resides in its capacity to pay attention to these factors and smoothly combine global and local data for improved caption.

1) We have experimented with multiple other weights (12-15 different combinations) but found the best results mainly for the below:

- (0.5, 0.5, 0, 0)
- (0.5, 0.25, 0, 0)
- (0.25, 0.25 , 0, 0)
- (0.25, 0.35 , 0, 0)

The probability of receiving a lower Bleu score (less than 50%) was lowest for the weights. Using the weights, we exceeded 70% accuracy for most of the test photos. Additionally, using the weights (0.5, 0.5, 0, 0), (0.5, 0.25, 0, 0), and (0.25, 0.5, 0, 0), we obtained a multiple accuracy of 100%.

CNN uses the Inception V3 Model with pre-trained weights (to extract feature vectors) 1. On the ImageNet dataset, the popular Inception v3 image recognition model showed greater than 78.1% accuracy. The model is the result of several concepts on which numerous scholars have been working for years. The model's symmetric and asymmetric building elements include convolutions, average pooling, max pooling, concatenations, dropouts, and entirely linked layers. The activation inputs of the model undergo extensive batch normalization. To calculate loss, Softmax is used. In this project, the decoder (RNN) has been employed with GRU. The attention Model is also utilized by the decoder to ensure that pertinent and areas of the image are focused on more intensively than the full image at a given timestamp. This ensured improved accuracy, reduced noise, and quicker computation. Losses were computed using the predictions, and the output of the decoder, which comprises the predicted caption and hidden state, is returned to the model. To determine the loss, we utilized cross entropy - SparseCategoricalCrossentropy. To choose the next input for the decoder, we also used teacher forcing. It ensured a faster convergence and a model that remained stable over time. We used 15 epochs to train the model, and the overall loss was decreased to 0.482. To forecast captions, it is computed how likely a word is to exist in the corpus. Based on the terms' frequency in the supplied vocabulary list, one may estimate the likelihood of each phrase, Greedy Search is used. The term with the highest probability is output. To assess the efficacy and performance of our model, we used BLEU Score. The model performs better when the BLEU score is higher.

VI. LIMITATIONS AND FUTURE WORK

The main constraint was that there was just one publicly available decent Dataset that contained photographs relevant to the motivation. It might be possible to evaluate different preprocessing and deep learning methods, but this would require further research and testing. Therefore, we may take this into account in our future work. The results of this study can be used by researchers by incorporating them into their ongoing projects. A larger dataset may be created and made available for use in this field's linguistic resources study. The customization of the use of extra deep learning methods is made possible by having a big dataset, and with better word representation and other characteristics, photo captioning performance may be greatly enhanced.

REFERENCES

[1] SS, Roshan Adhithya, M. Priyadharshini, and Lekshmi Kalinathan. "Image Caption Generation For Blind Users Of Social Media Websites." (2023).

[2] Alzubaidi, Laith, et al. "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions." Journal of big Data 8 (2021): 1-74.

[3] Verma, Akash, et al. "Intelligence Embedded Image Caption Generator using LSTM based RNN Model." 2021 6th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2021.

[4] Al Faraby, Hasan, et al. "Image to bengali caption generation using deep cnn and bidirectional gated recurrent unit." 2020 23rd international conference on computer and information technology (ICCIT). IEEE, 2020.

[5] Mishra, Sanjukta, and Minakshi Banerjee. "Automatic caption generation of retinal diseases with self-trained rnn merge model." Advanced Computing and Systems for Security: Volume Twelve (2020): 1-10.

[6] Soh, Moses. "Learning CNN-LSTM architectures for image caption generation." Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep 1 (2016).

[7] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.

[8] Hijazi, Samer, Rishi Kumar, and Chris Rowen. "Using convolutional neural networks for image recognition." Cadence Design Systems Inc.: San Jose, CA, USA 9 (2015): 1.

[9] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML deep learning workshop. Vol. 2. No. 1. 2015.

[10] Zheng, Heliang, et al. "Learning multi-attention convolutional neural network for fine-grained image recognition." Proceedings of the IEEE international conference on computer vision. 2017.

[11] Traore, Boukaye Boubacar, Bernard Kamsu-Foguem, and Fana Tangara. "Deep convolution neural network for image recognition." Ecological informatics 48 (2018): 257-268.

[12] Wang, Haoran, Yue Zhang, and Xiaosheng Yu. "An overview of image caption generation methods." Computational intelligence and neuroscience 2020 (2020).

[13] Chen, Xinlei, and C. Lawrence Zitnick. "Mind's eye: A recurrent visual representation for image caption generation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[14] Sadiku, Matthew, et al. "Data visualization." International Journal of Engineering Research And Advanced Technology (IJERAT) 2.12 (2016): 11-16.

[15] Ajibade, Samuel Soma, and Anthonia Adediran. "An overview of big data visualization techniques in data mining." International Journal of Computer Science and Information Technology Research 4.3 (2016): 105-113.

[16] Wu, Aoyu, et al. "Ai4vis: Survey on artificial intelligence approaches for data visualization." IEEE Transactions on Visualization and Computer Graphics (2021).

[17] Wang, Qianwen, et al. "Applying machine learning advances to data visualization: A survey on ml4vis." arXiv preprint arXiv:2012.00467 (2020).

[18] Raschka, Sebastian, Joshua Patterson, and Corey Nolet. "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence." Information 11.4 (2020): 193.

[19] Lee, Ga Young, et al. "A survey on data cleaning methods for improved machine learning model performance." arXiv preprint arXiv:2109.07127 (2021).

[20] Subasi, Abdulhamit. Practical machine learning for data analysis using python. Academic Press, 2020.

[21] Du, Peijun, et al. "Advances of four machine learning methods for spatial data handling: A review." Journal of Geovisualization and Spatial Analysis 4 (2020): 1-25.

[22] Yang, Jian-Bo, et al. "Likelihood Analysis of Imperfect Data." IEEE Transactions on Systems, Man, and Cybernetics: Systems (2023).

[23] Ahuja, Avani, Lidia Al-Zogbi, and Axel Krieger. "Application of noise-reduction techniques to machine learning algorithms for breast cancer tumor identification." Computers in Biology and Medicine 135 (2021): 104576.

[24] Reddy, G. Thippa, et al. "Analysis of dimensionality reduction techniques on big data." Ieee Access 8 (2020): 54776-54788.

[25] Sun, Lin, et al. "Multilabel feature selection using ML-ReliefF and neighborhood mutual information for multilabel neighborhood decision systems." Information Sciences 537 (2020): 401-424.

[26] Aganja, Aakriti, et al. IMAGE CAPTIONING USING CNN AND DEEP STACKED LSTM. 2021.

[27] Tanti, Marc, Albert Gatt, and Kenneth P. Camilleri. "Where to put the image in an image caption generator." Natural Language Engineering 24.3 (2018): 467-489.

[28] Pandey, Ashutosh, and DeLiang Wang. "A new framework for CNN-based speech enhancement in the time domain." IEEE/ACM Transactions on Audio, Speech, and Language Processing 27.7 (2019): 1179-1188.

[29] Xiao, Qingcheng, and Yun Liang. "Fune: An FPGA tuning framework for CNN acceleration." IEEE Design & Test 37.1 (2019): 46-55.

[30] Laptev, Aleksandr, et al. "Powerful and Extensible WFST Framework for Rnn-Transducer Losses." ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023.

[31] Wang, Chenglong, Feijun Jiang, and Hongxia Yang. "A hybrid framework for text modeling with convolutional RNN." Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017.

[32] Hoxha, Genc, Farid Melgani, and Jacopo Slaghenauffi. "A new CNN-RNN framework for remote sensing image captioning." 2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS). IEEE, 2020.

[33] Liu, Shuang, et al. "Image captioning based on deep neural networks." MATEC web of conferences. Vol. 232. EDP Sciences, 2018.

[34] Jiang, Wenhao, et al. "Learning to guide decoding for image captioning." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

[35] Dalianis, Hercules, and Hercules Dalianis. "Evaluation metrics and evaluation." Clinical text mining: secondary use of electronic patient records (2018): 45-53.

[36] Aafaq, Nayyer, et al. "Video description: A survey of methods, datasets, and evaluation metrics." ACM Computing Surveys (CSUR) 52.6 (2019): 1-37.

[37] Gaur, Eshan, Vikas Saxena, and Sandeep K. Singh. "Video annotation tools: A Review." 2018 International Conference

on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018.

[38] Zhou, Zelin, et al. "Can audio captions be evaluated with image caption metrics?." ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

[39] Amritkar, Chetan, and Vaishali Jabade. "Image caption generation using deep learning technique." 2018 fourth international conference on computing communication control and automation (ICCUBEA). IEEE, 2018.

[40] Fernandez-Viagas, Victor, Jorge MS Valente, and Jose M. Framinan. "Iterated-greedy-based algorithms with beam search initialization for the permutation flowshop to minimise total tardiness." Expert Systems with Applications 94 (2018): 58-69.