

VaaniChitran (A Visual Assistance)

A major project report submitted in partial fulfillment of the requirement
for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

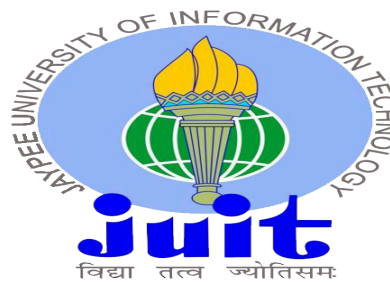
Submitted by

Chirag Jain(201267)

Amartya Vibhu(201413)

Under the guidance & supervision of

Dr Maneet Singh



**Department of Computer Science & Engineering and
Information Technology**

Jaypee University of Information Technology, Wagnaghat,

Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled “**VaaniChitran: A Visual Assistance** ” in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering and submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by “Chirag Jain (201267), Amartya Vibhu (201413).” under the supervision of Dr Maneet Singh, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Chirag Jain - (201267)

Amartya Vibhu- (201413)

The above statement made is correct to the best of my knowledge.

Dr. Maneet singh

Designation - Assistant Professor (Senior Grade)

Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat

Candidate's Declaration

We hereby declare that the work presented in this report entitled '**VaaniChitran**' in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr Maneet Singh** (Assistant Professor, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Student Name: Chirag Jain

Roll No.: 201267

(Student Signature)

Student Name: Amartya Vibhu

Roll No.: 201413

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr Maneet Singh

Designation: Assistant Professor (Senior Grade)

Department: CSE

Date:

ACKNOWLEDGEMENT

Firstly, We express our heartiest thanks and gratefulness to almighty God for His divine blessing making it possible for us to complete the project work successfully.

We are grateful and wish my profound indebtedness to Supervisor **Dr. Maneet Singh**, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of “**Research Area**” to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express my heartiest gratitude to **Dr. Maneet Singh**, Department of CSE, for his kind help to finish my project.

We would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, we would like to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, We must acknowledge with due respect the constant support and patience of our parents.

Chirag Jain - (201267)

Amartya Vibhu-(201413)

TABLE OF CONTENT

TITLE	PAGE NO.
List of Figures	v
Abstract	vi-vii
Chapter-1 Introduction	1-9
Chapter-2 Literature survey	10-12
Chapter-3 System Development	13-44
Chapter-4 Testing	45-46
Chapter-5 Results	47-51
Chapter-6 Conclusion and Future scope	52-59

LIST OF FIGURES

Figure No.	Page No.
Figure 1 Level-1 DFD	16
Figure 2 Level-2 DFD	17
Figure 3 Encoder (CNN) ResNet-50	20
Figure 4 Encoder (CNN)+Decoder (RNN)	23
Figure 5 Home page of website	48
Figure 6 Quantitative results	49
Figure 7 Quantitative results	50
Figure 8 Quantitative results	51

ABSTRACT

This is a new computer vision and natural language powered project on captions and visual assistive technologies integrating for the development of an inclusive human-computer interaction environment. It is a holistic solution that not only identifies and tags visual elements of shots but also smart assistants who will understand user commands and provide contextual information.

Extract with modern computer vision with precise rich visual features from images of the system. It uses natural language processing to interpret user's queries and commands and the proposed caption model uses deep learning based on convolutional neural network and recurrent neural network providing contextual and explanatory captions to the input images. It makes it accessible to the users by providing the system to present content through natural language.

The system will work as visual assistance on user questions about identified items or scenarios or any other visual context. It is through spoken and typed commands that users communicate with the system in order to achieve the user's interface.

The system develops context-aware responses by mixing the captions so user queries and user details. This makes the assistant's replies look realistic and relevant to the user's queries to enhance more intelligent and customized interaction.

As the users interact with the system it understands natural language and recognizes images over time. The third aspect involves adaptive learning, which refers to how the system learns to keep up with the changing user needs and preferences.

The incorporation of integrated subtitles and visual assistants within this project's framework would prove to be a major milestone in the completion of the human-computer interaction paradigm.

Subtitles enhance visual assistive features and give more semantic understanding of visual content. An intelligent system that is just not a pure recognition but offers descriptions and background information for user requests. In other words the semantic density that facilitates more meaningful and human communication.

It is through this that a more extensive range of people can view with combined visual

perception and natural language comprehension opening up AR applications content search plus interactive virtual environments. It depicts the potential of applying multimodal technologies to develop interfaces that seamlessly integrate the visual and the linguistic planes.

This integrated system was built on a modular approach that affords it the possibility of being scaled and extended. Adding new functions such as more languages and advanced detection systems as well as the integration to other technologies will make the system adaptable to the requirements of evolving users and new technologies.

These are tasks that use a specially created dataset containing images and informative captions. Also we notice two different tasks of description and search and come up with simple-minded baseline systems. In this study we prove that the proposed automatic evaluation metrics for our ranking-based task do not provide accurate results.

The problem comes in describing the caption of the image automatically and it has an issue with deep learning which connects computer vision and natural language processing.

So our project model generates the sentence and describes the image with deep recurrent architecture and achievements of computer vision and text to speech.

Based on the training images it provides the description of the training images with high probability .

To demonstrate the accuracy of the model and tests are executed on several datasets. Secondly the language model is evidently picked from picture descriptions only. Our model is very predictive most of the time but it is not always.

CHAPTER-1 INTRODUCTION

1.1 INTRODUCTION

By combining computer vision and natural language processing, it increases human-computer interaction and accessibility. Its aim is to create a descriptive caption auto generating system with spoken and written output.

This means that users and people with visual impairments are able to understand the content of the images by turning it into speech.

There have been converging computer vision and natural language processing in today's artificial intelligence world paving the way to transformative technologies that connect visual to the linguistically understood perception. The journey begins with introducing a user interface that combines captions and visual aids. The aim of fusing these two systems together is to develop a mighty multi-functional device which does not only indicate and describe graphical content, but also serves as an assistant that understands people's questions and delivers related material.

With the passage of time and growth of a more visual digital world, smart systems that are able to communicate with and comprehend visual data have become necessary. The approaches involved in human-computer interaction are represented by captions and text to speech.

Describing an image's content using well formed English words is an extremely hard job. However, it can have a big impact, for instance, in making understanding of visual images easier for blind people. For instance, this is far much more difficult compared to well studied tasks such as object detection and image classification that are common in the computer vision community. A description should not only be about what is seen in an image but it should also contain information about what objects are made of and what activities they participate in as well as how these objects relate to one another. In addition the semantic information should be coded into an English-like natural language and thus a language model is required as well.

This paper focuses mainly on developments made in the field of machine translation, whose intention is to make $p(T|S)$ as high as possible to translate a sentence S written in a source language into its translation T in the target language. Similarly as for a long period there were several discrete tasks which involved in the machine translation including translating word by word, word alignment and word reordering. Despite this the recent studies have shown that translation can be done in a more straightforward way with the latest performance. In

particular, an “encoder” RNN reads and encodes the source sentence to become a rich fixed-length vector representation, which subsequently acts as the initial hidden state of a “decoder” RNN that produces the target sentence.

To turn text into artificial human speech, a TTS synthesis is an essential part that is affected by the voice-enabled devices, navigation systems and visually impaired people have access to it. In this way, it creates a communication mode without optical devices. The most recent TTS systems are based on intricate, multi-stage, processing pipelines, which nearly all of them each may depend on hardwired features and quirks. It is so cumbersome for someone who is among the developers to build the new TTS systems. Deep Voice is inspired by traditional text-to-speech pipelines and adopts the same structure, while replacing all components with neural networks and using simpler features: first step is to transfer the text to phoneme and then we apply audio synthesis model that helps convert linguistic features into speech. Contrary to the previous works (that uses hand-crafted features like spectral envelope, spectral parameters, aperiodic orientation, etc.), our input only consists of phonemes with stress information, the timespan of phonemes (duration) and fundamental frequency (F0). The adoption of this set of features implies that our system would be less dependent on any particular dataset and would easily adapt to different voices and domains without the need for data annotation or additional feature engineering. Our claim is demonstrated through this process by training the entire pipeline again, all the while keeping the hyper parameter changes zero, using an all audio dataset that is mixed solely with unaligned textual transcriptions and then generating good quality speech after the training. In a conventional TTS system this adjustment procedure takes a day up to a week, while in Deep Voice platform this procedure is limited to few hours of manual effort and the time it takes models to train.

In real-time inference is essential for any fine-quality TTS system; otherwise, it is not practical for a vast majority of applications of TTS systems. Previous work has demonstrated that a text-to-speech system with the same WaveNet processors can generate the audio with very close to human-like quality. Yet, the inference of WaveNet is fraught with a major computational difficulty, which stems from the high-frequency autoregressive nature of the model and up till now, it is unknown whether these models can be trained for a production model. We answer this question affirmatively and present two WaveNet kernels that can run in the faster-than-real-time manner which create a high-quality audio and achieve a speedup over the previous WaveNet inference implementations.

1.2 PROBLEM STATEMENT

The proposed project aims to address this challenge by developing a good feature-rich web application that combines the advanced image captioning with seamless text-to-speech integration. In this user will get the option for converting their images or set of images into speech or caption. The web application will further incorporate text-to-speech conversion capabilities.

We make available a dataset of pictures that show a variety of commonplace activities and occurrences which are accompanied by five phrases that explain what the pictures depict. In order to assess systems on a sentence-based picture description and a sentence-based image retrieval task we present a ranking-based framework.

We create several robust baseline systems. Unlike previous research we demonstrate that high performance on this task might not necessitate explicit object and scene detectors. Our emphasis is on linguistic characteristics and we also demonstrate that lexical similarity and word order-capturing models perform better than basic bag of words methods. We compare multiple automatic evaluation metrics which on demonstration and compared to other metrics, ranking-based metrics correlate better with human judgements.

To generate the natural sounding speech, TTS has to infall a number of performances which is either imputed explicitly or implicitly which is not in very simple text. Prosody which is defined by the use of intonation, stress, rhythm and style are the four different types of parameters that are being discussed. While the mapping of texts to speech via text-to-speech is a problem of underdetermination, and since the message is not fully specified by the absent text, it must be added with appropriate intonation meanings. To amplify the meaning that they are not sure about their knowledge, the speaker may decide to draw sounds that are going up through the use of rising pitch. One vulnerable issue which hinders us from labeling the prosody issue is automatically schematizing it and lack of explicit annotations, we, therefore, attempt to discover methods of modeling prosody that don't need any annotations and then present a technique that extracts the latent prosody representation directly from the ground truth speech audio.

Therefore, the other problem of which has been created and is important to solve by a generative model is the sampling. In other words, it is challenging for applications to generate

a nice and fresh prosody or output speech even when simple phonetics, speaker identities, and channel effects are used. However, we can regard the more primitive problem of designing a space that can display a gap. We suggest the construction of an internal space in which prosodic aspects are taken into account and prove that it leads to naturalness when speech is transferred to an artificial voice using a latent representation to make one utterance sound like another. This is something like practicing the explanation by the idea of the usage sentence. The key part of our idea is a special encoder that creates a fixed vector characterizing prosody based on the acoustic input; we prove that we are able to transfer between utterances with the same text and different speakers almost independently using the encoder. To evaluate the performance in this manner of transference exercise we suggest a number of qualitative and quantitative metrics. Beside the aforementioned fact, reading the lyrics would be more complete with the audio samples.

Moreover, TTS system development encompasses not only the field of theory but also the area of practical implementation, where a TTS system with a text-to-speech system capable of turning written text into high-quality speech output is going to be developed. This includes the software design and setup, which involves text processing algorithms and synthesis, and a variety of audio processing methods intended for speech naturalization. With the construction of the TTS system, the embodiment of the theoretical concepts will become manifest, creating an environment where the system can be investigated, experimented, and improved. Furthermore, this stage aims to pen the development by examining the system performance with a series of the evaluations, which entails subjective as well as objective assessments.

1.2.1 Image Captioning

1. Accuracy: Creating precise and normal language descriptions about images is a hard thing owing to the complicated nature of visual data and natural speech. Most image captioning systems fail to correctly pick out objects which attribute them and establish a connection between them resulting in incorrect or missing captions in some cases
2. Context and Semantics: Most times it will be difficult for an image captioning system to get the semantics and context behind. Captions which generates from these may be accurate but are sometimes deprived from the deeper message embedded in the image. In fact image captioning systems typically capitalized on object recognition and language model techniques which are not fit for sensing subtleties of a person's language and visual speech.

3. Creativity and Style: Mostly we see that image captioning applications come up with boring captions. This may make the captions appear uninspiring or stylistic with little creativity to offer informationally. Image captioning systems receive their training on large volumes of text and image data and that might fail to cover the richness and innovation of human language.
4. Multilingual Support: Most image captioning systems fail at producing captions in more than one language. They are also trained using a monolingual dataset thus lacking the capacity to translate in other languages.

1.2.2 Visual assistant

1. Privacy and Security: Due to the high volume of information gathered by visual helpers, issues related to privacy and security have been raised. Care has to be taken when ensuring that this information is treated well and properly.
2. Accessibility: People with disabilities such as visual impairments, hearing deficiency, and cognitive problems should have accessibility of visual assistants.
3. Ethical Consideration: Some of these ethical aspects in regards to vision assistants development and usage are bias, discriminations, and replacement of workers in an organization. However, due consideration should be given to issues that will help establish responsible guidelines for visual assistant development and usage.

Therefore these are some of the several issues that scholars dealing with image captioning and visual assistants are grappling with. If properly addressed it will lead to improved, natural and useful systems that are beneficial not only to individuals but also many other aspects.

1.3 OBJECTIVES

This main aim is to produce Automatic Caption and enhanced Human-Computer Interaction especially for visually impaired. It has Versatile Application Potential. It blends with the captions and other visual assisted functions effortlessly. This system is meant to give users an all-inclusive solution to accessing accurate and appropriate descriptions of visual content, in addition to allowing interaction with the system through natural language input resulting in a much more realistic environment for users.

Build a framework that will integrate the caption system with other visual assistive tools smoothly. It needs a multi-modal model, which is able to recognize, describe images and

videos as well as get and give essential data based on an order set by a user. Provides an intuitive user interface that allows natural communication with the combined systems. It involves designing an interactive interface for users to issue voice or textual commands and provide them with visual descriptions along with contextual explanations.

To achieve real time images recognition capability, the system must be able to fast process visually inputted information and produce instant results with ease. Such an objective is important, especially in the case of leading the blind and quick information service provision. Combine user queries and image captioning results to develop a system designed for context-aware responses. These should go beyond just being graphically appropriate but should be targeted specifically at the users' requests showing great understanding of both the visual and linguistic elements in the context.

Integrate an adaptive learning process involving users over a period of time. The response will be customized depending on user settings towards improved image perception. There will also be continuous learning towards better user experience. The crux of our approach is a special encoder that designed independent of the acoustic input, and is capable to transfer between utterances that have the same text but different speakers with virtually no error rate. For analyzing the efficiency we suggest using certain measurement tools herewith presentation types, such as quantitative and qualitative ones, for making conclusions. Besides this fact already mentioned, the lyrics hearing will have a better sound with the audio recording samples.

The subjective assessments will comprise of human listeners being the ones to judge the voices' quality and naturalness, while the objective assessments will parades metrics such as word error rate, prosody alignment, and spectral distortions. Analyzing the difficulties, the current inefficiency, and less than adequate dynamic synthesis will be essential parts of the feedback to be incorporated for the next version of our TTS system. This holistic mode of TTS exploration covers the main fundamentals of the TTS algorithms, implementation, evaluation, and finally the mastery of the techniques used in the text-to-speech system. At the end, the method shall contribute to the language development and practical TTS applications. For the last part, the project emphasizes libraries and information system which assume learning, collaboration, and contribution to the whole TTS community. The methodology includes such supporting activities as writing academic publications and documents, sharing insights and using best practices, and engaging with other researchers in this field with the

aim to deliver the best text to speech technology on the modern market.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

In Deep learning image captioning is primarily presented as a sequence-to-sequence problem. The objective of sequence-to-sequence problems is to translate a given sequence into its proper corresponding sequence. Machine translation is a fundamental problem in sequence-to-sequence translation. The process of creating a textual description of an image is called image captioning. To create the captions it combines computer vision and natural language processing.

We use Convolutional Neural Network (CNN) as an encoder. CNN is tasked with extracting the features from the input image. The CNN's final hidden state is linked to the decoder. The start vector and the encoded output from the encoder are received in the first time step of the Recurrent Neural Network (RNN) Decoder which also performs language modeling up to the word level.

The two technologies namely image captioning and visual assistants may prove critical in transforming how humans operate in their environments. There exists an automatic generation of a natural language description of an image while visual assistants are software programs based on computer vision and ability to read and reply to visual inputs.

Image captioning is a difficult but very rewarding field of research that seeks to connect a sight and text. The process entails creating algorithms that will adequately read an image's meaning and produce a similar English description. While this seems simple on the face of it and it is in fact far more complicated because the system must not only see objects and their attributes but also recognize the relationship of objects, action they take and general context.

So there are quite some strong and convincing reasons for researching image captioning. These include:

1. **Accessibility:** Captioning of images is instrumental in helping people with visual impairments. It allows them to see, feel and comprehend the visual world because it transfers images into textual descriptions.

2. Search and Retrieval: Captioning of images will enhanced the efficiency of image search as well as retrieval systems. It provides a textual representation of images by allowing people to perform searches through describing what is in the picture instead of searching through various visual keywords.
3. Content Understanding and Summarization: Image captioning is also a useful tool to summarize the contents of an image and video by providing a brief synopsis for users. In such cases it can be vital in social media, aggregating news or educational content.
4. Machine Learning and AI Fundamentals: Image captioning provide an important tool for developing and testing high-end machine and deep learning algorithms and AI methods. Addressing the difficulties associated with interpreting and producing human language based on picture data will allow scientists to improve their algorithms and obtain information about human vision and speech.
5. Content Creation and Storytelling: Additionally an image captioning can be used to generate multimedia materials like video with an explanatory voiceover or enhanced reality experience with touchable labels. It makes a way for new stories as well as interactive experiences.
6. Robotics and Automation: Image captioning has some relevance for the development of robotics and computer automation that may help the computers to understand their surroundings and communicate with objects and environments more accurately.
7. Medical Image Analysis: In the area of medicine image captioning may be used in interpreting the images such as X-rays, CT scans and MRIs. Through brief but meaningful descriptions it will help in clinical decision-making.
8. Cultural Heritage Preservation: Image captioning could be used to safeguard and disseminate cultural heritage by producing understandable and captivating descriptions for historical artifacts, art works and landscape culture.
9. Education and Learning: Captioned images during the learning process help to better understand the contents of reading and listening materials. This may be very helpful especially to visually handicapped students and the learners of the foreign language.
10. Personal Assistants and Virtual Reality: The integration of image captioning into personal assistants such as Siri or VR experiences will give a more natural interaction than conventional text-based interaction.

In conclusion the area of image captioning is growing quickly and fastly it also should have an effect on many applications belonging to different industries and subject areas. Thus, it promises to promote accessibility, better search, content generation and a more relevant connection between people and machines through closing the gulf of the visual and linguistic universes.

1.5 ORGANIZATION FOR PROJECT REPORT

Chapter 2- Literature survey – This is a part of a chapter where we dig up knowledge that has already been there, using good materials like technical papers, books etc. We want to understand how our market looks at present and what we can address in our project.

Chapter-3 System Development – Here, in this chapter I shall cover the essentials on the project, from requirement analysis to the system design and implementation. Talking about challenges faced after development and strategic repair.

Chapter-4 Testing – The following section explains our testing strategy and procedures that were very thorough. We provide test scenarios together with results that reveal the consistency of the system very clearly.

Chapter 5: Results and Evaluation- Herein lies the concluding chapter whose focus is on results and examines the outcomes as well as compares them with existing remedies. It provides in-depth analysis of our work.

Chapter 6: Conclusions and Future Scope- Concluding our research development and working on improving the project by adding more functionality to widen its scope for future scope.

CHAPTER-2 LITERATURE SURVEY

2.1 OVERVIEW OF THE RELEVANT LITERATURE

Fine-tuned pre-trained used transformer-based neural networks, object detection models, and large-scale image-text datasets. An application of transformer architecture in overcoming distant information processing difficulties inherent on the image captioning process by incorporating cross-memory operation. Evaluation metrics for text-to-image retrieval using large scale databases like MS COCO. Combined bottom-up object detection via neural networks for images captioning and visual query answering. COCO Dataset Encoder decoder based convolutional neural networks for the COCO dataset. Attention aided ASR combined with LSTM models.

Many vision language tasks were done successfully with the object-semantic scheme used as a training technique. Besides, it is perhaps even more impressive because it works better than other approaches for far-away relations, especially if the data is common such as COCO dataset. We propose a novel image description strategy based on ranking that gives rise to superior new evaluation metrics surpassing traditional caption assessments for better labels. With a hierarchical attention based on bottom-up object-level features it was possible to enhance image captioning and visual question asking. Evaluation of encoder–decoder vs better captions for suitability in image captioning using BLEU, METEOR and CIDEr metrics. However, in the course of establishing relevancy or use of technology, accurate and appropriate image captioning could be boosted through speech recognition

Using big data and complex computations is not always possible if you don't have the right resources. Simpler captioning models might be easier to implement. The ranking system doesn't always show if a caption is good or bad - it just compares it to other captions. Generating culturally sensitive, unambiguous captions is tricky, especially if the image could be interpreted in different ways. Speech recognition systems still make mistakes fairly often. This can lead to incorrect captions if you're using speech to text. In conclusion complex captioning systems require more computing power and data. Simpler models are easier to implement widely. There are also challenges around managing ambiguity, cultural norms, and speech recognition errors that can impact caption accuracy.

In this we summarize the main points mentioned in the literature survey which indicates the

major trends and improvement in image captioning area of interest We underline the uses and tools used in literature survey through existing knowledge of image captioning approach We get the suggestion on future research and develops in the area of image captioning and visual assistance.

2.2 KEY GAPS IN THE LITERATURE

Bridging the gap between visual and linguistic understanding: Most image captioning models fail to capture the complete semantic meaning that an image represents and generate engaging captions that reflect the essence of the image. This gap is simply attributed to the natural diversity between visual representations and oral expression, in addition to the intricacies involved in comprehending multifaceted connections and happenings occurring between items and scenes in an image.

1. Incorporating multimodal data: Most of the image captioning models only rely on visual information. Including other modalities like audio or text could provide more context to enhance the correctness and comprehensiveness of captions. Research regarding multimodal fusion techniques as well as the ability to successfully combine data from different sources into an appropriate model is vital.
2. Handling multilingual captioning: However, most image captioning models will find it difficult to generate captions in multiple languages. This is because translation of language is quite complicated and each language has its own subtleties in transmitting pictures. Creation of models for cross-linguistic and cultural translation of captions for global accessibility.
3. Bridging the gap between visual and linguistic understanding: Despite this shortcoming in some leading image captioning algorithms, they manage to succinctly capture the unique essence of each depicted scene. Some part of this gap may be attributable to difference visual/verbalization comprehending the multi-dimensional relations among entities and setting within pictures.
4. Incorporating multimodal data: Much of the image captioning models is derived from visual data. Inclusion of other modes such as sounds and texts will enable capture's to become truthful, meaningful and whole. Investigations on the possibility of signal fusion and multi-modal fusion methods encompassed the capability for processing various types of data and incorporating them in statistical models.
5. Handling multilingual captioning: Nevertheless, many of the image captioning models

do not generate multilingual captions. Therefore, it is difficult considering all these difficulties and the distinctive picture that gets embedded in a language through its pictures. Global accessibility through cross-cultural, trans-lingual captioning models.

6. Addressing bias and fairness: The captions of the biased image captioning models that are derived from biased training data could discriminate unfairly, which causes bias. Nevertheless, it should be acknowledged that more effective means of identifying bias and its counteraction should be devised before the image captions can be considered reasonable by all participants in the project.
7. Enhancing explainability and interpretability: Users need to earn some level or trust factor in them because they also must know a little about the internal workings within an image captioning model in order for them to be acceptable. Adequacy of interpreter model in understanding could help to find out what are the strengths or biases or other errors.
8. Exploring real-world applications: It is sure that many of real-world application areas, including blind accessibility in different environs, will be transformed into a lot of specific fields, like image captioning for example posting pictures, social networking, images search and content creation. Therefore, future researchers should explore this field for addressing this theme. To design new strategies for enhancement of existing image based captioning systems in the practical world environments.
9. Addressing ethical considerations: Privacy, surveillance, and abuse-related ethics revolving around image captioning models. Therefore, it is imperative for us to come up with a code of ethics and also relevant regulations.

CHAPTER-3 SYSTEM DEVELOPMENT

3.1 REQUIREMENT AND ANALYSIS

3.1.1 Software resource

1. IDE: Visual Studio Code
2. Backend Framework: Express (Node.js)
3. DBMS: MongoDB
4. Frontend Framework: React
5. Deep Learning Frameworks
6. Data Preprocessing Tools and Text Processing Libraries
7. Deployment: Hosting Platform: Heroku, AWS, DigitalOcean

3.1.2 Hardware resource -

Development Computers enabled with:

Processor: Multi-core processors (e.g., Intel Core i5 or higher, AMD Ryzen) for efficient multitasking.

RAM: 8 GB or more And SSDs for faster read/write speeds and improved performance.

3.1.3 USES

This project is designed to help the people who have partial blindness by giving them the descriptions of the images that are presented on social media platforms. It applies computer vision and natural language processing to create text explanations of images that, thus, can be used by people who have problems in perceiving the visual content.

1. Image Description Generation- The system takes the photo when a user sees it on social media and processes it using computer vision algorithms automatically. CNNs or any other similar models are used to obtain the features of the image that are of the most importance. These features automatically gather some of the essential components of the image such as the objects, scenes, and the contextual information. The extracted features are transferred to a neural network model that is trained for image captioning and thus, the features are passed through it. This model creates a text that describes the image content in a language that is similar to the way a person speaks. The caption that is produced is a complete description of the picture, which includes the objects, activities, people, and any other visual elements that are in the picture. This illustration is therefore useful for users who have partial blindness and to help them to comprehend

and interpret the content of the image which they cannot rely on only on the visual impaired.

2. Besides generating captions for images, the project also gives additional information about newly introduced concepts or words that are used in the captions. Natural language processing methods, such as NER, or entity linking, are utilized to pinpoint the specific entities or concepts which are mentioned in the captions. After the user has given the system a hint, the system will find the required information about the entities from a knowledge base or external sources like online databases or encyclopedias.

The extra data may be the definitions, explanations, related sources, or multimedia content like images or videos that give more meaning and understanding to the said concepts. Thus, the feature is designed to provide users with partial blindness, not only with images descriptions but also with the additional information that will make their learning much better.

3.1.4 ANALYSIS

This project involves three steps

1. Image Feature Extraction using CNN:Image Feature Extraction using CNN. The primary step is the extraction of the important features from the input image. The mentioned is usually carried out with a convolutional neural network (CNN). CNNs are the perfect tools for image feature extraction tasks because they are able to automate the process of learning hierarchical representations of image features through the application of layers of convolutions and pooling operations. Pre-trained CNN models, e.g. VGG, ResNet, or Inception, are usually used for this. These algorithms, which have been trained on large-scale images over large-scale image datasets, are also of high importance and have the capability to extract high-level features from images correctly. The output of the CNN is a feature vector that is the outcome of the best significant features of the input image. This feature vector is basically a code of sorts which represents the information about the objects, shapes, textures and other visual components of the image.
2. Text-to-Speech (TTS) Conversion- After the caption is produced, the following is to be done that is to, the textual description be turned into speech. Text-to-Speech (TTS) synthesis is the technology of creating voiced language from written text. TTS systems typically consist of two main components: a text analysis component, which is converting the input text into a phonetic or linguistic representation, and a speech synthesis component, which is the speech waveform generation. Recently, deep learning-based TTS models like WaveNet or Tacotron have been able to provide the speech quality and naturalness that are much greater than those of the traditional TTS

systems. The produced caption is then sent through the TTS model to generate the speech that will be the corresponding spoken audio. The TTS model acquires the ability to produce speech that corresponds closely to the meaning and the manner of the input text.

3. **Caption Generation using Image Features:**Caption Generation using Image Features. The features extracted from the image are then sent through a different neural network, usually an RNN or a Transformer model, to make a textual description or a caption of the image. The image is the first input or context for the caption generation model of the sentences. The model then iteratively produces words one after the other, using the previously generated words and the image features as the conditioning factors. During the training, the model is trained to maximize the probability of generating the ground truth caption given the image features of the input image. The techniques of beam search or sampling can be used to produce varied and fluently captions.

3.2 PROJECT DESIGN AND ARCHITECTURE

3.2.1 Level-1 DFD

1. This DFD represents the whole project. It has following features:
2. User: Any individual or Organization using this Web Application.
3. Application Interface: It will be the face of Web Application consisting of options like
4. upload an image, get a caption and get speech.
5. Receive Image and Upload: This will upload images to the database.
6. Generate Caption: This will take the stored images and will generate text according to the features of images. And then it will upload these texts to the database.
7. Convert to Speech: This will take the stored text from the database and then convert it into speech. Conversion will be done with the help of Google's Text to Speech API.
8. Return Caption And Speech: This will return text and speech for the input image as an outcome to the user.

The whole process takes place in this way:

1. First User gives the image as an input and then the image is uploaded to the database for processing.
2. During processing feature based text is generated and get stored in the database.
3. This text is further converted into text and stored in the database.
4. In the end stored text and speech is passed to the user through the user interface of the web application

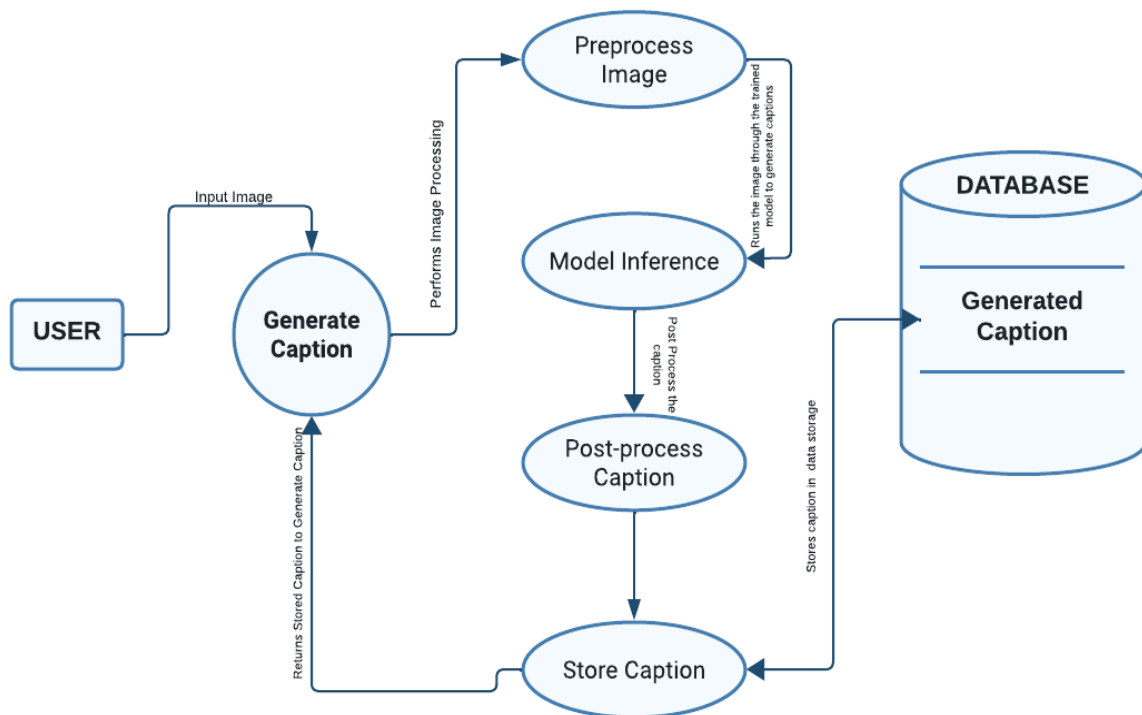


Figure.1- It represents Level-1 DFD

3.2.2 Level-2 DFD:

1. This DFD represents a particular part of the project. This part has role of caption Generation which is done with the help of following features
2. Preprocess Images: In this, images are resized so that it remains consistent for further processing. To introduce some variability during training, a random crop of 224X224 is done. Preprocessing is done on both training and testing data.
3. Model Inference: It is shown using a trained model in the format of Encoder and Decoder.
4. Encoder is A CNN model. Decoder is Rnn.
5. Post-Processing: Generated tokens of words are converted into sentences by mapping them with vocabulary.
6. The words are then concatenated to form a sentence for the visual inspection.
7. Store Caption: The sentences created are stored in a database for further processing.

The above given process takes place in the following manner:

1. This DFD is basically representing the DeepLearning part of the project in which the features of the image are being extracted with the help of a pretrained model of CNN ResNet-50.
2. Then the output of this CNN model is being processed with RNN for Text Generation.
3. In the end the output is stored and displayed to the user.

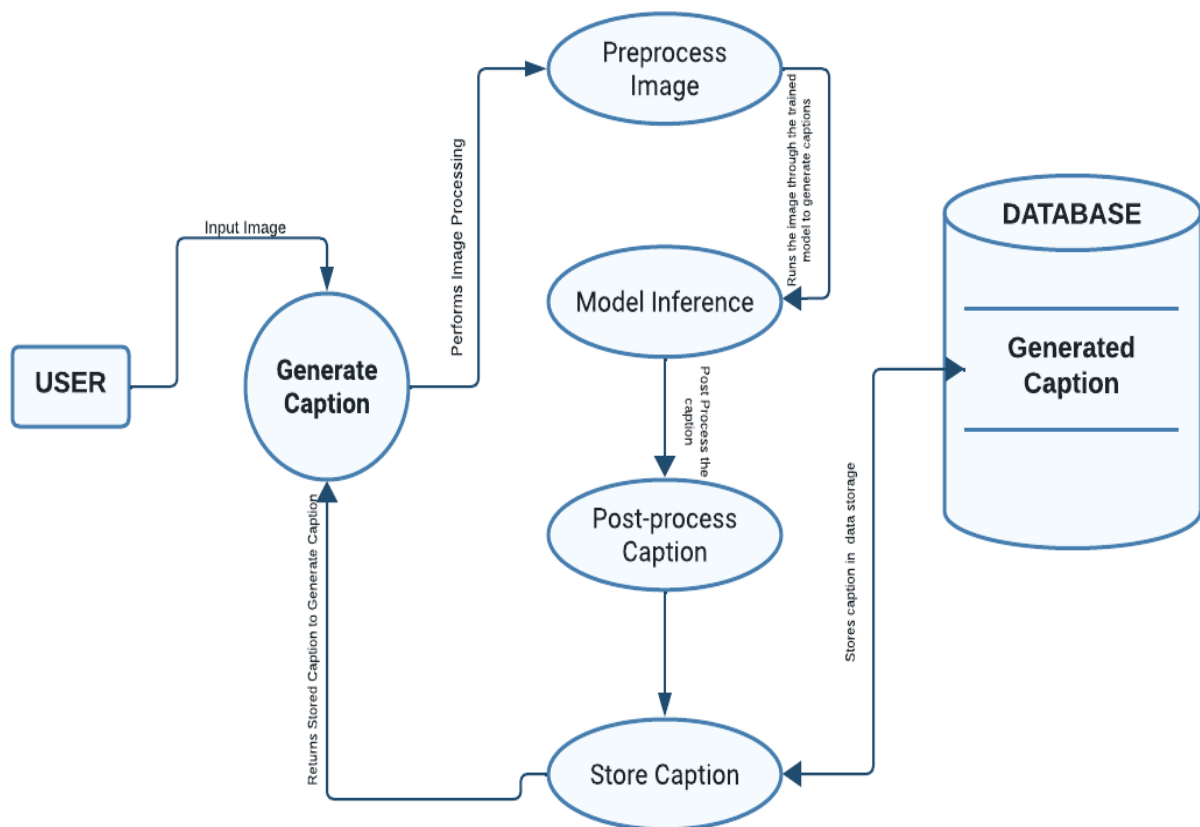


Figure.2 - It represents the level-2 DFD

3.2.3 Encoder(CNN)

ResNet-50

1. Input: RGB images are usually called the three-dimensional array, the first two dimensions are the height and width and the third dimension is the color channels (Red, Green, and Blue). In the ResNet-50 model, the input images are usually the standard size which is 224x224 pixels. This standardization is the reason why a similar model is possible to be used and enables the comparison between different models easily.

2. Convolution Layers: The first convolutional layer in ResNet-50 is instrumental in the process of extracting the low-level features from the input image. Through the application of a big kernel size of 7x7 and a stride of 2, the network can grasp more spatial information and at the same time the spatial dimensions of the feature maps are decreased at the early stage of the network. This leads to a lower computational complexity and hence the following layers of the network can then concentrate on the more high-level and abstract features.

3. Residual Blocks: The residual blocks are the main components of ResNet architectures. Each residual block consists of several convolutional layers that are then connected to the shortcut. The shortcut connection is a conduit that makes the gradient to pass through the block without being affected by the weights of the intermediate layers. Such an approach to gradient decay helps in dealing with the vanishing gradient problem which can occur in very deep networks during training. Through the introduction of the network, the ResNet architectures are able to keep the residual mappings, which, in turn, makes the network of much deeper networks than the previous architectures like plain convolutional neural networks

4. Average Pooling: Convolutional layers are followed by the average pooling which is then applied to the feature maps in order to diminish the spatial dimensions. On the contrary to max pooling, which is the maximum value within each pooled region, average pooling calculates the mean value. The operation offers a way to condense the information in the feature maps while keeping the significant spatial relationships intact. In ResNet-50, global average pooling is generally used, where the spatial dimensions are decreased to a single value for each channel and thus the feature vector has a fixed size on the whole irrespective of the input image size.

5. Fully Connected Layer: In most of the typical classification tasks, a fully connected layer

is appended right after the convolutional layers and the pooling layers to generate the final classification scores. Nevertheless, in some cases, for instance, in the field of feature extraction or transfer learning, the fully connected layer is not necessary. This is the reason the network can yield a feature tensor instead of the classification scores, which could be more helpful for tasks like feature extraction, embedding, or further processing with more layers or models.

In general, ResNet-50 is a strong convolutional neural network architecture that has been everywhere and proved to be very effective in different computer vision tasks such as the image classification, object detection, and semantic segmentation. Its modular design, which has residual blocks and skip connections, helps to train deep networks easily and also enables transfer learning for a variety of applications.

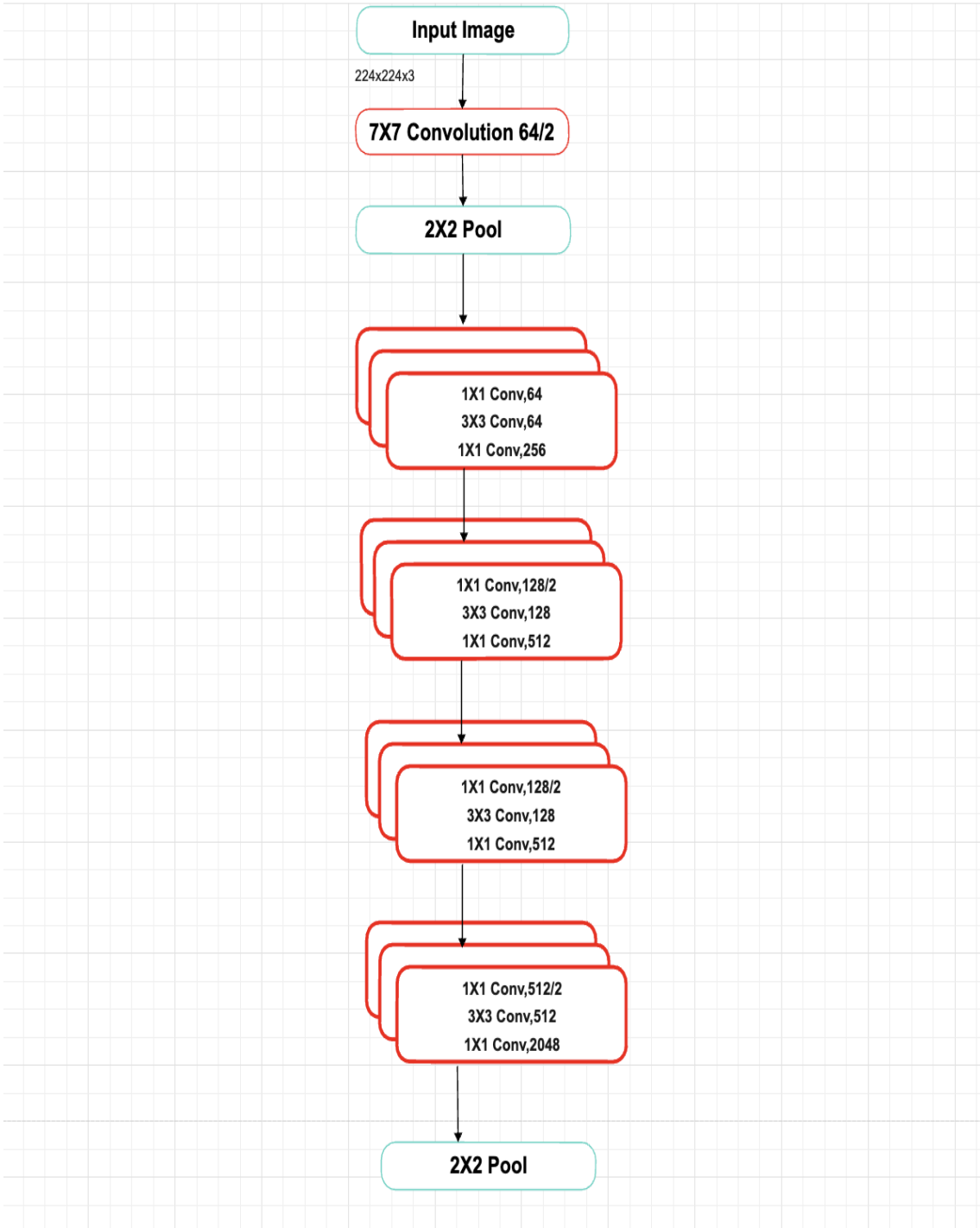


Figure.3 - Encoder(CNN) ResNet-50

3.2.4 Decoder(RNN):

1. Input: In the generation of sequences tasks, such as language modeling or text generation, the input usually is a sequence of words that are represented by their corresponding indices. These indices are usually got through tokenizing the input text and then by mapping each word to a unique integer index.
2. Embedding Layer: The embedding layer converts the input word indices into a dense vector of fixed size. Every word index is assigned to a high-dimensional vector which is a representation of similar meanings or context of the words. This compact representation of the semantics of the words enables the model to understand the input sequence and, thus, it learns the meaningful representations for the input sequence.
3. LSTM Layer: The LSTM (Long Short-Term Memory) layer is a special type of RNN architecture which is used to process the sequential data and also, it solves the vanishing gradient problem. LSTMs have a hidden state that records the information of the past time steps and at each time step the hidden state is updated according to the current input and the previous hidden state. This makes the model to be able to recognize and understand the long-lasting connections and the sequences in the input data.
4. Hidden State Initialization: Initially, when the input sequence is being processed, the hidden state of the LSTM layer is created with the zero-valued vectors. When the model processes each word in the input sequence, the hidden state is being updated according to the input word embeddings and the previous hidden state. This hidden state is the model's memory that has an internal mechanism of storage and captures the contextual information which is the whole input sequence.
5. Linear Layer: The input sequence is transformed by the LSTM layer into the hidden state representation which then passes into the linear layer. The linear layer then converts the high-dimensional hidden state features into a space with a dimensionality equal to the vocabulary size. For each word in the vocabulary, the linear layer gives a score which is the probability that the word is the next word in the sequence. This step is, in a way, a classification task that uses the vocabulary to predict the next word.
6. Loss Calculation: The scores predicted from the linear layer are compared to the actual word indices with the cross-entropy loss function. This loss function calculates the gap between the predicted probability distribution over the vocabulary and the true distribution (one-hot encoded version of the actual next word). The model is trained to

reduce the loss which is a sum of these weights, by using backpropagation to adjust its parameters (embedding weights, LSTM weights, and linear layer weights).

7. Inference: The model that was trained is employed to produce text line by line during the inference. The procedure begins with the input of the first data, which is given to the model for it to start working. The model produces a probability distribution over the words in the vocabulary for the next word, and a word is picked from this distribution. The sampled word is then fed back into the model as the next input, and the process keeps on with the same word until the predefined token size is reached. Finally, a new sentence is formed or a maximum sentence length is attained.

In a nutshell, this architecture allows the model to learn to produce sequential text that is both coherent and contextually relevant to the inputted word sequence, This is of great help for language modeling, text generation, and machine translation.

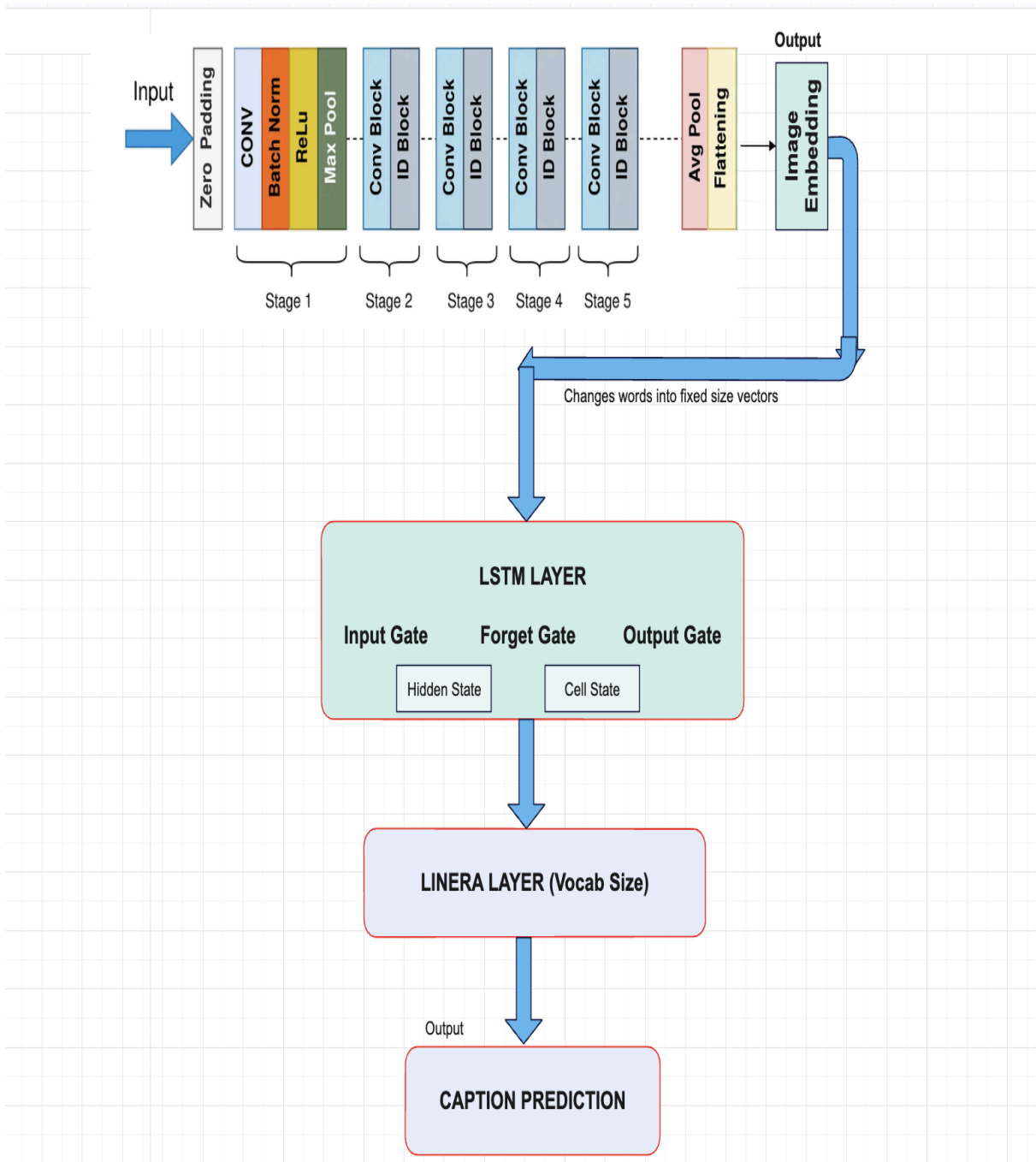


Figure.4-Encoder(CNN) +Decoder(RNN)

3.3 Data Preparation:

1. Downloading COCO Dataset:

This is done using COCO API.

```
Download the required data : Annotations,Captions,Images

import os
import sys
from pycocotools.coco import COCO
import urllib
import zipfile

os.makedirs('opt', exist_ok=True)
os.chdir( '/content/opt' )
!git clone 'https://github.com/cocodataset/cocoapi.git'

Cloning into 'cocoapi'...
remote: Enumerating objects: 975, done.
remote: Total 975 (delta 0), reused 0 (delta 0), pack-reused 975
Receiving objects: 100% (975/975), 11.72 MiB | 25.82 MiB/s, done.
Resolving deltas: 100% (576/576), done.

Download the Annotations and Captions :

[ ] os.chdir('/content/opt/cocoapi')

# Download the annotation :
annotations_trainval2017 = 'http://images.cocodataset.org/annotations/annotations_trainval2017.zip'
image_info_test2017 = 'http://images.cocodataset.org/annotations/image_info_test2017.zip'

urllib.request.urlretrieve(annotations_trainval2017 , filename = 'annotations_trainval2017.zip' )
urllib.request.urlretrieve(image_info_test2017 , filename= 'image_info_test2017.zip' )

('image_info_test2017.zip', <http.client.HTTPMessage at 0x7e01881ea3b0>)
```


2.Extracting Annotations and Captions

```
▶ with zipfile.ZipFile('annotations_trainval2017.zip' , 'r') as zip_ref:
    zip_ref.extractall( '/content/opt/cocoapi' )

    try:
        os.remove( 'annotations_trainval2017.zip' )
        print('zip removed')
    except:
        None

with zipfile.ZipFile('image_info_test2017.zip' , 'r') as zip_ref:
    zip_ref.extractall( '/content/opt/cocoapi' )

    try:
        os.remove( 'image_info_test2017.zip' )
        print('zip removed')
    except:
        None

⇒ zip removed
zip removed
```

3.Initialize and Verify loaded Data

```
▶ os.chdir('/content/opt/cocoapi/annotations')
# initialize COCO API for instance annotations
dataType = 'val2017'
instances_annFile = 'instances_{}.json'.format(dataType)
print(instances_annFile)
coco = COCO(instances_annFile)

# initialize COCO API for caption annotations
captions_annFile = 'captions_{}.json'.format(dataType)
coco_caps = COCO(captions_annFile)

# get image ids
ids = list(coco.anns.keys())

⇒ instances_val2017.json
loading annotations into memory...
Done (t=0.69s)
creating index...
index created!
loading annotations into memory...
Done (t=0.05s)
creating index...
index created!
```

4.Vocabulary Creation:

The Vocabulary Class is created for training captions.It maps words to indices,which is very important for feeding these inputs to models for processing.

5.Data Loaders:

This class facilitates the data preprocessing and loading for training and testing.It uses COCO API for obtaining caption,image paths and various important information.

Creating DataLoader:

```
import sys
from pycocotools.coco import COCO
!pip install nltk
import nltk
nltk.download('punkt')
from torchvision import transforms

# Define a transform to pre-process the training images.
transform_train = transforms.Compose([
    transforms.Resize(256),           # smaller edge
    transforms.RandomCrop(224),      # get 224x224
    transforms.RandomHorizontalFlip(), # horizontally
    transforms.ToTensor(),           # convert the
    transforms.Normalize((0.485, 0.456, 0.406), # normalize i
                        (0.229, 0.224, 0.225))]

# Set the minimum word count threshold.
vocab_threshold = 8

# Specify the batch size.
batch_size = 200

# Obtain the data loader.
data_loader_train = get_loader(transform=transform_train,
                                mode='train',
                                batch_size=batch_size,
                                vocab_threshold=vocab_threshold,
                                vocab_from_file=False,
                                cocoapi_loc = '/content/opt')
```

6.Image Transformation: It is a pipeline which involves resizing,normalization and cropping. This guarantees compatibility of input image for pretrained model ResNet-50.

7.Batch Generation: DataLoader instance is created for iterating over batches of captions and images during training and testing.

Code:

```
def get_loader(transform,mode='train',
```

```

batch_size=1,

vocab_threshold=None, vocab_file='./vocab.pkl',

start_word("<start>"),

end_word("<end>"),

unk_word("<unk>"),

vocab_from_file=True,num_workers=0,

cocoapi_loc='/opt'):

    assert mode in ['train', 'test'], "mode must be one of 'train' or 'test'."

    if vocab_from_file==False: assert mode=='train', "To generate vocab from captions file,
must be in training mode (mode='train')."

    # Based on mode (train, val, test), obtain img_folder and annotations_file.

    if mode == 'train':

        if vocab_from_file==True: assert os.path.exists(vocab_file), "vocab_file does not exist.
Change vocab_from_file to False to create vocab_file."

        img_folder = os.path.join(cocoapi_loc, 'cocoapi/images/train2017/')

annotations_file = os.path.join(cocoapi_loc,
'cocoapi/annotations/captions_train2017.json')

    if mode == 'test':

        assert batch_size==1, "Please change batch_size to 1 if testing your model."

        assert os.path.exists(vocab_file), "Must first generate vocab.pkl from training data."

        assert vocab_from_file==True, "Change vocab_from_file to True."

        img_folder = os.path.join(cocoapi_loc, 'cocoapi/images/test2017/')

                                annotations_file = os.path.join(cocoapi_loc,
'cocoapi/annotations/image_info_test2017.json')

    # COCO caption dataset.

    dataset = CoCoDataset(transform=transform,

```

```
mode=mode,  
batch_size=batch_size,  
vocab_threshold=vocab_threshold,  
vocab_file=vocab_file,  
start_word=start_word,  
end_word=end_word,  
unk_word=unk_word,  
annotations_file=annotations_file,  
vocab_from_file=vocab_from_file,  
img_folder=img_folder)
```

```
if mode == 'train':
```

```
    # Randomly sample a caption length, and sample indices with that length.
```

```
    indices = dataset.get_train_indices()
```

```
    # Create and assign a batch sampler to retrieve a batch with the sampled indices.
```

```
    initial_sampler = data.sampler.SubsetRandomSampler(indices=indices)
```

```
    # data loader for COCO dataset.
```

```
    data_loader = data.DataLoader(dataset=dataset,
```

```
                                  num_workers=num_workers,
```

```
                                  batch_sampler=data.sampler.BatchSampler(sampler=initial_sampler,
```

```
                                  batch_size=dataset.batch_size,
```

```
                                  drop_last=False))
```

```
else:
```

```
    data_loader = data.DataLoader(dataset=dataset,
```

```
                                  batch_size=dataset.batch_size,
```

```
                                  shuffle=True,
```

```
num_workers=num_workers)
```

```
return data_loader
```

In total we can say that,

The Data preprocessing pipeline ensures a model of having properly formatted input during training and testing which facilitates effective learning and caption generation.

3.4 Implementation (include code snippets, algorithms, tools and techniques, etc.)

First we have to plot a sample images

```
[ ] import matplotlib.pyplot as plt
import skimage.io as io
import numpy as np
%matplotlib inline


[ ] #Pick a random annotation id and display img of that annotation :
ann_id = np.random.choice( ids )
img_id = coco.anns[ann_id]['image_id']
img = coco.loadImgs( img_id )[0]
url = img['coco_url']
print(url)
I = io.imread(url)
plt.imshow(I)

# Display captions for that annotation id :
ann_ids = coco_caps.getAnnIds( img_id )
print('Number of annotations i.e captions for the image: ' , ann_ids)
print()
anns = coco_caps.loadAnns( ann_ids )
coco_caps.showAnns(anns)

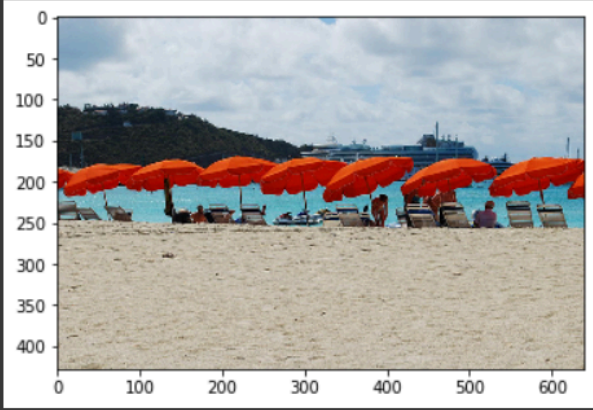
http://images.cocodataset.org/val2014/COCO\_val2014\_000000350000.jpg
Number of annotations i.e captions for the image: [103372, 104554, 111691, 112603, 112669]

There are orange beach umbrellas lined up on a beach
```

Output of image with caption

 http://images.cocodataset.org/val2014/COCO_val2014_000000350000.jpg
Number of annotations i.e captions for the image: [103372, 104554, 111691, 112603, 112669]

There are orange beach umbrellas lined up on a beach
some chairs water sand and orange umbrellas and sunny day
tourists at a beach cove under umbrellas by a cruise ship
A row of umbrellas lined up at the beach.
A number of red umbrellas and beach chairs near the ocean.



Coco dataset images using python

```
[ ] os.chdir('/content/opt/cocoapi')  
  
train2014 = 'http://images.cocodataset.org/zips/train2014.zip'  
test2014 = 'http://images.cocodataset.org/zips/test2014.zip'  
val2014 = 'http://images.cocodataset.org/zips/val2014.zip'  
  
urllib.request.urlretrieve(train2014, 'train2014')  
urllib.request.urlretrieve(test2014, 'test2014')  
#urllib.request.urlretrieve(val2014, 'val2014')  
  
( 'test2014', <http.client.HTTPMessage at 0x7f6adaeaf588> )
```

unzip the download image zip files

```
[ ] os.chdir('/content/opt/cocoapi')
    with zipfile.ZipFile( 'train2014' , 'r' ) as zip_ref:
        zip_ref.extractall( 'images' )

    try:
        os.remove( 'train2014' )
        print('zip removed')
    except:
        None

os.chdir('/content/opt/cocoapi')
with zipfile.ZipFile( 'test2014' , 'r' ) as zip_ref:
    zip_ref.extractall( 'images' )

try:
    os.remove( 'test2014' )
    print('zip removed')
except:
    None

zip removed
```

Now we explore the data loader

> Vocabulary.py

[] ↪ 1 cell hidden

> data_loader.py

[] ↪ 1 cell hidden

> Dataloader creation

[] ↪ 4 cells hidden

Implement the RNN encoder

```
class DecoderRNN(nn.Module):
    def __init__(self, embed_size, hidden_size, vocab_size, num_layers=1):
        super(DecoderRNN, self).__init__()
        self.embed_size = embed_size
        self.hidden_size = hidden_size
        self.vocab_size = vocab_size
        self.num_layers = num_layers
        self.word_embedding = nn.Embedding(self.vocab_size, self.embed_size)
        self.lstm = nn.LSTM(input_size = self.embed_size,
                            hidden_size = self.hidden_size,
                            num_layers = self.num_layers,
                            batch_first = True)
        self.fc = nn.Linear(self.hidden_size, self.vocab_size)

    def init_hidden(self, batch_size):
        return (torch.zeros(self.num_layers, batch_size, self.hidden_size).to(device),
                torch.zeros(self.num_layers, batch_size, self.hidden_size).to(device))

    def forward(self, features, captions):
        captions = captions[:, :-1]
        self.batch_size = features.shape[0]
        self.hidden = self.init_hidden(self.batch_size)
        embeds = self.word_embedding(captions)
        inputs = torch.cat((features.unsqueeze(dim=1), embeds), dim=1)
        lstm_out, self.hidden = self.lstm(inputs, self.hidden)
        outputs = self.fc(lstm_out)
        return outputs
```

```
def Predict(self, inputs, max_len=20):
    final_output = []
    batch_size = inputs.shape[0]
    hidden = self.init_hidden(batch_size)

    while True:
        lstm_out, hidden = self.lstm(inputs, hidden)
        outputs = self.fc(lstm_out)
        outputs = outputs.squeeze(1)
        _, max_idx = torch.max(outputs, dim=1)
        final_output.append(max_idx.cpu().numpy()[0].item())
        if (max_idx == 1 or len(final_output) >= 20):
            break

        inputs = self.word_embedding(max_idx)
        inputs = inputs.unsqueeze(1)
    return final_output
```



```

num_layers = 1
num_epochs = 4
print_every = 150
save_every = 1
vocab_size = len(data_loader_train.dataset.vocab)
total_step = math.ceil( len(data_loader_train.dataset.caption_lengths) / data_loader_train.batch_sampler.batch_size )

decoder = DecoderRNN( embed_size , hidden_size, vocab_size ,num_layers)
criterion = nn.CrossEntropyLoss()
lr = 0.001
all_params = list(decoder.parameters()) + list( encoder.embed.parameters() )
optimizer = torch.optim.Adam( params = all_params , lr = lr )

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model_save_path = '/content/drive/My Drive/Colab Notebooks/ComputerVision/RNN_LSTM/image_caption/CVND---Image-Captioning-Project/checkpo
os.makedirs( model_save_path , exist_ok=True)

# Save the params needed to created the model :
decoder_input_params = { 'embed_size' : embed_size ,
                        'hidden_size' : hidden_size ,
                        'num_layers' : num_layers,
                        'lr' : lr ,
                        'vocab_size' : vocab_size
                      }

with open( os.path.join(model_save_path , 'decoder_input_params_12_20_2019.pickle'), 'wb') as handle:
    pickle.dump(decoder_input_params, handle, protocol=pickle.HIGHEST_PROTOCOL)

```

```

import sys
for e in range(num_epochs):
    for step in range(total_step):
        indices = data_loader_train.dataset.get_train_indices()
        new_sampler = data.sampler.SubsetRandomSampler( indices )
        data_loader_train.batch_sampler.sampler = new_sampler
        images,captions = next(iter(data_loader_train))
        images , captions = images.to(device) , captions.to(device)
        encoder , decoder = encoder.to(device) , decoder.to(device)
        encoder.zero_grad()
        decoder.zero_grad()
        features = encoder(images)
        output = decoder( features , captions )
        loss = criterion( output.view(-1, vocab_size) , captions.view(-1) )
        loss.backward()
        optimizer.step()
        stat_vals = 'Epochs [%d/%d] Step [%d/%d] Loss [%.4f] ' % ( e+1,num_epochs,step,total_step,loss.item() )
        if step % print_every == 0 :
            print(stat_vals)
            sys.stdout.flush()
        if e % save_every == 0:
            torch.save( encoder.state_dict() , os.path.join( model_save_path , 'encoderdata_{}.pkl'.format(e+1) ) )
            torch.save( decoder.state_dict() , os.path.join( model_save_path , 'decoderdata_{}.pkl'.format(e+1) ) )

```

Load the saved checkpoint

```
[ ] model_save_path = '/content/drive/My Drive/Colab Notebooks/ComputerVision/RNN_LSTM/image_caption/CVND---Image-Captioning-Project/checkpoint'
os.makedirs(model_save_path, exist_ok=True)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

with open(os.path.join(model_save_path, 'decoder_input_params_12_19_2019.pickle'), 'rb') as handle:
    decoder_input_params = pickle.load(handle)

embed_size = decoder_input_params['embed_size']
hidden_size = decoder_input_params['hidden_size']
vocab_size = decoder_input_params['vocab_size']
num_layers = decoder_input_params['num_layers']

encoder = EncoderCNN(embed_size)
encoder.load_state_dict(torch.load(os.path.join(model_save_path, 'encoderdata_{}.pkl'.format(1))))

decoder = DecoderRNN(embed_size, hidden_size, vocab_size, num_layers)
decoder.load_state_dict(torch.load(os.path.join(model_save_path, 'decoderdata_{}.pkl'.format(1))))
```

Create data loader for testing data

```
from torchvision import transforms

# Define a transform to pre-process the training images.
transform_test = transforms.Compose([
    transforms.Resize(256), # smaller edge of image resized to 256
    transforms.RandomCrop(224), # get 224x224 crop from random location
    transforms.RandomHorizontalFlip(), # horizontally flip image with probability=0.5
    transforms.ToTensor(), # convert the PIL Image to a tensor
    transforms.Normalize((0.485, 0.456, 0.406), # normalize image for pre-trained model
                        (0.229, 0.224, 0.225))]

# Obtain the data loader.
data_loader_test = get_loader(transform=transform_test,
                              mode='test',
                              cocoapi_loc = '/content/opt')

data_iter = iter(data_loader_test)
```

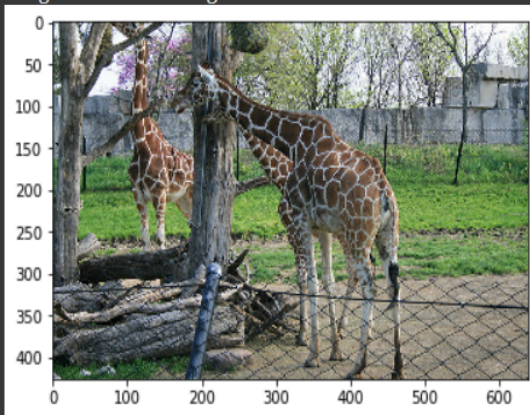
Vocabulary successfully loaded from vocab.pkl file!

```
def get_sentences( original_img, all_predictions ):
    sentence = ' '
    plt.imshow(original_img.squeeze())
    return sentence.join([data_loader_test.dataset.vocab.idx2word[idx] for idx in all_predictions[1:-1] ] )
```

```
[ ] encoder.to(device)
    decoder.to(device)
    encoder.eval()
    decoder.eval()
    original_img , processed_img = next( data_iter )

    features = encoder(processed_img.to(device) ).unsqueeze(1)
    final_output = decoder.predict( features , max_len=20)
    get_sentences(original_img, final_output)
```

'a giraffe standing in a field with trees .'



3.4.1 Frontend Components:

1.React Components:

1. Home- This is the front page of the application. On the about page, it displayed a rundown of the application functionality, giving users a sense of what it is for. Moreover, it leverages a link or a button which points to the registration page of the system allowing customers to begin their journey with the performing the primary action. The layout could be a combination of various UI elements like headings, text paragraphs and even images that together create a compelling introduction.
2. login- This part displays a login form where the users can input their credentials to verify themselves and thus can be able to access the application. The template is usually designed with spaces for the user's email and password, and a click on/ enter button is usually included. One client-side validation method could be to check that the user will enter correct data before the form is submitted. Once the user inputs their credentials, the component proceeds with the authentication process by sending the data to the backend for verification.
3. Register- The Register component will display the registration form where new users can sign up for the application by filling in their details. The form fields will contain: first name, last name, email, and password required fields. Clients-side validation, like the login form, guarantees the completion of data correctness within all mandatory fields before submission. After the applicant fills in the registration form and submits it, the component then initiates the registration process, sending the data to the backend for validation and storage
4. ImageCaptionGenerator- This part is concerned about how the functionality is built within the application for images to be loaded and they are presented to users as they trigger the generation of image captions. The main character is generally a user input or drag and drop area that allows them to choose or drag the images which they want to submit are commonly available. The component could also show the user a preview of the uploaded image, after the user has selected an image. When the user confirms it, the component will send the uploaded image data to the backend side to process, so it is the trigger for the captions generation by AI algorithms.
5. Results- The Result component is the component that displays the output of the image caption generation process. It usually arrives with the respective picture that has been uploaded to the model, accompanied by its corresponding caption as well. One extra

feature, for example, offering text-to-speech functionalities along with this caption, is also likely to be part of this. The users can communicate with the results displayed, they may even distribute them or perform more actions according to the generated content.

6. Loader- The Loader component shows a loading animation to let the user know that the process is still running, for example, when you are uploading an image or generating a caption. It gives the users an impression that their application is still functional as the user is being given feedback indicating that there is no freeze after the task is being worked on. The loader could for instance be made dynamic by appearing every time the user submits a form which in return cause some processing time to the backend.

2. React router - React Router is one of the most popular libraries that helps developers fully manage on navigation and routing for our React apps. That gives developers the possibility to define many circuits within their application which are linking to the required component in each specific case. React Router makes it possible to have multiple views or pages in the Single Page Applications (SPAs) without the whole page reload when navigating between them. Here's a more detailed explanation of React Router: React Router allows cases to be described in a declarative manner using JSX notation through which the app's navigation structure can be defined. Developers can build routes and make them function using Routers and Route components, which are respectively nested within the aptly named Router component. Routes are normally specified in the main component of the application, for instance, in App. In web applications, we employ router that enables us to recognize the JS and issue the appropriate URL for the rendering of components based on URL paths.

image_caption_generator-master

```

1 import React from 'react';
2 import styles from '../styles.module.css';
3 import projLogoVideo from '../videos/proj_logo.mp4';
4 import projectLogoVideo from '../videos/project_logo.mp4';
5 import catImage from '../background/cat.png';
6 import { Link } from 'react-router-dom';
7
8
9
10
11 const Home = () => {
12
13   return (
14     <div className={styles.topContainer}>
15       <video id="bg-video" autoPlay loop muted>
16         <source src={projectLogoVideo} type="video/mp4" />
17       </video>
18       <div className={styles.login}>
19         <Link className={styles.logbtn} to="/login">
20           Login
21         </Link>
22       </div>
23       <div className={styles.info}>
24         <h2>The Next-Gen Image Insight Generation</h2>
25         <div className={styles.description}>
26           <img className={styles.cat} src={catImage} alt="Cat" />
27           <h3>
28             Let our AI do the talking! Generate captions for your images in
29             seconds...
30           </h3>
31           <h3>So come, upload your images and listen to them come to life</h3>
32         </div>
33         <div className={styles.getStarted}>
34           <Link className={styles.btn} to="/upload">
35             Get Started!
36           </Link>
37         </div>
38       </div>
39     </div>
40   );
41 };

```

PROBLEMS OUTPUT TERMINAL PORTS CODE REFERENCE LOG DEBUG CONSOLE

Connection Successful

Ln 7, Col 1 Spaces: 2

image_caption_generator-master

```

1 import React, { useState, useEffect } from "react";
2
3 import "../index.css"
4 // import Topbar from "../Topbar";
5 import Topbar from "../Topbar";
6 // import back2 from "../background/back2.jpg"
7 import back2 from "../background/back2.jpg"
8 import Result from "../Result";
9 import Loader from "../Loader";
10 // BHushan
11
12
13 const ImageCaptionGenerator = () => {
14
15   const [selectedFile, setSelectedFile] = useState("");
16   // const [selectedFile, setSelectedFile] = useState("No file choosen");
17   const [preview, setPreview] = useState("");
18   const [bool, setBool] = useState(false);
19
20   const [name, setName] = useState("");
21
22   const handleImageChange = (event) => {
23     // const img = event.target.files[0].name;
24     const img = event.target.files[0];
25     setSelectedFile(img);
26   };
27
28   const handleGenerateCaption = (event) => {
29
30     if (selectedFile)
31       setBool(true);
32     else {
33       window.alert("Select image first!");
34     }
35   };
36
37   const fetchUser = async () => {
38     const url = `http://localhost:8000/fetchnotes`;
39
40

```

PROBLEMS OUTPUT TERMINAL PORTS CODE REFERENCE LOG DEBUG CONSOLE

Connection Successful

Ln 20, Col 42 Spaces: 2

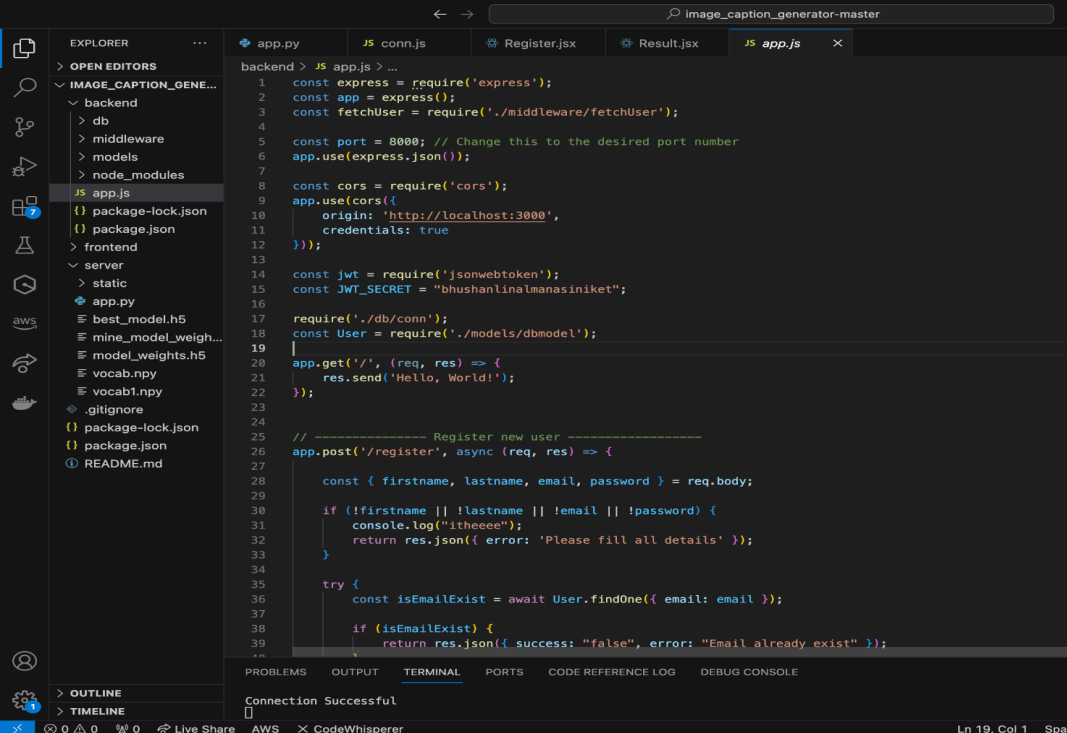
3.4.2 Backend Components:

Node.js with Express: Server configuration starting from 'app.js': Your files layout should be: app. It is the portal that begins the Node.js application. Express runs at that which is controlled through the app file which was created. The configuration file has the settings like port number, middleware, and route handling which are controlled here. The CORS (Cross-Origin Resource Sharing) middle-ware is set up to allow requests from different origins, thus the client applications can communicate with the server. The application of body parsers as a middleware may be configured to generate JSON file on the request from the external body so that the request data can be worked appropriately. Routes are defined in these files, where the URL paths are applied and the orchestration of controllers is managed in respect of incoming requests.

Middleware - This file encompasses these middleware functions that are used to filter the incoming requests before leading them to the designated handlers. Here, users can be asked 'fetchUser' ;(makes sense only if this is in an example when the user or 'fetchUser' can be asked a question) J's middleware is the one that is in charge of checking and decoding the JWT (JSON Web Token) tokens for user authentication. If the request is not protected, this middleware is executed to authenticate the request with a valid JWT token and everything is ok. If a valid token is existed, on the other hand, it is being decoded to extract user's info like user ID and role, which will take a part in further route processing. In case the token is not valid or is missing, the middleware may send an error response that informs the user that he/she is not authenticated and access to the route is not allowed.

Database Models- Through this file one shall define his/her MongoDB schema for users. In the schema, his/her will be all the user documents stored, their structure will be specified. It usually relies on an Object Data Modeling (ODM) library such as Mongoose for specifying the schema and interacting with the MongoDB database. The schema will typically comprise of fields like username, email, password (that is hashed ideally to keep the data secure), and any other relevant user information. Furthermore, the file can include methods or hooks to operate user authentication, data validation, and both pre and post save hooks covering data manipulation.

Routes - Authentication Routes with these routes, user registration and login authentications are done by MongoDB. Besides that, with a POST request to the "/register" route, the server examines and confirms the user data entered, creates a new user document on the basis of the given information, and stores it into the MongoDB database. The same principle applies when a user sends a POST request to the '/signin' route with his/her credentials; the server checks the credentials against the data that is stored in the database. If the credential credential is valid, then it recalls a JWT token to the client that will allow for future authentications. Protected Route method is authorization protected where access to it via valid JWT token will require authentication. Whenever a request is made to this route, the 'fetchUser' program is launched. The middleware returns to verify and validate the JWT token, which is part of the request. If the token is valid and authentication of the user occurs, server searches for user information (in the form of notes or other data) using the ID as the index, and sends it back as a server response.



```
backend > $ npm run dev
1  const express = require('express');
2  const app = express();
3  const fetchUser = require('./middleware/fetchUser');
4
5  const port = 8000; // Change this to the desired port number
6  app.use(express.json());
7
8  const cors = require('cors');
9  app.use(cors({
10   origin: 'http://localhost:3000',
11   credentials: true
12 }));
13
14 const jwt = require('jsonwebtoken');
15 const JWT_SECRET = "bhushanlinalmanasiniket";
16
17 require('./db/conn');
18 const User = require('./models/dbmodel');
19
20 app.get('/', (req, res) => {
21   res.send('Hello, World!');
22 });
23
24 // ----- Register new user -----
25 app.post('/register', async (req, res) => {
26
27   const { firstname, lastname, email, password } = req.body;
28
29   if (!firstname || !lastname || !email || !password) {
30     console.log("itheeee");
31     return res.json({ error: 'Please fill all details' });
32   }
33
34   try {
35     const isEmailExist = await User.findOne({ email: email });
36
37     if (isEmailExist) {
38       return res.json({ success: "false", error: "Email already exist" });
39     }
40   }
41 });
```



```
backend > middleware > JS fetchUser.js > fetchUser
1 const jwt = require("jsonwebtoken");
2 const JWT_SECRET = "bhushanLinaImanasiniket";
3
4 const fetchUser = (req, res, next) => {
5   // Get the user from jwt token and add id to request object
6   const token = req.header('token');
7
8   if (!token) {
9     res.status(401).send({ error: "ePlease authenticate using a valid token" });
10  }
11
12  try {
13    const data = jwt.verify(token, JWT_SECRET); // check what it returns
14    // console.log('middleware ', data);
15    req.user = data; // 'req.user' is a object of user data, going to use everywhere where we are using middleware(fetchUser)
16    next();
17  } catch (error) {
18    res.status(401).send({ error: "bPlease authenticate using a valid token" });
19  }
20 }
21
22 module.exports = fetchUser;
```

PROBLEMS OUTPUT TERMINAL PORTS CODE REFERENCE LOG DEBUG CONSOLE

Connection Successful

Ln 9, Col 25 Spaces: 4 UTF-8 LF JavaScript

```
backend > models > JS dbmodel.js > ...
1 const mongoose = require("mongoose");
2
3 const userSchema = new mongoose.Schema({
4   firstName: {
5     type: String,
6     required: true
7   },
8   lastName: {
9     type: String,
10    required: true
11  },
12  email: {
13    type: String,
14    required: true,
15    unique: true
16  },
17  password: {
18    type: String,
19    required: true
20  }
21 });
22
23 // now we need to create a collection
24
25 const CapRegister = new mongoose.model("CapRegister", userSchema);
26
27
28 module.exports = CapRegister;
```

PROBLEMS OUTPUT TERMINAL PORTS CODE REFERENCE LOG DEBUG CONSOLE

Connection Successful

Ln 23, Col 1 Spaces:

3.4.3 Application Flow:

User Registration- The user will fill in their details like first name, last name, email, and password in the frontend registration form. Backend gets this data and validates it, guarantees, that all fields are filled, required information is provided and the email format is correct. After the validation is okayed, the backend creates a MongoDB database where the user data can be stored. In many cases, this is done by making a new document in a 'users' collection, and putting the information provided in there. Also, more steps of validation such as validating the input of the email to make sure its not already registered to avoid having two accounts with the same email address can be added.

User Authentication- Logged in users can log in by entering their email and password in a login form in the frontend. The backend gets the credentials for the login and runs a check to that provided in the data that is kept in the MongoDB database. This will verify if this email actually exists and, if it does, if the password provided belongs to the email in question Once the credentials are validated, the backend generates a JSON Web Token (JWT). For proof of the user's identity, this token is used, and hence it authenticates the later requests made by the users. The JWT in general carry data such as the user's ID and may be include other information that is important for authorization purposes.

Image Caption Generation- The frontend is equipped with input field and/or drag-and-drop function to allow users to send images via file input. The uploading of an image is the first step of the process, as the frontend sends it to the backend for further processing. This is accomplished through an HTTP inquiry, where an HTTP request is made with the image data included as part of the request payload. The image data is then gotten by the backend and is passed to an AI model carefully trained on image captioning. This model is based on the contents of the picture and it produces textual descriptions or captions that describe what is presented in the image. Next, where the AI model generates the captions, the backend sends this icon text back to the frontend. This is basically to reply the original quest made by the user through the frontend. The backend then visualizes the captions along with the images, giving the users the opportunity to see the text generated by the AI model.

3.5 Key Challenges:

During training the model, we have faced several challenges and listed down some of the challenges:

Training Stability- Training deep neural networks especially ones with recurrent architectures like LSTM can be hard because of gradient problem which includes clear gradient problem found in long short-term memory (LSTM). The decrease in gradients, called vanishing gradients, take place, when the gradients become very small during backpropagation, and thus, the process of training becomes harder. On the contrary, exploding gradients involve large gradients that are out of control, which is a source of instability in the training process. The key address of these problems was the Batch Normalization (BN). This method normalizes the activations within layers resulting in the stabilization of the training process. Furthermore, Gradient Clipping is introduced to cause gradient overflowing by means of limiting of a certain predefined threshold, which may be a pretty good remedy to exploding gradients. In this regard, proper initialization of LSTM weights helped to overcome the problem of vanishing gradients, thus ensuring a smooth training convergence.

Overfitting- Overfitting is an issue that arises when a model learns to memorize the training data which then prevents it from mastering to generalize on new samples and in the end the model will perform very poorly. Being the model including a large number of parameters, we had a great chance of the model to overfit and this is among the main problems. To overcome this, we adopted the dropout regularization, which deactivated some units at random during the training to reduce the model's dependence on specific features and thus to prevent the overfitting. Through implementing dropout to both encoder and decoder architectures of our model we stimulated learning that was robust as well improved overall generalization characteristics.

Data Loading and Preprocessing- Data handling with such datasets that include pictures and text of varying lengths was carried out via optimized data loading and preparation techniques. The sequence of packing and padding through wording like captions helped us to deal with variable-length inputs in our data loader in a convenient way. Niches contain introducing zeros to rows to make them uniform in length while in sets one combine vectors that are having different lengths. However, the data loader also helped to make the preprocessing end process easier, thus improving the efficiency of the processor during the training phase and alleviating wastage of computing resources.

Hyperparameter Tuning- Choosing right mask of hyperparameters is very important for the

purpose of getting best performance. Whether we consider architecture, batch size or a learning rate, they all have an enormous impact on the process of the training and the final model. The grid search technique was used to a systematic exploration of the hyperparameter space and find the best configuration. Gradient weight decay, step/schedule learning rate, and a scheduler programming which combines the stochastic and non-stochastic techniques was applied to fine -tune the convergence process. We fine-tuned our model through various hyperparameters adjustment methods that were based on the performance metrics. This allowed us to produce optimized model for reliability and accuracy. Through a mixture of cutting-edge technologies and painstaking experiments, we managed to create a deep learning model that can handle different datasets and perform better than any other task.

CHAPTER-4 TESTING

4.1 Testing Strategy:

1.Data Split: Our dataset is splitted into three parts: training, testing, and validation sets.

1. Training set is for training the model.
2. Validation set for early stopping and hypertunning.
3. Testing set is for evaluating the performance of the model.

2.Evaluation Metrics: For evaluating model's performance we have used BLEU score.

3.Tools Used:

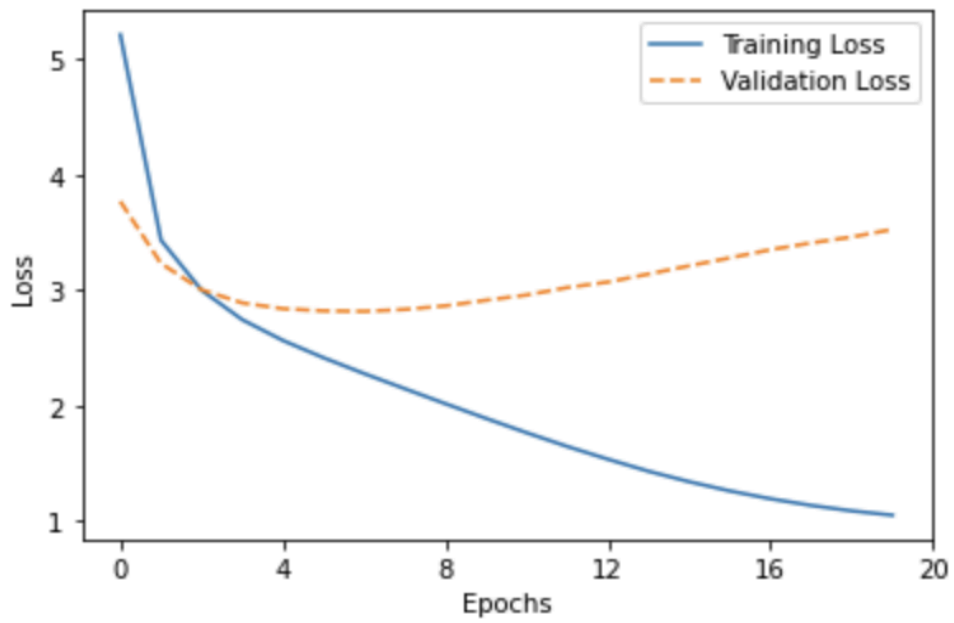
1. NLTK(Natural Language Toolkit) used for calculating BLEU.
2. Matplotlib for displaying images with their captions.

4.2 Test Cases and Outcomes:

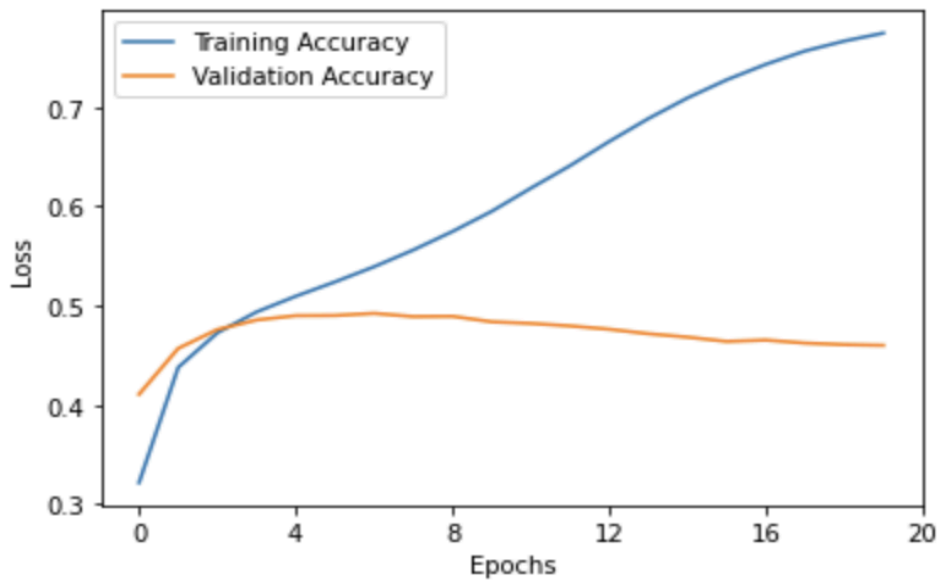
Loss Function Calculation:

```
750/750 [=====] - 1093s 1s/step - loss: 4.9639 - acc: 0.2538 - val_loss: 3.4216 - val_acc: 0.4755
Epoch 2/25
750/750 [=====] - 1117s 1s/step - loss: 3.0946 - acc: 0.4868 - val_loss: 2.9216 - val_acc: 0.5174
Epoch 3/25
750/750 [=====] - 1104s 1s/step - loss: 2.6874 - acc: 0.5291 - val_loss: 2.6898 - val_acc: 0.5363
Epoch 4/25
750/750 [=====] - 1105s 1s/step - loss: 2.4458 - acc: 0.5498 - val_loss: 2.5705 - val_acc: 0.5459
Epoch 5/25
750/750 [=====] - 1094s 1s/step - loss: 2.2801 - acc: 0.5653 - val_loss: 2.5073 - val_acc: 0.5515
Epoch 6/25
750/750 [=====] - 1101s 1s/step - loss: 2.1474 - acc: 0.5784 - val_loss: 2.4833 - val_acc: 0.5531
Epoch 7/25
750/750 [=====] - 1087s 1s/step - loss: 2.0301 - acc: 0.5901 - val_loss: 2.4691 - val_acc: 0.5551
Epoch 8/25
750/750 [=====] - 1091s 1s/step - loss: 1.9187 - acc: 0.6024 - val_loss: 2.4794 - val_acc: 0.5546
Epoch 9/25
750/750 [=====] - 1105s 1s/step - loss: 1.8099 - acc: 0.6163 - val_loss: 2.4939 - val_acc: 0.5522
Epoch 10/25
750/750 [=====] - 1190s 2s/step - loss: 1.6997 - acc: 0.6325 - val_loss: 2.5297 - val_acc: 0.5506
Epoch 11/25
750/750 [=====] - 1090s 1s/step - loss: 1.5909 - acc: 0.6506 - val_loss: 2.5693 - val_acc: 0.5503
Epoch 12/25
...
Epoch 24/25
750/750 [=====] - 1083s 1s/step - loss: 0.8172 - acc: 0.8232 - val_loss: 3.2498 - val_acc: 0.5228
Epoch 25/25
750/750 [=====] - 1082s 1s/step - loss: 0.8033 - acc: 0.8258 - val_loss: 3.2871 - val_acc: 0.5224
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Loss VS Epochs



Accuracy Vs Epochs



CHAPTER-5 RESULT AND EVALUATION

5.1 Results

1. Quantitative result

Overall Performance Metrics: For instance, BLEU score, METEOR score, CIDEr score, and ROUGE-L scores. These are all the parameters that can be used for the estimation

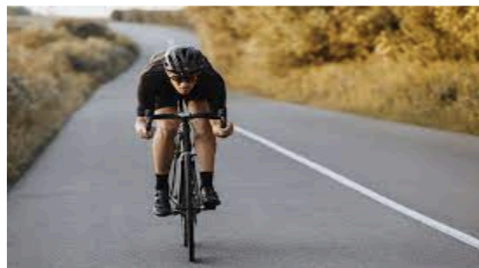
BLEU Score calculation:

```
# Calculate BLEU scores
bleu_1 = corpus_bleu(references, hypotheses, weights=(1.0, 0, 0, 0), smoothing_function=smoother.method4)
bleu_2 = corpus_bleu(references, hypotheses, weights=(0.5, 0.5, 0, 0), smoothing_function=smoother.method4)
bleu_3 = corpus_bleu(references, hypotheses, weights=(0.33, 0.33, 0.33, 0), smoothing_function=smoother.method4)
bleu_4 = corpus_bleu(references, hypotheses, weights=(0.25, 0.25, 0.25, 0.25), smoothing_function=smoother.method4)

# Print BLEU scores
print("BLEU-1: %f" % bleu_1)
print("BLEU-2: %f" % bleu_2)
print("BLEU-3: %f" % bleu_3)
print("BLEU-4: %f" % bleu_4)
```

```
BLEU-1: 0.311194
BLEU-2: 0.160884
BLEU-3: 0.071419
BLEU-4: 0.030901
```

[Go back](#)



a man wearing a helmet is riding a bike

[Convert text to speech](#)

English ▾

[Translate](#)

Go back



a person surfing a wave

Convert text to speech

English ▾

Translate

Go back



a little girl is posing for a play . . .

Convert text to speech

English ▾

Translate

Figure 5 - Home page of website

2. Qualitative results

Images with caption



"man in blue wetsuit is surfing on wave."



"young girl in pink shirt is swinging on swing."



black and white dog carries tennis ball in its mouth



A person is walking along a beach with a big dog

Figure-6 Qualitative results of images



A soccer player takes a soccer ball in the grass



A man is doing a trick on a snowboard



A surfer dives into the ocean



A black and white dog leaps to catch a Frisbee



A female tennis player in action on the court.



A group of young men playing a game of soccer

Figure-7 Qualitative results of images



A brown bear standing on top of a lush green field.



A person holding a cell phone in their hand.



A bunch of fruit that are sitting on a table.



A toothbrush holder sitting on top of a white sink.

Figure-8 Qualitative results of images

CHAPTER-6 CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

6.1.1 Key Findings

Image captioning is one of the best ways to connect spoken and visual communication. Image captioning is therefore beneficial for education and creative expression, as well as accessible for the blind and visually handicapped. The creation of picture descriptions is possible using this technique.

1. Improved Accuracy and Fluency- Captions are becoming more natural and grammatical also semantically appropriate image captioning models. They have become more accurate and natural. We are largely attributable to RNNs and CNNs that can sufficiently depict relationships between linguistic elements and visual attributes by deep learning architectures and models
2. Domain Adaptation and Multimodal Learning- Image captioning models thus are applicable to medical imaging, satellite imagery and art analysis since the vocabulary and concepts are domain-specific. This is mainly due to the recent methods of domain adaptation which enable models to learn about a given domain and multimodal learning techniques that combine visual and text information from different sources.
3. Real-Time Performance and Explainability- Real-time image captioning and interpretability of the model. The concept of explainability is important for understandability, debugging and trust to be effective in real time applications. Future breakthroughs in these fields will add to the practical value and weight of picture captioning.
4. Applications and Impact- Image captioning technology is already finding applications in various domains including-Accessibility,Education,Creative Expression.

This project related to the text-to-speech (TTS) has been the journey of learning and engaging in cutting edge science and technology aimed at clarifying aspects involving speech synthesis and advancing the state-of-the-art in TTS technology. By working on a step-by-step plan which interweaves theory, algorithmic designing, practical implementation, assessment, and information sharing, the project has achieved the prime goal, and the contribution it has brought to the field is worthwhile. The major success on our project has been to get a speech output that could be the most realistic among all text in speech systems having the proven ability to mimic human speech through providing high-quality accurate and expressive speech. With the help of information from phonetic analysis, modeling acoustic and machine learning techniques, the TTS system can show the power of modern speech synthesis software to produce real-time readout of high fidelity speech. The erroneous process of implementation and optimization towards a better synthesis quality has led to better and more accurate results with respect to computational efficiency and user experience.

Thus, it has contributed to the success and the future advancement in speech synthesis technology. In addition, the project has sketched the theoretical foundations and the practical issues involved in TTS and it has helped us to understand their important role in the TTS research and development. Through investigating a wide variety of the synthesizers species, from the oldest method of rule-based ones to currently cutting edge human able machines, the project has cast a light on the strengths, limitations and trade-offs of differing approaches. This detailed understanding is the foundation of all design decisions, optimization strategies, and evaluation framework approached in the project and thus, the entire work is based on properly defined solutions.

Beyond that, project brought to light the necessity of implementation of strict testing procedures and validations for the purpose of the evaluation of the capabilities and efficiency of the developed TTS technology. Metrics and also test subjects have offered the source to learn about synthesis quality, naturalness, and also user satisfaction among others, so, we could go to iterative improvements and optimizations. While the project has proven to be a reliable source of information, it has also proven to be knowledgeable and productive. The culture of transparency, reproducibility and knowledge dissemination has opened the platform to sharing the methodologies, findings and best practices with the community of TTS which in turn has created an ability for collaboration, innovation and expansion.

In reality, the making of a text-to-speech project is the next step in the process to further use

the technology in closing the gap caused by the language of communication being written down. The research team is pioneering the production of more realistic and personalized synthetic speech, higher efficiency, and increased versatility of applications. Therefore, a door is opened to areas such as accessibility, education, entertainment, communication and assistive technology for such speech. Looking back on all the things we have achieved and the values we have learned from this project is not only exciting but also encourages us to carry on our journey of finding new things, discovering new things and keeping up with novel technologies in the dynamical and ever-changing field of text-to-speech technology.

6.1.2 Limitations

The visual aids and image captioning can literally change the way people see the world around them. They are also able to extend the power of visual communication, foster creative expression, promote conversation and raise human–computer interaction. Some of the challenges that these technologies face include subjectivity, domain adaptation, explainability, real-time performance, data bias, privacy, sensory integration and user acceptance. So research and development are ongoing to resolve these issues and to enable them to be used widely and revolutionize. When these barriers are removed, society becomes sophisticated, colorful, and fully interacting with visual information, and hence, realizing its revolutionary potential. Visual assistance projects, which are projects that provide support and aids in computer science to visually impaired individuals using technology, face some limitations, a power that can influence performance, usability and accessibility of the projects. These limitations come from many distinctive sources, such as the technology provisions, the user characteristics, the environmental stresses, and ethical principles. Ensuring that you understand and rectify the shortcomings is significantly important for the improvement of your visual assistance systems that consider wider coverage and better end results.

1. **Technological Constraints:** Visual aid projects are mainly concerned with computer vision, image processing, and machine learning approaches that provide such assistance by capturing the visual world and gradually understanding it. The algorithms used which may be affected by variations in image quality, illumination, occlusions, and wide variations in appearance of the object, however, their effectiveness and accuracy can be limited. Dealing with complex compositions, cluttered scenes and uncertain visual hints carries the danger of misinterpretation and

may result in errors or inaccuracies in the assistance provided.

2. **Limited Object Recognition and Classification:** The accuracy of the vision-based helping systems will be one of the difficulties they might confront, in identifying and categorizing the objects in the real world. The de-novo object categorization comprising the variation between similar objects or subtle changes and identifying the related objects to a given context frequently over-delay the running algorithms. This limitation, however, may result in a loss of reliability or effectiveness of visual systems, especially pertaining to the provision of pertinent information and guidance.
3. **Text Recognition and Reading:** Last but not the least is one of the major challenges of these visual assistance projects, being the imprecision in the recognition of text from images or documents. Tensor detection systems have a hard job behind them, because they are expected to recognize complex fonts, gappy layouts, handwritten text or different quality of images, which also introduces mistakes or errors in text transcription. Besides that, compared with the reaction speed and accuracy of human or text recognition algorithms, their function might not be fast enough to fulfill some of the needs of users or even fail in complex environments.
4. **Navigation and Spatial Awareness:** Main objectives of sight based projects commonly are orienting users in public areas and providing them with safe and confident environment navigation. Nonetheless, the localization accuracy may be limited, the mapping can't cover areas by mistake, and route planning algorithms which can influence the reliability of navigation assistance you get. An uncertain location, a mismatch of real physical surroundings and problems with connecting to data in real time will help to decrease their capability to plan their trips.
5. **User Interface Design and Interaction:** In visual assistance undertakings the interface design and interaction mechanisms are the most important aspect for the usability from the user with visual disorders. But on the other hand some interface designs are complex and have issues of accessibility as every user may not have the skill level, knowledge and experience to fully handle it. On the other hand, users with limited vision might face usability obstacles from the complex navigation schemes, that interfaces are cluttered, small touch targets, and uncertain feedback. In the same way, those with limited dexterity could face the same obstacles.
6. **Privacy and Ethical Considerations:** Privacy and consent issues arise in visual assistance projects when modeling the environment or recording people which could lead to unwanted consequences and loss of confidentiality. Making sure that user

privacy, informed consent, and confidential handling of various data is top priority for the most part in order to build trust and maintain accountability for visual assistance systems. Furthermore, it is paramount to remove biases and guarantee fairness along the journey of algorithmic judgements in order to avoid discriminating or causing harm to marginalized user groups.

7. Integration with Existing Infrastructure and Ecosystems: There are obstacles with visual assistance projects that should be taken into consideration such as whether integration with existing infrastructure, assistive technologies and digital ecosystems can take place. Compatibility issues between devices, interoperability of devices or platforms, and fragmentation in device layers or software frameworks can create logistic and communication impediments. In addition, adopting the standards that lead to accessibility across numerous hardware devices, operating systems, and assistive technology platforms is a key to maximize the effectiveness of the visual assistance solutions.

6.1.3 Contribution

1. Enhanced Accessibility and Understanding: Imaging captioning solutions, if integrated into visual aid programmes, definitely will add to the accessibility and comprehension of the visually impaired of the visual content. The task of image captioning involves the creation of captions for images, objects, and scenes, which are then presented gradually over time, in a similar way to how real-time captions are displayed. This means that users can understand and engage with visual information and interpretations that would otherwise have remained inaccessible or difficult to understand. The ease of access increases the ability and opportunities for individuals to participate actively and independently in various sectors of life like education, entertainment, as well as socialization.
2. Improved Navigation and Spatial Awareness: Visual aids projects stand to benefit from image captioning as it supplies links of contextual information for movement and space guidance. Through presenting the layout of indoor settings, by listing landmarks and by mentioning some interesting features, image captioning contributes to the swiftness and the user confidence in the process of navigation in new environments. The higher degree of situated awareness helps to boost independence and orientation skills for the persons with visual impairments thereby making it

- possible for them to maneuver their surroundings with greater ease and independence.
3. **Enhanced Object Recognition and Classification:** The integration of image captioning for visually assisted projects can help in the recognition and classification of objects, especially in the cases of complicated sights or situations where the environment is crowded. Through giving all the detailed descriptions, such as labels and attributes, for the objects which have been detected above image captioning is an aid for people recognizing the objects correctly and also accordingly. By improving the perception of objects, orientation facilities get improved which enables independence in decision making, task completion and interaction with the surrounding climate which will, in turn, result in increased independence and self-reliance.
 4. **Facilitated Text Recognition and Reading:** Text commenting facilitates the recognition and reading for people with visual impairments, which are complementing the existing machine vision process for identification and reading OCR. Image captioning has the ability to give word-by-word description for images which contain text, it helps users to benefit from the textual content in diverse settings such as streets, offices and shops. These mean providing an environment in which individuals with visual impairments can access printed information on their own, further pursuing education and career aspirations, and become literate with inclusion.
 5. **Enriched Social Interaction and Communication:** Vision assistant technology is a recent development that not only enhances the social interaction of blind people in their visual surroundings, but can also provide a better understanding of shared visual content. Through providing textual explanations for images shared in social networks, instant messaging or online forums, captioning images opens more opportunities for users to interact with visuals and be involved in popular conversation. It is undoubtedly true that this emphasizes the feeling of belonging in the world of social networks and contributes to the diversity of social networks and vice versa which brings more accessibility and equity in digital platforms of communication.
 6. **Promotion of Technological Innovation and Collaboration:** The integration of image captioning projects with visual assistance technology enables the progress in computer vision, natural language processing and assistive technology, by creating opportunities for research, technological development, and cooperation among specialists. Built upon the advances in understanding of images, semantic analysis, and language generation, image captioning is a critical component of improving the interaction level and intelligence of more context sensitive visual assistance systems.

The interdisciplinary nature of collaborations between researchers drives advances of innovations in technology which are more accessible, usable, and effective, and a benefit to individuals with visual impairments as this leads to progress in our society.

7. Ethical Considerations and User Empowerment: The issue of image captioning technology being used for visual assistance purposes evoke ethical questions like what about privacy and before kind permission. Privacy and reliability of user information, as well as the integrity of the analysis and feedback processes are a must for the deployment of assistive technology which is trustworthy and accountable. Furthermore, the power handed back to the users, who can customize what they see, what they like, and how they interact, furthers their autonomy, dignity, and determination, which in turn correspond with the ethical approach to technology design and use.

6.2 FUTURE SCOPE

We will be seeing and defining the world in a new perspective with image captioning and visual assistance technologies. These technologies described above can help visually impaired people with real time visual information, education, and also give new opportunities for artistic self-expression.

1. Enhancing accessibility for people with visual impairments- For example image captioning and many other similar visual support tools could greatly facilitate the access to visual content for these people. These instruments give the blind the opportunity to feel motion, communicate and participate in society. For instance image captioning can be added to augmented reality devices to give real explanations of objects and scenes as well as visual aids apps that might improve visuals recognition among the poor-visioned persons and many others.
2. Enhancing human-computer interaction- Computer responsiveness and intuition can be enhanced by image captioning and visual assistance technologies. Such technologies are able to decode the semantic content of visual data and prompt or give the true answers and even customize the user interaction. For example they can be employed for image captioning to converse naturally, and development of smart home gadgets that are capable of answering visual orders.
3. Expanding the reach of visual communication- Visual assistive technologies and image captioning can increase the range of visual communication, making visual

content more understandable for a larger circle of people. These tools however, are able to break through barriers such as language and provide translations and description of an image. As such people with different levels of visual acuity can easily interpret visual content. For example image captioning facilitates translation of the visual instructions, warnings, and visual assistance

4. Transforming education and learning- Image captioning and assistive visual technologies can be engaging and interesting for students. These tools could also be applied for the explanation of a picture and diagram or any other visual element which could help the students with disabilities remember and understand the information. Image captioning is also important for creating interactive learning environments that involve learners working in teams. For instance it can be employed when the users receive comments from the learning software concerning the visual materials that they understand.
5. Enabling new forms of creative expression- There are possibilities of image captioning and visual assistance technologies that may generate new ways of creative expression through experimentation of ways one never might think possible. For example image captioning can serve as a basis for creative texts such as poems, scripts, etc based on visual inputs. Further visual assistance tools could be used to develop interactive and sensory art installations to question our perception of the world.

REFERENCES

1. T. Gupta and A. Schwing, "Meshed-Memory Transformer for Image Captioning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9360-9369.
2. P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6077-6086.
3. O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: A Neural Image Caption Generator," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3156-3164.
4. M. Tanti, A. Gatt, and K. P. Camilleri, "Framing Image Description as a Ranking Task: Data, Models, and Evaluation Metrics," Journal of Artificial Intelligence Research (JAIR), vol. 64, pp. 647-704, 2019.
5. L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9329-9340.
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proceedings of the 25th International Conference on Neural Information Processing Systems (NeurIPS), 2012, pp. 1097-1105.
7. Listen, Attend and Spell" (LAS) for Image Captioning proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) in 2017.
8. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 1
9. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013. 2
10. R. Socher, A. Karpathy, Q. V. Le, C. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. In ACL, 2014. 2 [30] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In NIPS, 2014. 1, 2, 3

11. R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. In arXiv:1411.5726, 2015. 5, 6
12. B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S.-C. Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8), 2010. 2
13. P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *ACL*, 2014. 5
14. W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. In arXiv:1409.2329, 2014
15. Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems 24*, pages 1143–1151, 2011.
16. [Papineni et al., 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, July 2002.
17. [Shawe-Taylor and Cristianini, 2004] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
18. [Simonyan and Zisserman, 2014] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
19. [Vedaldi and Fulkerson, 2008] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
20. [Vinyals et al., 2014] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014.
21. [Yang et al., 2011] Yezhou Yang, Ching Teo, Hal Daume III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 444–454, 2011.
22. [Young et al., 2014] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014
23. Xu, J., Mei, T., Yao, T., & Rui, Y. (2016). MSR-VTT: A large video description

- dataset for bridging video and language. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
24. Chen, X., Lawrence Zitnick, C., & Dolan, B. (2015). Microsoft COCO captioning and visual question answering dataset. arXiv preprint arXiv:1504.00325.
 25. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NeurIPS), 2015.

Chirag_Report

ORIGINALITY REPORT

7 %	6 %	2 %	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	github.com Internet Source	3 %
2	www.arxiv-vanity.com Internet Source	1 %
3	cse.anits.edu.in Internet Source	<1 %
4	hockenmaier.cs.illinois.edu Internet Source	<1 %
5	dokumen.pub Internet Source	<1 %
6	ebin.pub Internet Source	<1 %
7	escholarship.org Internet Source	<1 %
8	docplayer.net Internet Source	<1 %
9	Hind Khalid. "Efficient Image Annotation and Caption System Using Deep Convolutional	<1 %

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: _____ Department: _____ Enrolment No _____

Contact No. _____ E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none">• All Preliminary Pages• Bibliography/Images/Quotes• 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

**Checked by
Name & Signature**

Librarian

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com