IJCSPUB.ORG ISSN : 2250-1770
INTERNATIONAL JOURNAL OF CURRENT SCIENCE (IJCSPUB)
IJCS PUBLICATION (IJCSPUB.ORG) An International Open Access, Peer-reviewed, Refereed Journal

ENHANCING VISUAL UNDERSTANDING THROUGH VERBAL IMAGE CAPTION GENERATOR USING ADVANCED DNN TECHNIQUES

[1]Tiruveedhula Amrutha Lakshmi, [2]Rayala Meghana, [3]Tummalapudi Sowmya, [4]Koppula Bhavya Sree, [5]Buradagunta Avinash

[1-4]Student, [5]Assistant Professor
[1-5]Information Technology,
[1-5]Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andhra Pradesh, India

Abstract: The use of Deep Learning techniques by image caption generators to automatically provide meaningful descriptions for images has advanced significantly in recent years. This project introduces a Verbal Image Caption Generator, employing cutting-edge DNN techniques to create informative and contextually relevant image descriptions. We utilize state-of-the-art CNNs and RNNs for feature extraction and natural language generation, ensuring coherent and accurate captions. Additionally, a text-to-speech module is integrated, enabling the conversion of captions into audible voice. This technology improves accessibility for visually impaired individuals and enriches multimedia platforms. By combining image captioning and voice synthesis, this project promotes a more inclusive and comprehensive understanding of the digital visual world.

Index Terms – Deep Learning, Deep Neural Networks, Visual Understanding, Image Caption Generator, Verbal Description, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM)

I. INTRODUCTION

This Project addresses the need for comprehensive image descriptions in the digital landscape. Leveraging advanced Deep Neural Network (DNN) methods, we generate precise, contextually relevant image captions from datasets. This improves accessibility and comprehension. By utilizing CNNs for image feature extraction and RNNs for language generation, we ensure coherent and accurate captions. We also incorporate Google Text-to-Speech (gTTS) technology for audio conversion, benefiting the visually impaired and enhancing multimedia experiences. This project aims to bridge the gap between visual and auditory understanding, making the digital visual world more inclusive and accessible.

II. LITERATURE SURVEY

Generating image captions is an important aspect of Computer Vision and Natural language processing. Image caption generators can find applications in Image segmentation as used by Facebook and Google Photos, and even more so, its use can be extended to video frames. They will easily automate the job of a person who has to interpret images. Not to mention it has immense scope in helping visually impaired people. [1]

For evaluation of the performance of the described model we have used BLEU scores. Through the scores, one can apart the generated captions as good captions and bad captions. Main applications of this model include usage in virtual assistants, for image indexing, for social media, for visually impaired people, recommendations in editing applications and much more. [2]

The application of image caption is extensive and significant, for example, the realization of human-computer interaction. This paper summarizes the related methods and focuses on the attention mechanism, which plays an important role in computer vision and is recently widely used in image caption generation tasks. [3]

Natural languages are used by humans to describe scenes because they are brief and compact. Machine vision systems, on the other hand, characterize the scene by capturing an image that is a two-dimensional array. The concept is to combine the image and captions into one place and then learn a mapping from the visual to the phrases. [4]

To overcome the limitations occurring by using open datasets, this paper proposes a domain-specific image caption generator, which generates a caption based on attention mechanism with object and attribute information, and reconstruct a generate caption using a semantic ontology to provide natural language description for given specific-domain. To show the effectiveness of the proposed model, we evaluate the image caption generator with a dataset, MSCOCO, quantitatively and qualitatively. [5]

III. EXISTING SYSTEM

The current system employs computer vision and deep learning to process input images. It extracts visual features, identifies objects and context, and uses NLP for caption generation. The NLP model is trained on image-caption datasets for accuracy. User interaction occurs via GUI or API with image input and caption output. Users receive captions describing the image's content on the interface.

IV. PROPOSED SYSTEM

The proposed method generates captions through deep learning by detecting and recognizing objects, employing voice synthesis. The process includes object detection, feature extraction, and a CNN for scene classification. This approach merges these architectures into a CNN-RNN model, utilizing the Exception pre-trained model. An LSTM incorporates CNN information to generate image descriptions. The generated caption is played using a Text-to-Speech Engine (gTTS). This combined system offers an integrated approach for image caption generation and vocalization.

V. OBJECTIVE

The goal of this project is to develop a cutting-edge image caption generator using advanced DNN techniques. It will generate accurate and contextually relevant image descriptions. Additionally, we will integrate text-to-speech technology to convert these captions into audible voice, enhancing accessibility. Our goal is to provide a user-friendly and inclusive solution for a richer understanding of visual content while contributing to the field of image captioning technology.

VI. METHODOLOGY

MODULE 1: Data Collection

MODULE 2: Pre-processing and Training

MODULE 3: Image Caption Generation

MODULE 4: Voice Synthesis Integration

VII. DESIGN

7.1 Architecture Diagram:

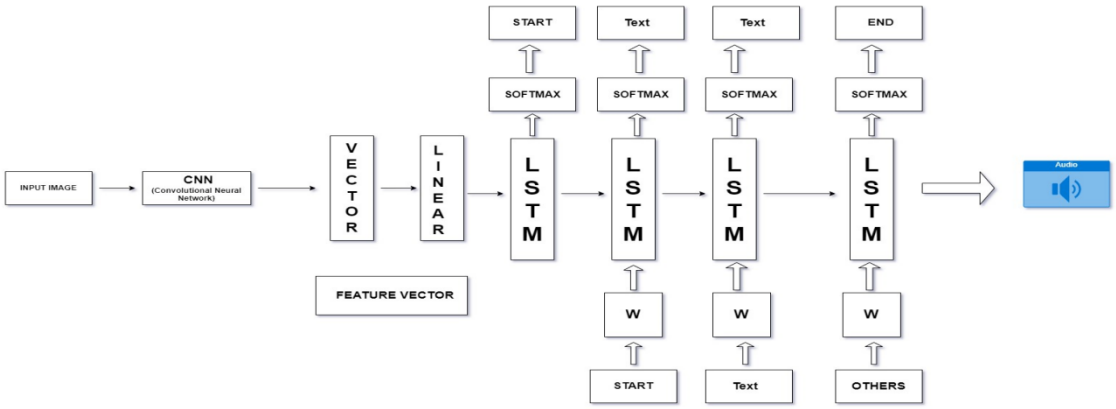

Figure 1: ARCHITECTURE OF SYSTEM

7.2 Data-Flow Diagram:

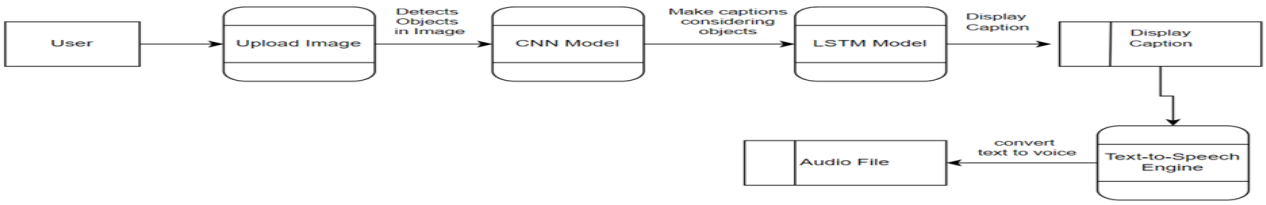

Figure 2: DATA FLOW DIAGRAM

VIII. ALGORITHMS USED

8.1 Convolutional Neural Network(CNN):

Convolutional Neural Networks are specialized deep neural networks which can process the data that has input shape like a 2D matrix. Images are easily represented as a 2D matrix and CNN is very useful in working with images. CNN is basically used for image classifications and identifying if an image is a bird, a plane or Superman, etc. It scans images from left to right and top to bottom to pull out important features from the image and combines the feature to classify images. It can handle the images that have been translated, rotated, scaled and changes in perspective.

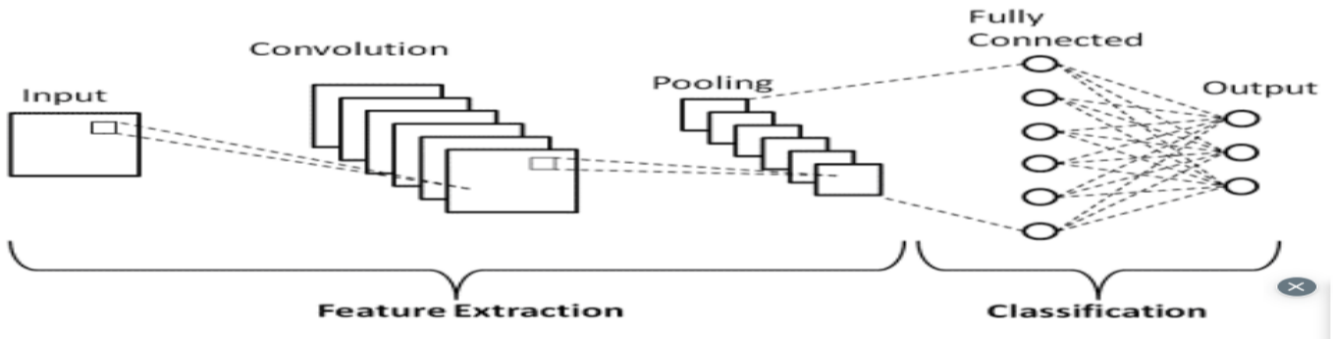

Figure 3: CNN ARCHITECTURE

8.2 Long Short-Term Memory(LSTM):

LSTM stands for **Long Short-Term Memory**, they are a type of RNN (**Recurrent Neural Network)** which is well suited for sequence prediction problems. Based on the previous text, we can predict

what the next word will be. It has proven itself effective from the traditional RNN by overcoming the limitations of RNN which had short term memory. LSTM can carry out relevant information throughout the processing of inputs and with a forget gate, it discards non-relevant information. LSTM uses the information from CNN to help generate a description of the image.

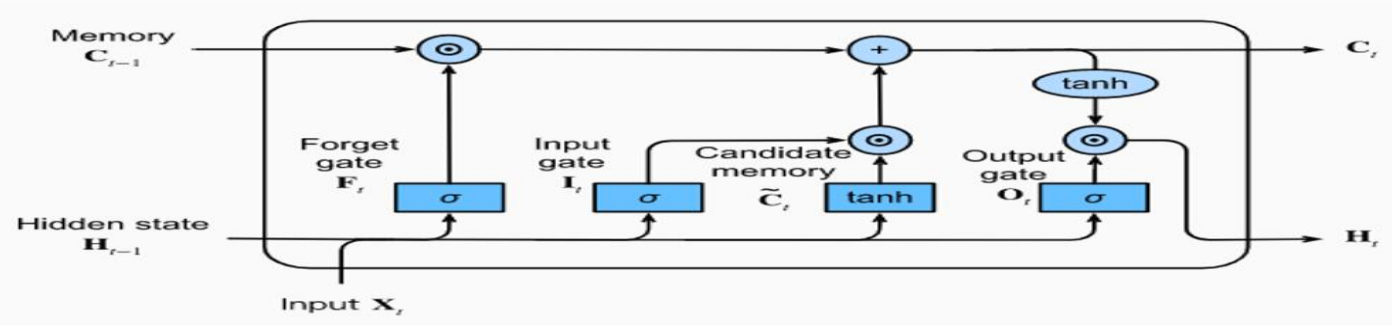

Figure 4: LSTM ARCHITECTURE

IX. DATASET

- For the Image Caption Generator, we used the Flickr_8K dataset. There are also other big datasets like Flickr_30K and MSCOCO dataset but it can take weeks just to train the network so we will be using a small Flickr8K dataset. The advantage of a huge dataset is that we can build better models.
- The Flickr_8K text folder contains file Flickr8k.token which is the main file of our dataset that contains image name and their respective captions separated by newline ("\n").

X. IMPLEMENTATION

10.1 Import all the necessary Packages:

The Packages imported in this project are: tensorflow, keras, pillow, numPy, tqdm, pandas.

10.2 Getting and Performing Data Cleaning:

The main text file which contains all image captions is **Flickr8k.token** in **Flickr_8k_text** folder. Each image has 5 captions and we can see that # (0 to 5) number is assigned for each caption.

Have a look at the file –

Flickr8k.token - Notepad

File Edit Format View Help

```
1000268201_693b08cb0e.jpg#0     A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg#1     A girl going into a wooden building .
1000268201_693b08cb0e.jpg#2     A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg#3     A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e.jpg#4     A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70.jpg#0     A black dog and a spotted dog are fighting
1001773457_577c3a7d70.jpg#1     A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70.jpg#2     A black dog and a white dog with brown spots are staring at each other in the street .
1001773457_577c3a7d70.jpg#3     Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70.jpg#4     Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8.jpg#0     A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
1002674143_1b742ab4b8.jpg#1     A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg#2     A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
1002674143_1b742ab4b8.jpg#3     There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg#4     Young girl with pigtails painting outside in the grass .
1003163366_44323f5815.jpg#0     A man lays on a bench while his dog sits by him .
1003163366_44323f5815.jpg#1     A man lays on the bench to which a white dog is also tied .
1003163366_44323f5815.jpg#2     a man sleeping on a bench outside with a white and black dog sitting next to him .
1003163366_44323f5815.jpg#3     A shirtless man lies on a park bench with his dog .
1003163366_44323f5815.jpg#4     man laying on bench holding leash of dog sitting on ground
1007129816_e794419615.jpg#0     A man in an orange hat starring at something .
1007129816_e794419615.jpg#1     A man wears an orange hat and glasses .
1007129816_e794419615.jpg#2     A man with gauges and glasses is wearing a Blitz hat .
1007129816_e794419615.jpg#3     A man with glasses is wearing a beer can crocheted hat .
1007129816_e794419615.jpg#4     The man with pierced ears is wearing glasses and an orange hat .
1007320043_627395c3d8.jpg#0     A child playing on a rope net .
1007320043_627395c3d8.jpg#1     A little girl climbing on red roping .
1007320043_627395c3d8.jpg#2     A little girl in pink climbs a rope bridge at the park .
1007320043_627395c3d8.jpg#3     A small child grips onto the red ropes at the playground .
1007320043_627395c3d8.jpg#4     The small child climbs on a red ropes on a playground .
1009434119_febe49276a.jpg#0     A black and white dog is running in a grassy garden surrounded by a white fence .
1009434119_febe49276a.jpg#1     A black and white dog is running through the grass .
1009434119_febe49276a.jpg#2     A Boston terrier is running in the grass .
1009434119_febe49276a.jpg#3     A Boston Terrier is running on lush green grass in front of a white fence .
1009434119_febe49276a.jpg#4     A dog runs on the green grass near a wooden fence .
1012212859_01547e3f17.jpg#0     A dog shakes its head near the shore , a red ball next to it .
```

Figure 5: CAPTIONS FOR IMAGES

10.2.1 Data Cleaning:

In this project, we remove punctuations, converting all text to lowercase and removing words that contain numbers. So, a caption like “A man riding on a three-wheeled wheelchair” will be transformed into “man riding on three wheeled wheelchair.

```
#Data cleaning- lower casing, removing puntuations and words containing numbers
def cleaning_text(captions):
    table = str.maketrans('','',string.punctuation)
    for img,caps in captions.items():
        for i,img_caption in enumerate(caps):

            img_caption.replace("-"," ")
            desc = img_caption.split()

            #converts to lowercase
            desc = [word.lower() for word in desc]
            #remove punctuation from each token
            desc = [word.translate(table) for word in desc]
            #remove hanging 's and a
            desc = [word for word in desc if(len(word)>1)]
            #remove tokens with numbers in them
            desc = [word for word in desc if(word.isalpha())]
            #convert back to string

            img_caption = ' '.join(desc)
            captions[img][i]= img_caption
    return captions
```

Figure 6: DATA CLEANING

10.3 Extracting the feature vector from all images:

This technique also known as transfer learning, leverages pre-trained models to extract features for specific tasks. The Xception model, originally designed for ImageNet, is imported from Keras for feature extraction. The model’s last classification layer is removed, yielding a 2048 feature vector. Features are extracted for all images, associated with their names and saved as a pickle file. This approach streamlines the process by reusing pre-trained model knowledge for feature extraction.

model= Xception(include_top=False, pooling=’avg’)

10.4 Loading dataset for Training the model:

In the **'Flickr_8k_test'** folder, **'Flickr_8k.trainImages.txt'** holds a list of 6000 image names for training. **'load_photos (filename)'** loads this list and returns the image names. **'load_clean_descriptions (filename, photos)'** generates a dictionary of image captions, adding start and end tokens for LSTM. **'load_features (photos)'** provides a dictionary linking image names to their feature vectors extracted from Xception. These functions facilitate data preparation for training and caption generation using image features.

10.5 Tokenizing the vocabulary:

To enable computer processing, English words are represented with unique numeric indices. The Keras library's tokenizer function maps words to tokens, saving them in a "tokenizer.p" pickle file. The vocabulary comprises 7577 words. Determining a maximum description length of 32 is crucial for model parameter decisions. These steps prepare the data by converting words to numerical tokens and setting a maximum sequence length for the model.

10.6 Create Data Generator:

For supervised learning, input and output pairs are needed for model training. This includes 2048-length feature vectors for 6000 images. Due to memory constraints, a generator method is used to yield data in batches. The generator provides input sequences [x1, x2] and corresponding output sequences (y). In the example, x1 is the image feature vector, x2 is the input text sequence, and y is the target output text. This setup enables the model to predict words in a sequential manner.

x1(feature vector)	x2(Text sequence)	y(word to predict)
feature	start,	two
feature	start, two	dogs
feature	start, two, dogs	drink
feature	start, two, dogs, drink	water
feature	start, two, dogs, drink, water	end

Figure 7: DATA GENERATION

10.7 Defining the CNN-RNN model:

The model structure, implemented with Keras Functional API, comprises three main components. The Feature Extractor reduces the 2048-dimensional image features to 256 nodes using a dense layer. The Sequence Processor handles textual input via an embedding layer and LSTM layer. The Decoder combines the output from the previous two layers, processing it through a dense layer. The final layer's node count matches the vocabulary size for making predictions.

```
# define the captioning model
def define_model(vocab_size, max_length):

    # features from the CNN model squeezed from 2048 to 256 nodes
    inputs1 = Input(shape=(2048,))
    fe1 = Dropout(0.5)(inputs1)
    fe2 = Dense(256, activation='relu')(fe1)

    # LSTM sequence model
    inputs2 = Input(shape=(max_length,))
    se1 = Embedding(vocab_size, 256, mask_zero=True)(inputs2)
    se2 = Dropout(0.5)(se1)
    se3 = LSTM(256)(se2)

    # Merging both models
    decoder1 = add([fe2, se3])
    decoder2 = Dense(256, activation='relu')(decoder1)
    outputs = Dense(vocab_size, activation='softmax')(decoder2)

    # tie it together [image, seq] [word]
    model = Model(inputs=[inputs1, inputs2], outputs=outputs)
    model.compile(loss='categorical_crossentropy', optimizer='adam')

    # summarize model
    print(model.summary())
    plot_model(model, to_file='model.png', show_shapes=True)

    return model
```

Figure 8: DEFINING THE CNN-RNN MODEL

Visual representation of the final model is given below:

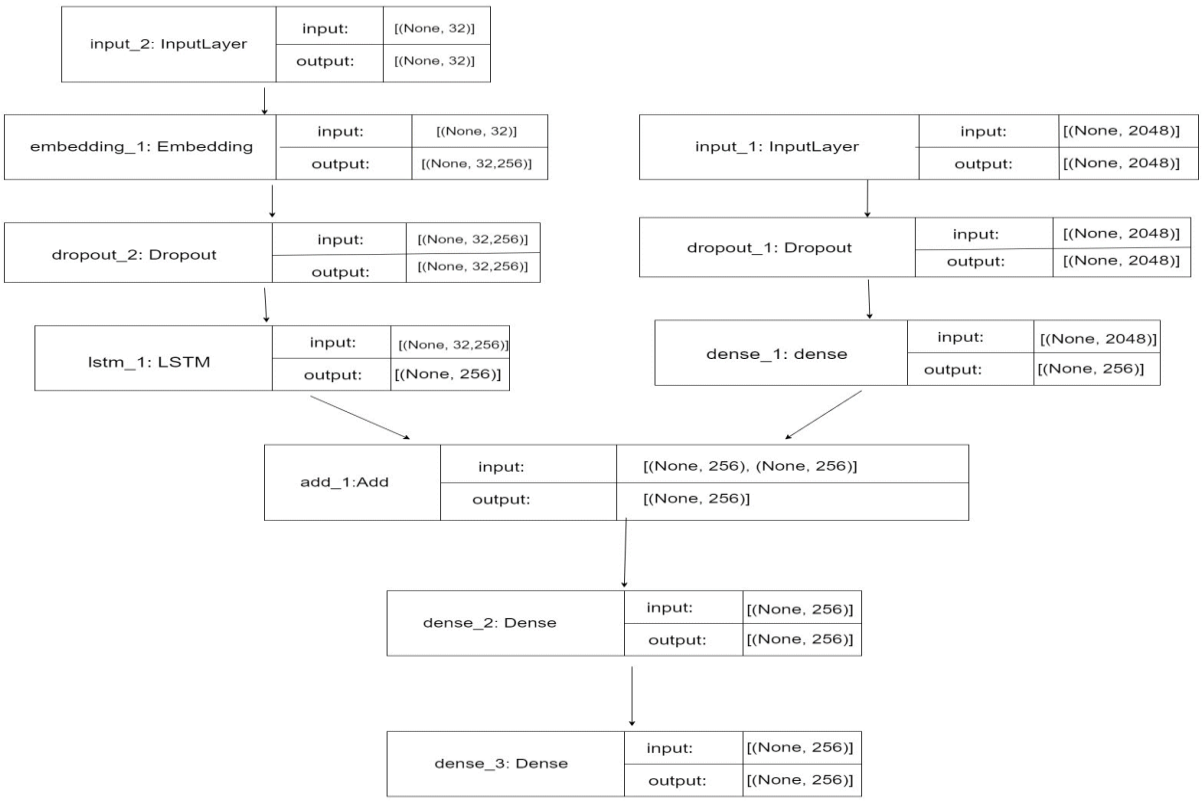

Figure 9: REPRESENTATION OF CNN-RNN MODEL

10.8 Training the model:

Training uses 6000 images with batch-generated input and output sequences, utilizing 'model.fit_generator ()'. The trained model is saved in the "models" folder, and the time required depends on system capabilities.

```
# train our model
print('Dataset: ', len(train_imgs))
print('Descriptions: train=', len(train_descriptions))
print('Photos: train=', len(train_features))
print('Vocabulary Size:', vocab_size)
print('Description Length: ', max_length)

model = define_model(vocab_size, max_length)
epochs = 10
steps = len(train_descriptions)
```

Figure 10: TRAINING THE MODEL

10.9 Testing the model:

After training, a separate file, "testing_caption_generator.py," is created to load the model and make predictions. The predictions, represented by index values, are converted back to words using the "tokenizer.p" pickle file for post-processing.

```
# Specify the image path here
img_path ="C:\\Users\\Pavan\\Downloads\\python-project-image-caption-generator\\Flickr8k_Dataset\\Flicker8k_Dataset\\49553964_cee
# Specify the maximum caption length
max_length = 32

# Extract features from the image
photo = extract_features(img_path, xception_model)

if photo is not None:
    # Generate a description for the image
    description = generate_desc(model, tokenizer, photo, max_length)
    print("\nGenerated Caption:")
    print(description)
```

Figure 11: TESTING THE MODEL

10.10 Generating Voice from Caption:

Voice generation from captions is accomplished using a Text-to-Speech (TTS) engine like gTTS (Google Text-to-Speech). The engine converts the text-based captions into audible speech, making it accessible for users.

```
In [14]: from gtts import gTTS
         from playsound import playsound
         mytext = description
         language = 'en'
         filename1 = "voice0003.mp3"
         myobj = gTTS(text=mytext, lang=language, slow=False)
         myobj.save(filename1)
         playsound(filename1,block=False)
```

Figure 12: GENERATING VOICE FROM CAPTION

XI. RESULT ANALYSIS

The image caption generator consistently produced accurate, contextually relevant captions, enhancing visual understanding. The integrated text-to-speech (TTS) system provided clear, natural voice synthesis for accessibility, ensuring audibility and inclusivity for a diverse user base. User testing validated its user-friendliness, and ethical compliance was maintained, making it a valuable contribution to accessibility and image captioning technology. This analysis underscores the project's success in making visual content more accessible and user-friendly.

img_path=”D:\\Projects\\ImageCaptionGenerator\\Flickr8k_Dataset\\Flicker8k_Dataset\\101669240_b2d3e7f17b.jpg”
img_path=”D:\\Projects\\ImageCaptionGenerator\\Flickr8k_Dataset\\Flicker8k_Dataset\\101669226_a153423b2.jpg”

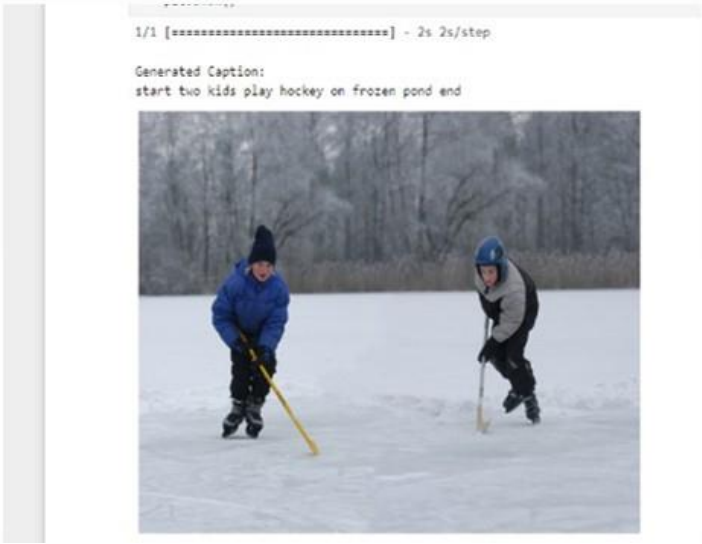

Figure 13: CAPTION GENERATION FOR INPUT IMAGE1 FOR INPUT

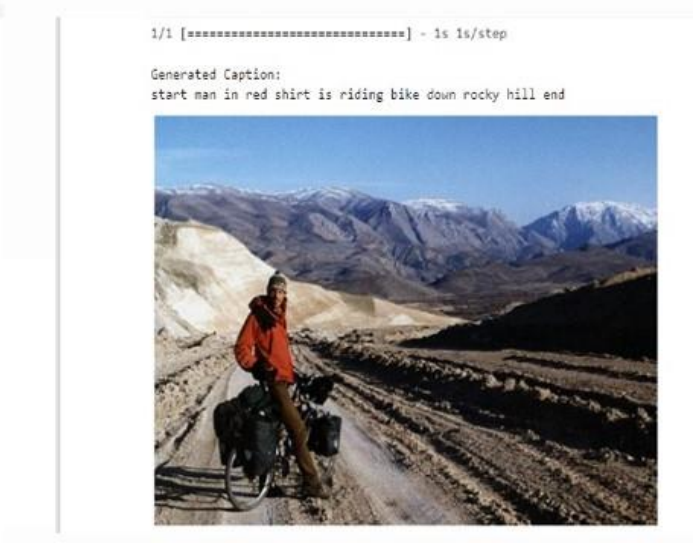

Figure 14: CAPTION GENERATION IMAGE2

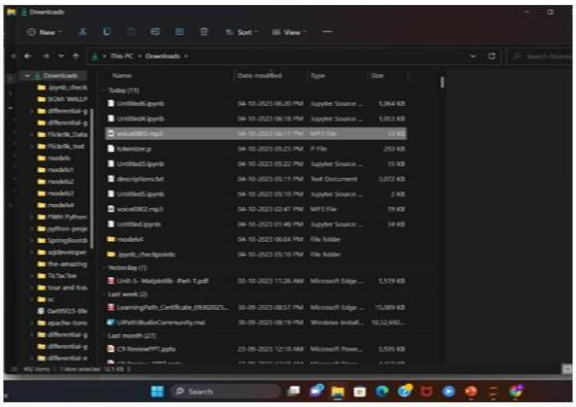

Figure 15: AUDIO FILE OF GENERATED CAPTION

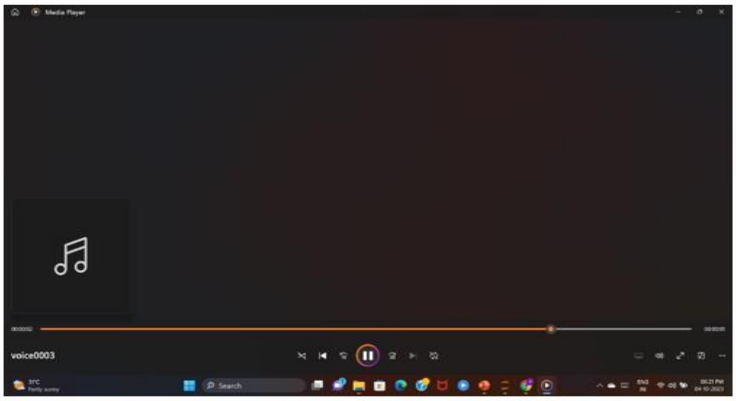

Figure 16: PLAYING AUDIO

XII. CONCLUSION

In conclusion, this project successfully enhances visual understanding and accessibility through an accurate image caption generator and clear voice synthesis. User testing confirmed inclusivity and user-friendliness. Future work involves expanding datasets, real-time captioning, multilingual support, advanced accessibility features, and model improvements, ensuring the project remains at the forefront of image captioning and accessibility technology. This project bridges the gap between visual content and understanding, making multimedia accessible to a global audience while contributing to accessible technology advancement.

XIII. FUTURE WORK

In this project's future work, we plan to expand the image dataset to enhance caption accuracy and diversity. Real-time image captioning will enable users to upload or capture images on the spot, promoting dynamic caption generation. Multilingual support is on the horizon, making captions accessible in various languages. Advanced accessibility features will be explored to meet diverse user needs, and continuous model improvement will ensure our technology remains up-to-date with the latest advancements, bolstering caption quality and accuracy. These initiatives aim to keep our project at the forefront of accessible image captioning technology.

XIV. REFERENCES

[1] Panicker, M. J., Upadhayay, V., Sethi, G., & Mathur, V. (2021). Image caption generator. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 10(3).

[2] Raypurkar, M., Supe, A., Bhumkar, P., Borse, P., & Sayyad, S. (2021). Deep learning based image caption generator. International Research Journal of Engineering and Technology (IRJET), 8(03).

[3] Wang, H., Zhang, Y., & Yu, X. (2020). An overview of image caption generation methods. Computational intelligence and neuroscience, 2020.

[4] Krishnakumar, B., Kousalya, K., Gokul, S., Karthikeyan, R., & Kaviyarasu, D. (2020). Image caption generator using deep learning. International Journal of Advanced Science and Technology, 29(3s), 975-980.

[5] Han, S. H., & Choi, H. J. (2020, February). Domain-specific image caption generator with semantic ontology. In 2020 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 526-530). IEEE.

[6] Kamal, A. H., Jishan, M. A., & Mansoor, N. (2020, November). Textmage: The automated bangla caption generator based on deep learning. In 2020 International Conference on Decision Aid Sciences and Application (DASA) (pp. 822-826). IEEE.

[7] Shinde, V. D., Dave, M. P., Singh, A. M., & Dubey, A. C. (2020). Image caption generator using big data and machine learning. International Research Journal of Engineering and Technology (IRJET), 7(04).

[8] Kumar, N. K., Vigneswari, D., Mohan, A., Laxman, K., & Yuvaraj, J. (2019, March). Detection and recognition of objects in image caption generator system: A deep learning approach. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS) (pp. 107-109). IEEE.

[9] Sharma, G., Kalena, P., Malde, N., Nair, A., & Parkar, S. (2019, April). Visual image caption generator using deep learning. In 2nd international conference on advances in Science & Technology (ICAST).

[10] Kesavan, V., Muley, V., & Kolhekar, M. (2019, October). Deep learning based automatic image caption generation. In 2019 Global Conference for Advancement in Technology (GCAT) (pp. 1-6). IEEE.