INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

# VISUAL DISCOVERY: NAVIGATING ENVIRONMENTS

**Mr. A. Parvej[1], Tanduri Anvith Reddy[2], Garishe Saikumar[3], Purma Dileep Reddy[4], Vasam Nikhil Kumar[5]**

[1]Associate. Professor, CSE Dept, ACE Engineering College, Hyderabad, India.

[2,3,4,5]Student, CSE Dept, ACE Engineering College, Hyderabad, India.

## ABSTRACT

The advancement of technologies facilitates the creation of visual contents on various platforms. Visual contents convey information easily through graphical representations. Understanding visual content is natural for sighted people, but not for the visually impaired. Web accessibility for visually impaired people is limited, with barriers and challenges in accessing online content. A study found that most visually impaired participants had difficulty visiting new websites and completing online tasks. Many areas, such as mobile applications and gaming, have exclusive visual oriented designs. There is a lack of detailed guidelines for enhancing visual web content accessibility. The promotion of augmented reality technology does not equally cater to those unable to perceive visual effects

## 1. INTRODUCTION

The main goal of the project is to examine and implement modern techniques which will help visually impaired individuals easily perceive visual content. It aims to address the challenges that the visually impaired face in accessing visual information. For sighted people, this kind of digital content is easily accessible. However, use of visual media and graphical interfaces represents an enormous challenge for individuals who are blind or visually impaired. In order to find ways to enhance the accessibility of visual content to visually impaired people, we have to first understand the problems they are facing today. Visual content is effectively in a foreign language to the blind. However, with the ever-increasing amount of digital information and services available, the ability for visually impaired people to access what is considered to be the predominant method of conveying information is becoming more and more vital.

Without the ability to effectively understand and use the huge amount of information in the global digital world, visually impaired individuals run the risk of further exclusion from many of the benefits that society and technology in the 21st century are offering. The traditional methods for conveying visual digital content used in the majority of webpages and software are via textual descriptions. For example, when a web page which includes some images is loaded, the alternative text, which is essentially a textual description of the visual content, is loaded. However, the usefulness of this kind of method is limited on several different levels. It means that if one was to try and provide a "full" verbal description of the content, it could be very time-consuming for the blind user and it could be quite invasive for anybody who is listening to it. Plus, for someone who has never seen before, describing simple things, like say a person's face, may require a vast amount of descriptive text.

## 2. OBJECTIVES

The objective of the project is to enhance the accessibility of visual content for the visually impaired. It addresses the challenges that the visually impaired face in accessing visual information. The traditional methods for conveying visual content, such as alt text descriptions, have limitations. The project explores the utilization of advanced computer vision techniques, including object detection, scene recognition, and image segmentation, to improve accessibility. It also focuses on recognizing critical components of visual content for better understanding by visually impaired individuals. It is important to identify the main object that the author tries to present in the photo. For the visually impaired, it is difficult to locate the main object in a photo, as they could not analyze the photo from global to local as what the sighted people usually do. By adapting the state-of-the-art object detection and image recognition techniques, the project aims to develop a robust method to locate the main object in a photo and generate related descriptions automatically. The visually impaired could then be provided with more direct and accurate information conveyed from the photo. Lastly, many infographics, diagrams and charts in the document are not accessible for the visually impaired because the current screen readers could only process text but not the visual content. The project aims to investigate the potential of an AI-powered system that could convert the visual graphics into a readable format. By using the graph-cut image segmentation algorithm, the project proposes to automatically separate different components in the chart, such as the bars, text labels, and the legend. By analyzing their spatial distribution and content, the system is able to generate a textual description for each component and then assemble them into a full description for the chart. With the development of such a system, the visually impaired could have barrier-free access to infographics and charts.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

## 3. PROBLEM STATEMENT

The visually impaired suffer from challenges when it comes to accessing the same information as those with full sight. In the modern digital age, these problems manifest most acutely in digital contexts, such as on the internet. Only around 1 in 4 websites are optimized for the visually impaired. Graphically rich and visually stunning sites are often the least accessible. This is a source of frustration for many, leading to a growing call for better accessibility. Screen readers, the most common tools used by the blind and visually impaired community, are essential for interpreting digital information. However, they have limitations. Alt text, where a screen reader reads out a piece of text assigned to an image, are a staple method for web accessibility. But it's a manual process and is bottlenecked by human error and input. Similarly, text descriptions of visual content, used by screen readers to relay information, often lack context. The visually impaired community also has to contend with a digital divide, the gap between those with access to computers and the internet, and those without. Among the disabled and visually impaired community, the divide is striking. Only around 50% of people with a disability use the internet, compared to 80% of the wider population. These factors contribute to a segregated online experience for the visually impaired. The need for a solution, or a number of varied solutions, is pronounced. Such a conclusion could have a substantial, positive impact on the daily experiences of the visually impaired, with far-reaching implications for many on a global scale. Work in academia and industry, such as this project, serve to break down barriers and engage further with technological solutions to improve the lives of many. Such efforts can only serve to close the digital divide, empower the visually impaired and, from a more cynical standpoint, open up a new, lucrative market.

## 4. LITERATURE SURVEY

"Virtual Touch: Computer Vision Augmented Touch-Free Scene Exploration for the Blind or Visually Impaired" by Xixuan Julie Liu and Yi Fang was published in 2021. هما

The article titled "Virtual Touch: Computer Vision Augmented Touch-Free Scene Exploration for the Blind or Visually Impaired" by Xixuan Julie Liu and Yi Fang was published in 2021.

In This Paper, the development of a system that uses computer vision techniques to provide a touch-free way for blind or visually impaired individuals to explore their surroundings. The authors begin by introducing the challenges faced by visually impaired individuals in accessing and comprehending visual content, such as images and videos. They highlight the limitations of traditional approaches like alternative text descriptions and propose the use of computer vision technology to provide a more comprehensive solution.

The authors then describe the system they developed, which uses a camera and computer vision algorithms to analyse the user's surroundings and provide audio feedback about the objects and surfaces detected. The system can recognize and provide information about common objects like doors, chairs, and tables, as well as the layout of the room and the distance between objects. To evaluate the system's effectiveness, the authors conducted a user study with six visually impaired participants. The participants were asked to navigate a room with the help of the system and provide feedback on its accuracy and usefulness. The results showed that the system was able to provide accurate and useful information about the user's surroundings, and that the participants found it easy to use and helpful.

In their conclusion, the authors emphasize the potential benefits of touch-free scene exploration for visually impaired individuals, including increased independence and improved quality of life. They also highlight the need for further research and development in this area to improve the accuracy and usability of the technology.

Overall, the paper provides a valuable contribution to the field of assistive technology for the visually impaired, showcasing the potential of computer vision to enhance accessibility and independence for this population. The system developed by the authors represents a promising step towards creating touch-free, computer vision-based solutions for visually impaired individuals to explore and interact with their environments.

**Limitations:**

It requires a costly hardware for scene exploration. It doesn't give a scene description it is just an object detection model.

## 5. PROPOSED SYSYTEM

By implementing advanced computer vision techniques, our solution intends to alter the manner in which visually impaired people may access the web. We employ object identification, scene recognition, and photo segmentation to address the drawbacks of standard techniques like alt text. The main purpose is to automatically identify the important aspect of an image and deliver accurate, direct information to increase accessibility. Our approach also solves concerns with infographic and chart accessibility. To turn visual representations into comprehensible forms, we develop an AI-driven system that employ the graph-cut picture segmentation approaches. Using this strategy, people with visual impairments will have no issue accessing and interpreting sophisticated visual material. Through the elimination of these

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

challenges, our technology helps bridge the digital gap, empowers persons with visual impairments, and fosters a more inclusive online environment, all of which have far-reaching and revolutionary consequences on a worldwide scale.

## 6. HARDWARE AND SOFTWARE REQUIREMENTS

### 6.1 HARDWARE REQUIREMENTS:

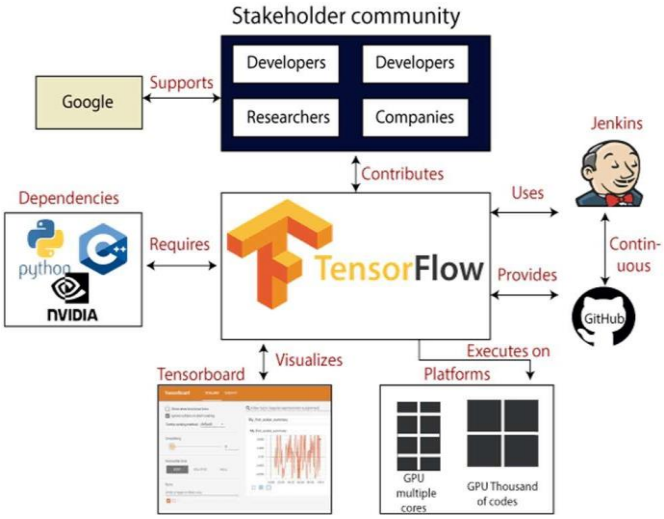
- Processor – Intel i5 (5th gen)
- RAM – 8 GB (min)
- Hard Disk – 20 GB
- Key Board – Standard Windows Keyboard
- Mouse – Two or Three Button Mouse
- Monitor – SVGA

### 6.2 SOFTWARE REQUIREMENTS:

- Operating system – Windows 7, 8, 10, 11, Mac OS
- Coding Language – Python
- Platform – Google Collab

## 7. PACKAGES USED

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. It was developed by the Google Brain team for Google's internal use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released an updated version, TensorFlow 2.0, in September 2019.TensorFlow can be used in a wide variety of programming languages, including Python, JavaScript, C++ and Java facilitating its use in a range of applications in many sectors. TensorFlow serves as a core platform and library for machine learning. TensorFlow's APIs use Keras to allow users to make their own machine learning models. In addition to building and training their model, TensorFlow can also help load the data to train the model, and deploy it using TensorFlow Serving.




## 8. ALGORITHM

- **Inception V3**: Inception v3 is Convolutional neural network which is developed by google for large image datasets. It is used for variety of tasks like object detection. It is a 48 layers convolutional neural network. It has a total 21,802,784 parameters. It has Conv2d, batch_ normalization, activation layers. It has inception module which is designed to capture features at multiple scales by concatenating outputs from various convolutional
- **Convolution 2d**: The image features is convoluted into the tensor for the input using convolution kernel. It applies a set of filters to the input image, which helps to identify important features in the image and ultimately make a prediction about its content. We first slide each filter across the image, computing the dot product between the filter weights and the corresponding pixels in the input image while applying the Conv2D layer. For each filter, this procedure generates a 2D activation map that highlights the areas of the image that have the highest correlation with that filter.
- **Maxpooling**: In addition to giving some degree of translation invariance, which implies that the network can recognize the same feature regardless of its position in the image, it aids in shrinking the size of the feature maps and extracting the most crucial features. In order to apply the MaxPooling layer to the image, We initially partition

the input image to 2x2 pix into 2x2 pixel non-overlapping parts then taking the maximum value within each region. By effectively reducing the input's spatial dimensions by a factor of 2, this process keeps the most crucial details. • Divide the dataset into training and testing sets.

- **Batch Normalization**: Layer that normalizes its inputs. Batch normalization applies a transformation that maintains the mean output close to O and the output standard deviation close to 1. Batch Normalization normalizes the activations of the previous layer during training by dividing by the batch's standard deviation and subtracting its mean. This normalization ensures that each activation function receives input that is within a similar range, which can speed up and stabilize training. • Write a code for displaying the evaluation result considering accuracy metrics.
- **LSTM**: Recurrent neural network (RNN) architectures such as LSTM are made to represent data sequences including time series, audio, and text. By utilising a unique memory cell and gating methods, LSTM networks are able to retain long-term dependencies in contrast to conventional RNNs, which experience the vanishing gradient problem. A memory cell at the center of the LSTM cell stores the network's long-term state. Three gating systems-the forget gate, the input gate, and the output gate-can access and alter the cell.
- **Attention Layers Network**: Attention networks are neural networks which will improve the accuracy when we need to focus on a thing in our input sequence. Attention layers helps to focus on like images, text interested area in the sequences. There are some areas in the sequences which are hidden, and it is important to note them. These are usually seen in a RNN networks for text tasks, where we need a context.

## 9. SOURCE CODE

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import os import json
from sklearn.model_selection
import train_test_split
from sklearn.utils import shuffle keras = tf.keras
#settingup config parameters
max_cap_len = 15
img_dimension = 299
num_words = 10000
encoding_size = 512
LSTM_size = 512
batch_size = 128
n_epochs = 15
Buffer_size = 1000
validation_and_test_split = 0.2
test_to_val_split = 0.5
num_examples = None
##data preprocessing
annotation_folder = '/annotations/'
annotation_zip = tf.keras.utils.get_file('captions.zip', cache_subdir=os.path.abspath('.'), origin = 'http://images.cocodataset.org/annotations/annotations_t rainval2014.zip', extract = True) annotation_file=os.path.dirname(annotation_zip)+'/anno tations/captions_train2014.json'os.remove(annotation_zi p)
# Download image files image_folder = '/train2014/' image_zip = tf.keras.utils.get_file('train2014.zip' cache_subdir=os.path.abspath('.'), origin = 'http://images.cocodataset.org/zips/train2014.zip', extract = True) PATH = os.path.dirname(image_zip) + image_folder os.remove(image_zip)
## Data Preprossesing def load_img(path): img = tf.io.read_file(path) img = tf.image.decode_jpeg(img, channels=3)
img = tf.image.resize(img, (img_dimension, img_dimension)) return img
def preprocess_func(path_index, caption):
#Reading the image path_index = tf.reshape(path_index, ()) path =
```

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

tf.gather(img_name_vector_train,indices=path_index)

img = load_img(path)

#/255.0 #Preprocessing text teacher_caption = caption[:-1] tar_caption = caption[1:]

h_and_c_init = tf.zeros((LSTM_size))

return (img,h_and_c_init,teacher_caption), tar_caption

#Creating an array to index each img path for reading path_index_vec_train = np.array(list(range(0,len(img_name_vector_train)))).res hape(-1,1) path_index_vec_val = np.array(list(range(0,len(img_name_vector_val)))).resha pe(-1,1) dataset1_train = tf.data.Dataset.from_tensor_slices(path_index_vec_train ) dataset2_train = tf.data.Dataset.from_tensor_slices(captions_train)

dataset = tf.data.Dataset.zip((dataset1_train,dataset2_train)) dataset = dataset.map(preprocess_func, num_parallel_calls=tf.data.experimental.AUTOTUNE) dataset1_val = tf.data.Dataset.from_tensor_slices(path_index_vec_val) dataset2_val = tf.data.Dataset.from_tensor_slices(captions_valid)

dataset_val = tf.data.Dataset.zip((dataset1_val,dataset2_val))

dataset_val = dataset_val.map(preprocess_func_val, num_parallel_calls=tf.data.experimental.AUTOTUNE)

dataset = dataset.shuffle(Buffer_size).batch(batch_size,drop_rema inder=True).prefetch(1)

dataset_val = dataset_val.shuffle(Buffer_size).batch(256,drop_remain der=True).prefetch(1) ##Building Model from keras.applications.inception_v3 import preprocess_input incep=keras.applications.inception_v3.InceptionV3(inpu t_shape=(img_dimension,img_dimension,3), include_top=False) incep.trainable=False incep.summary() encoder = keras.models.Sequential([ keras.layers.Lambda(preprocess_input,input_shape=(im g_dimension,img_dimension,3),name="preprocessing_la yer"),incep,keras.layers.Dense(encoding_size,activation= 'relu',name="encoding_layer"), keras.layers.Reshape((8*8,encoding_size),name="reshap e_layer")],name="Encoder") encoder.summary()

##Completing the model W1 = keras.layers.Dense(512,name="W1") W2 = keras.layers.Dense(512,name="W2") V = keras.layers.Dense(1,name="V") repeater = keras.layers.RepeatVector(8*8) doter = keras.layers.Dot(axes=1) concatenator = keras.layers.Concatenate() def attention_step(enc,h_prev): h = repeater(h_prev) score = tf.nn.tanh(W1(enc)+ W2(h)) alphas =tf.nn.softmax(V(score),axis=1) context = doter([alphas,enc]) return context

## Building Decoder encodings=keras.layers.Input(shape=(8*8,encoding_size) ) init_h = keras.layers.Input(shape=(LSTM_size)) init_c = keras.layers.Input(shape=(LSTM_size)) teacher_forcing = keras.layers.Input(shape=(1)) embedding_layer keras.layers.Embedding(words+1,256,) context_prev_tar_concat_layer=keras.layers.Concatenat e() decoder_lstm_layer=keras.layers.LSTM(LSTM_size,ret urn_state=True,dropout=0.2) decoder_dense_layer=keras.layers.Dense(words+1,activa tion='softmax') h = init_h c = init_c context = attention_step(encodings,h) embedds = embedding_layer(teacher_forcing) decoder_lstm_input=context_prev_tar_concat_layer([co ntext,embedds]) h,c=decoder_lstm_layer(decoder_lstm_input,initial_state =[h,c]) out = decoder_dense_layer(h) decoder=keras.models.Model([encodings,init_h,init_c,tea cher_forcing],[out,h,c]) decoder.summary()

## Define Training Custom Loop optimizer = keras.optimizers.Adam()@tf.function def train_step(img,init_state,teacher,target): with tf.GradientTape() as tape: encodings = encoder(img)

h = init_state

c = init_state

loss = 0

for i in range(max_cap_len+1): dec_inp = teacher[:,i:i+1] o, h , c = decoder([encodings,h,c,dec_inp]) loss += sparse_it_up(target[:,i],o)

total_loss = (loss / int(target.shape[1]))

trainable_variables = encoder.trainable_variables + decoder.trainable_variables gradients=tape.gradient(loss,trainable_variables)optimiz er.apply_gradients(zip(gradients,trainable_variables)) return loss, total_loss

'"This function is for forward passing features for calculating losses with no backprop"' def valid_step(img,init_state,teacher,target): with tf.GradientTape() as tape: encodings = encoder(img)

h = init_state

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

c = init_state

loss = 0

for i in range(max_cap_len+1): dec_inp = teacher[:,i:i+1] o, h , c = decoder([encodings,h,c,dec_inp])

loss += sparse_it_up(target[:,i],o)

return loss

## Training Model

EPOCHS = 5

prev_loss = np.inf

for epoch in range(EPOCHS):

start = time.time() total_loss = 0 for (batch, ((img, s, teacher) ,target)) in enumerate(dataset): batch_loss, t_loss = train_step(img, s, teacher ,target) total_loss += t_loss if batch % 50 == 0: print ('Epoch {} Batch {} Loss {:.4f}'.format( epoch + 1, batch, batch_loss.numpy() / int(target.shape[1])))

for (i, ((img, s, teacher) ,target)) in enumerate(dataset_val): val_loss += valid_step(img, s, teacher ,target)

val_loss = val_loss.numpy()/((i+1)*(max_cap_len+1)) print("val_loss=",val_loss)

#This code makes sure we only save the best val_loss score weights if val_loss < prev_loss:

print("Model_imporved. Saving..") encoder.save("encoder.hdf5")

decoder.save("decoder.hdf5")

else:

print("Model didn't imporve.")

prev_loss = val_loss

# Loading the best validation accuracy score weights encoder.load_weights("encoder.hdf5") decoder.load_weights("decoder.hdf5")

##Prediction functionThis function takes its input as path of an image and captions it

def caption_image(path):

image = load_img(path) encodings=encoder.predict(tf.reshape(image,(1,img_dim ension,img_dimension,3)))

texts = [""] h = np.zeros((1,LSTM_size))

c = np.zeros((1,LSTM_size))

for _ in range(max_cap_len + 1):

dec_inp = np.array(tok.word_index.get(texts[- 1])).reshape(1,-1)

#print(dec_inp) props,h,c = decoder.predict([encodings,h,c ,dec_inp])

props= props[0] idx = np.argmax(props) texts.append(tok.index_word.get(idx))

if idx == tok.word_index['']:

break

if tok.word_index.get(texts[-1]) != tok.word_index['']: texts.append('')

print(' '.join(texts))

plt.imshow(image/255.0)

plt.axis("off")

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

## 10. OUTPUT

```
Model: "model_2"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_20 (InputLayer)           [(None, 512)]        0
__________________________________________________________________________________________________
input_19 (InputLayer)           [(None, 64, 512)]    0
__________________________________________________________________________________________________
repeat_vector_1 (RepeatVector)  (None, 64, 512)      0           input_20[0][0]
__________________________________________________________________________________________________
W1 (Dense)                      (None, 64, 512)      262656      input_19[0][0]
__________________________________________________________________________________________________
W2 (Dense)                      (None, 64, 512)      262656      repeat_vector_1[0][0]
__________________________________________________________________________________________________
tf_op_layer_AddV2_3 (TensorFlow [(None, 64, 512)]    0           W1[1][0]
                                                                 W2[0][0]
__________________________________________________________________________________________________
tf_op_layer_Tanh_3 (TensorFlowO [(None, 64, 512)]    0           tf_op_layer_AddV2_3[0][0]
__________________________________________________________________________________________________
V (Dense)                       multiple             513         tf_op_layer_Tanh_3[0][0]
__________________________________________________________________________________________________
tf_op_layer_Transpose_6 (Tensor [(None, 1, 64)]      0           V[1][0]
__________________________________________________________________________________________________
tf_op_layer_Softmax_3 (TensorFl [(None, 1, 64)]      0           tf_op_layer_Transpose_6[0][0]
__________________________________________________________________________________________________
tf_op_layer_Transpose_7 (Tensor [(None, 64, 1)]      0           tf_op_layer_Softmax_3[0][0]
__________________________________________________________________________________________________
input_22 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
dot_1 (Dot)                     multiple             0           tf_op_layer_Transpose_7[0][0]
                                                                 input_19[0][0]
__________________________________________________________________________________________________
embedding_4 (Embedding)         (None, 1, 256)       2560256     input_22[0][0]
__________________________________________________________________________________________________
concatenate_10 (Concatenate)    (None, 1, 768)       0           dot_1[1][0]
                                                                 embedding_4[0][0]
__________________________________________________________________________________________________
input_21 (InputLayer)           [(None, 512)]        0
__________________________________________________________________________________________________
lstm_4 (LSTM)                   [(None, 512), (None, 2623488     concatenate_10[0][0]
                                                                 input_20[0][0]
                                                                 input_21[0][0]
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 10001)        5130513     lstm_4[0][0]
==================================================================================================
Total params: 10,840,082
Trainable params: 10,840,082
Non-trainable params: 0
__________________________________________________________________________________________________
```

```
Click here to ask Blackbox to help you code faster
caption_image(img_name_vector_test[1700])
                                                                                    Python
<sos> baby boy holding toothbrush in his hand <eos>
```

```
Click here to ask Blackbox to help you code faster
caption_image(img_name_vector_test[9888])
                                                                                    Python
<sos> group of planes flying in formation in sky <eos>
```

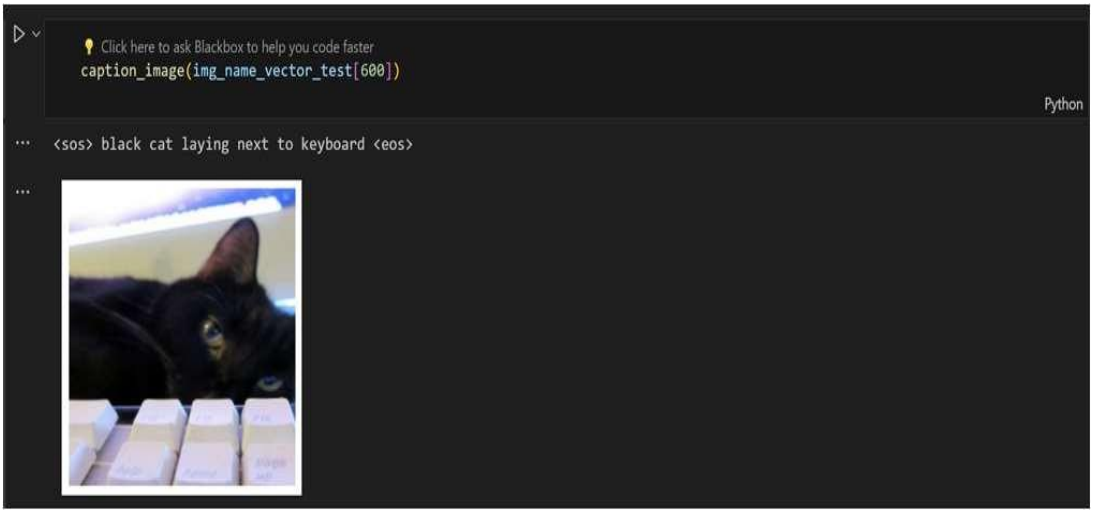
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

Impact Factor: 5.725


## 11. TESTING

The testing of the "Visual Discovery" system involves various methodologies to ensure its reliability, accuracy, and effectiveness. Here are key testing methodologies employed during different stages of development:

**TYPES OF TESTS**

**Unit Testing**: Test individual modules and components in isolation to ensure they function as intended. Verify that each unit of code, such as algorithms, preprocessing functions, and user interface components, produces the expected output.

**Integration Testing:** Evaluate the interaction between different modules and components to ensure seamless integration. Verify that the system components work together as a cohesive unit, detecting and resolving any issues related to data flow and communication.

**Functional Testing**: Validate that the system meets the specified functional requirements. Test core functionalities such as data analysis, predictive modeling, user interface interactions, and cross disciplinary adaptability.

**Performance Testing**: Evaluate the system's responsiveness, scalability, and resource usage under different conditions. Test the application's ability to handle large datasets and complex computations without compromising performance.

**Regression Testing:** Verify that new updates or modifications do not adversely impact existing functionalities. Re-run previously conducted tests to ensure that any changes have not introduced new errors or broken existing features.

**Security Testing**: Assess the system's resistance to unauthorized access, data breaches, and other security vulnerabilities. Implement measures to safeguard sensitive data processed by the application.

**Adaptive Learning and Continuous Improvement Testing**: Assess the system's ability to adapt and improve over time. Validate that the adaptive learning mechanisms are effectively enhancing the system's predictive capabilities based on new data and scenarios.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1583-1591

**User Acceptance Testing (UAT):** Involve end-users, researchers, and scientists in the testing process to gather feedback on the system's usability and effectiveness. Ensure that the system aligns with user expectations and fulfills their requirements. The model is a combination of computer vision and natural language processing traditional evaluation metrics doesn't work. We use Bilingual Evaluation understudy (Bleu). It is used mostly for valuation of the translation tasks. Since, our output is a caption which a natural language we are using Bleu score for evaluation Our model performs very good. sometimes it gives quality descriptions better than humans. Since the bleu score is 60. Our model is generating a very high quality and fluent translations. We are getting good understanding of what is happening in the image

```
[48]
from nltk.translate.bleu_score import corpus_bleu
print('Cumulative 1-gram: %f' % corpus_bleu( refernce,predictions, weights=(0.3, 0, 0, 0)))
print('Cumulative 2-gram: %f' % corpus_bleu( refernce,predictions, weights=( 0,0.3, 0, 0)))
print('Cumulative 3-gram: %f' % corpus_bleu( refernce,predictions, weights=(0, 0, 0.3, 0)))
print('Cumulative 4-gram: %f' % corpus_bleu(refernce, predictions,weights=(0, 0, 0,0.3)))

Cumulative 1-gram: 0.604923
Cumulative 2-gram: 0.409867
Cumulative 3-gram: 0.297250
Cumulative 4-gram: 0.216448
```

## 12. CONCLUSION

We conclude that Visual Discovery offers enormous promise in benefiting the daily lives of visually impaired persons by giving thorough scene descriptions and protecting them from potential threats. The inclusion of visuals offers a sophisticated comprehension of actions in their environment, allowing blind people to obtain useful insights. The Visual Discovery tool serves as an excellent mechanism for comprehending and navigating unknown situations, overcoming the limits of conventional solutions like blind canes or haptics. Our proposed Visual Discovery system offers a substantial improvement, giving detailed scene descriptions using a convergence of computer vision and natural language processing approaches. This revolutionary technique empowers blind and visually impaired persons, giving them with a real-time comprehension of their world. By integrating cutting-edge technology, this system not only promotes safer navigation but also offers doors for a more autonomous and informed living for the visually impaired.

This initiative serves as a promising step towards addressing accessibility barriers and creating inclusiveness for the visually impaired population.

## 13. REFERENCES

[1] Base Paper: Xixuan Julie Liu, Yi Fang. "Virtual Touch: Computer Vision Augmented Touch Free Scene Exploration for the Blind or Visually Impaired" 2021 2021 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW).

[2] Shubham Melvin Felix, Sumer Kumar, and A. Veeramuthu,"A Smart Personal Al Assistant for Visually Impaired People",2018.

[3] Prachi Waghmare Swati Shinde, Jayshree Katti, "Image Captioning using Neural Network Model",2022.

[4] Alec Radford, Jong Wook Kim Tao Xu | Greg Brockman Christine McLeavey, Ilya Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision", 2020.

[5] Wei Xie, Fangli Xu, Hongling Wang, Xiaoxu Han, and Xiaodan Zhang. "Graph-based reasoning for image captioning." Neurocomputing, vol. 484, pp. 39-50, 2021.

[6] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. "Show and Tell: A Neural Image Caption Generator". IEEE Conference on Computer Vision and Pattern Recognition