e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

Drishti - The Smart Application for Visual Assistance

Prof. Swetha P M [*1], Sangamesh[*2], Paritosh[*3], Sagar[*4], Nikhil[*5]

[*1]Professor, Department of Computer Science and Engineering, SJCE, Mysuru, Karnataka, India

[*2]Student, Department of Computer Science and Engineering, SJCE, Mysuru, Karnataka, India

[*3]Student, Department of Computer Science and Engineering, SJCE, Mysuru, Karnataka, India

[*4]Student, Department of Computer Science and Engineering, SJCE, Mysuru, Karnataka, India

[*5]Student, Department of Computer Science and Engineering, SJCE, Mysuru, Karnataka, India

ABSTRACT

Blindness is a broad word that describes a person's state when his or her perception of vision is disrupted or obstructed.Blind persons are split into two categories based on the severity of their condition: fully blind and those with residual eyesight (Low Vision).Blind searchers emphasize other sensory functions such as touching, smell, and hearing as a result of the loss/reduction of the sensory function of vision.As a result, one approach is to develop a mobile Drishti app that may assist the blind and assistance communities in dealing with and responding to blind individuals.This article is about a voice-activated programme that captures images from the camera when the user requests them and narrates the image situation in text and audio.

Keywords: Image, Feature extraction,Captioning,Research,Image vector,ResNet50,Text preprocessing.

I. INTRODUCTION

The Drishti app is described in this study as giving end-to-end support to visually impaired persons by presenting sceneric information in the form of voice.The user may utilize the voice command to launch and run the same app.The user is asked to provide their name at the start of the application.The visitor is then sent to the following page, where audio is used to transmit all of the information regarding the DRISHTI.

The user must say "START" to activate the camera, after which the webcam will click photos in response to the voice command "TAKE A PICTURE."The photographs that are clicked are sent to the backend to be learnt and labelled.After then, the captions are transformed to voice and sent to the user.This captioning and information conveyance continues until the user order "STOP DRISHTI" is issued.When the application hears "STOP DRISHTI," it shuts down.

II. METHODOLOGY

The main objective is to extract the image's characteristics and caption it.The features from the picture are extracted using ResNet50 in this article.The output of ResNet50 is divided into 2000 classes.The second final layer serves as the output since it comprises neurons with 2048 values that have image-like characteristics.

The text that serves as the caption for the visually impaired is generated using feature vector extraction.

III. SYSTEM IMPLEMENTATION

Dataset

It features two folders, one for photographs and the other for captions, using the Flickr 8K dataset.There are five different subtitles for each image.To accommodate the vast range of human ideas on each event, multiple captions for a picture are available.

Overview of the working

- Image feature extraction
- To create the caption, use the picture feature extracted vector.

e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

- Caption is converted to speech

Feature extraction

Image is transferred into ResNet 50 to extract the features of the image.The feature is converted into a vector of 2048 values.Resnet takes an image of size (224,224) and gives the output in 2000 classes. Use of the second last layer as the output as it contains the neurons of 2048 values which has the features of an image.

Preprocessing for caption dataset

To add the start of string(sos) and end of string(eos) to our captions. For example : cat sitting on a table

<sos> cat sitting on a table <eos>

<sos> → cat

<sos> cat -> sitting

<sos> cat sitting -> on

:

:

:

We get the next possible word from it.

It works like : IMAGE VECTOR + CAPTION ====> NEXT WORD

IV. CODE

Importing libraries

```
import numpy as np
import pandas as pd
import cv2
import os
from glob import glob
```

Image preprocessing

```
images_path = PATH FOR THE IMAGE FILES
images = glob(images_path+'*.jpg')
len(images)
```

To read the top 5 images and then convert them from BGR to RGB.This code snippet is just to understand the dataset.

```
import matplotlib.pyplot as plt
for i in range(5):
plt.figure()
img = cv2.imread(images[i])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

Import the ResNet50 model and use the last but one layer as the output layer.

```
incept_model = ResNet50(include_top=True)
from keras.models import Model
last = incept_model.layers[-2].output
modele = Model(inputs = incept_model.input,outputs = last)
modele.summary()
```

e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

Preprocess of all the images from the dataset,images are converted from BGR to RGB and then resize into 224,224 as this is the required input image size for ResNet50

```
images_features = {}
count = 0
for i in images:
img = cv2.imread(i)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224,224))
img = img.reshape(1,224,224,3)
pred = modele.predict(img).reshape(2048,)
img_name = i.split('/')[-1]
images_features[img_name] = pred
```

Text preprocessing

To read the captions for a particular image and then create a captions dictionary with image name as key and captions as value.

```
caption_path = PATH FOR THE CAPTION FILE
captions = open(caption_path, 'rb').read().decode('utf-8').split('\n')
captions_dict = {}
for i in captions:
try:
img_name = i.split('\t')[0][:-2]
caption = i.split('\t')[1]
if img_name in images_features:
if img_name not in captions_dict:
captions_dict[img_name] = [caption]
else:
captions_dict[img_name].append(caption)
except:
pass
```

Convert the text into lowercase and add ‘startofseq’ and ‘endofseq’ to the modified text.

```
def preprocessed(txt):
modified = txt.lower()
modified = 'startofseq ' + modified + ' endofseq'
return modified
for k,v in captions_dict.items():
for vv in v:
captions_dict[k][v.index(vv)] = preprocessed(vv)
Creating Vocabulary :
count_words = {}
for k,vv in captions_dict.items():
for v in vv:
```

e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

```
for word in v.split():
if word not in count_words:
count_words[word] = 0
else:
count_words[word] += 1
```

Encode the caption dictionary based on the frequency of words in the whole captions.

```
for k, vv in captions_dict.items():
for v in vv:
encoded = []
for word in v.split():
if word not in new_dict:
encoded.append(new_dict['<OUT>'])
else:
encoded.append(new_dict[word])
captions_dict[k][vv.index(v)] = encoded
```

Model

```
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
from keras.utils import plot_model
from keras.models import Model, Sequential
from keras.layers import Input
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Embedding
from keras.layers import Dropout
from keras.layers.merge import add
from keras.callbacks import ModelCheckpoint
from keras.layers import Dense, Flatten,Input, Convolution2D, Dropout, LSTM, TimeDistributed,
Embedding, Bidirectional, Activation, RepeatVector,Concatenate from keras.models import Sequential,
Model
embedding_size = 128
max_len = MAX_LEN
vocab_size = len(new_dict)
image_model = Sequential()
image_model.add(Dense(embedding_size, input_shape=(2048,), activation='relu'))
image_model.add(RepeatVector(max_len))
image_model.summary()
language_model = Sequential()
language_model.add(Embedding(input_dim=vocab_size, output_dim=embedding_size,
input_length=max_len))
language_model.add(LSTM(256, return_sequences=True))
```

e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

```
language_model.add(TimeDistributed(Dense(embedding_size)))
language_model.summary()
conca = Concatenate()([image_model.output, language_model.output]) x = LSTM(128,
return_sequences=True)(conca)
x = LSTM(512, return_sequences=False)(x)
x = Dense(vocab_size)(x)
out = Activation('softmax')(x)
model = Model(inputs=[image_model.input, language_model.input], outputs = out)
# model.load_weights("../input/model_weights.h5")
model.compile(loss='categorical_crossentropy', optimizer='RMSprop', metrics=['accuracy'])
model.summary()
```

Predictions

```
no = np.random.randint(1500,7000,(1,1))[0,0]
test_feature = modele.predict(getImage(no)).reshape(1,2048)
test_img_path = images[no]
test_img = cv2.imread(test_img_path)
test_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)
text_inp = ['startofseq']
count = 0
caption = ''
while count < 25:
count += 1
encoded = []
for i in text_inp:
encoded.append(new_dict[i])
encoded = [encoded]
encoded = pad_sequences(encoded, padding='post', truncating='post', maxlen=MAX_LEN)
prediction = np.argmax(model.predict([test_feature, encoded])) sampled_word = inv_dict[prediction]
caption = caption + ' ' + sampled_word
if sampled_word == 'endofseq':
break
text_inp.append(sampled_word)
plt.figure()
plt.imshow(test_img)
plt.xlabel(caption)
```

V. RESULT AND ANALYSIS

e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

Some of the results from the model are shown in the figure 1 and 2.

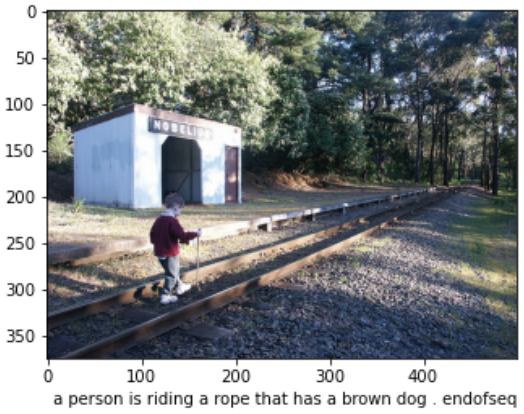

Figure 1: Sample result

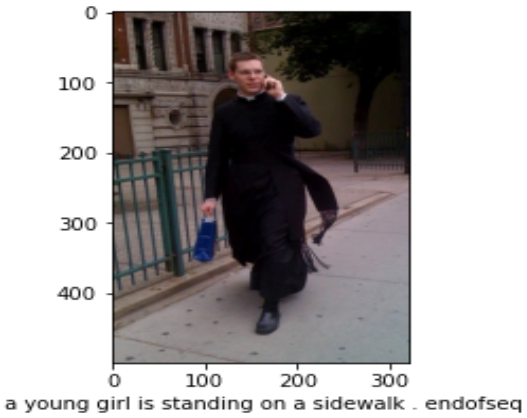

Figure 2: Sample result

This model achieves an accuracy of 73% with a loss of 1%.

Table 1. Comparison of model captions v/s correct captions

SN.	Image	Model caption	Correct caption
1	Figure 1	A person is riding a rope that has a brown dog	A person is walking on a brown track with a stick
2	Figure 2	A young girl is standing on a sidewalk	A young boy is standing on a sidewalk

VI. CONCLUSION

e-ISSN:
2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

The programme responds to the user's vocal command by taking an image, captioning it, and then converting it into speech.It aids the visually handicapped in understanding the situation.Drishti is easy to use for visually challenged people because it does not require any visual action.

VII. REFERENCES

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio and Dumitru Erhan, “show and tell: A neural image caption generator”

[2] Xinlei Chen and C. Lawrence Zitnick, “Mind's eye: A recurrent visual representation for image caption generation”

[3] Shuang Bai and Shan An, “A survey on automatic image caption generation”

[4] Takashi Miyazaki and Nobuyuki Shimizu, “Cross-Lingual Image Caption Generation”