Manzarah

A vision for the visually impaired

Shah Rukh Khan
Computer Science and Electrical Engineering
University Of Missouri: Kansas City
Kansas City, USA
srkhvx@mail.umkc.edu

***Abstract*—Manzarah is name of a software that is specifically designed and developed to fit the needs of the visually impaired. Manzarah uses Deep Learning and new models inspired from the already existing models but molded into different shape to avoid copy right infringement. Inspired by VGG16 and Inception Model it uses a multi modal approach. Manzarah is a caption generation and validation application built to help the visually impaired population in such a way that it not just generates captions for pictures instead it takes the whole data base and trains a hybrid model. Now a user can ask the application if the application can help find what he is looking for. The application takes sample pictures from user's surrounding and creates captions for them all. Now natural language processing and matching algorithm will try and find a relation between what the user asked and what the application could find in the images and provide a reasonable response. It can be a yes and a no or it can be a suggestion, depending on the nature of the question. Imaging the impact of this application on the lives of the visually impaired. There are a lot of basic caption generation models in the market but a visually impaired needs more than just a caption for a picture but an assistant to ask whether what the user is looking for is around him or where is the user standing or any such question which can be answered by using vision analysis on stills taken from camera of a phone or a professional camera. This can give them a hope and complete awareness of their surroundings. The main part of this application is accuracy and to check that we can use a standard set by the big companies such as Microsoft and Google etc.**

Keywords—VGG16,CNN, RNN, NLP (*key words)*

I. INTRODUCTION (*HEADING 1*)

This paper in its entirety is a complete implementation of how the technologies at hand can be used to help the impaired population. This paper targets one such application where the Show and Tell model has been combined with Ask and Report Model. This project is worth the extra effort because the market is full of application where the user just depends on what the application generates as a caption. Human nature has always been obsessed with having choice, by generating a simple strict caption for an image takes that choice away from the user and instead make the user feel as if the application is in charge. Thus, Ask and report model is different in a way that it lets the user control the information that the application generates by asking questions and keep questioning and getting different response that fits to full satisfy the needs. Thus, the paper is about an Application where a user asks the application a question pertaining to what is in his surrounding and the application takes pictures and reports back and answers any specific question of the user related to the vision surrounding him. Thus, giving a vision to the visually impaired thus, the name Manzarah (منظره i: e. Persian for Sight). It sheds light on the complete end to end integration and also accuracy of the model by running it on different Data sets provided by the fortune 500 companies. Further more the author has kept Microsoft's Caption-Bot as a standard to measure how good the Application performs against Microsoft's model. Once implemented this will be one of its kind in the market!

II. RELATED WORK

This section describes relevant background on RNN and Image Caption generation. Lately, a lot of research has gone into automatic captioning of images. This is so that the millions of pictures that get uploaded daily are correctly captioned and models are trained to help further the research and make the machines smarter. [1] first used the research of [3] coming up with a graphical model like the human-engineered features. [4] however showed the use of neural networks for caption generation of images and suggested a multi-model, demonstrating neural networks able to decode image representations from a CNN encoder. Top-down approaches: These were initial approaches and were followed by [5] and [6] using the more recent CNNs for encoding while replacing the forward feed algorithm as shown in [7] with an RNN, and in particular an LSTM. This was also demonstrated by using it on captioning videos. The name "Top Down" is due to the nature of works of all these showing images as the top layer of a big CNN. This produced models that could be trained from end to end. Bottom-up approaches: [8] here the whole problem was divided into two separate problems. One part was to train a CNN and RNN that maps images and small captions to the multimodal which gave really good results on getting information. later, they trained a Recurrent NN that is able to learn and combine the input generated from various objects detected initially. This improved the results because instead of looking at the whole picture, small fragments of information were generated pertaining to the image features extracted and thus combined to give complete meaning. Later these were further improved by utilizing a 3-stage pipelining and other algorithms but the foundation had been laid. But such accuracy comes with a price. Here the price was that the models were not end to end trainable. This weakness can be compensated by inducing attention to the models as shown in question-answering [9] and handwriting generation [10]. Attention basically makes the model to focus on just specific features or aspects of the input; and the model learns this itself. This has been shown in auto-encoders in [11]. This project directly uses simpler architectures and more basic models and layered CNN combined with LSTM RNN to generate Captions and then a Voice to Text to get a better user experience.

III. Proposed Work

The project is a combination of two projects. The 1st phase includes a complete implementation of the Caption generation Bot. for that we need a dataset of labeled pictures.

The first Stage has successfully been now completed with the implementation of the 2nd Model named the Inception V3 model. I faced a lot of issues on running the model on the same dataset since VGG16 requires different sized images where as InceptionV3 requires different sized images. So, it took a lot of time to finally implement the Model on the same dataset by resizing and by averaging. It is all mentioned in the code.

We then downloaded the Flickr8K dataset for the very same purpose. This data set has 8 thousand labeled pictures of both indoor and outdoor activities. This data set is rich enough to contain enough material to boost our accuracy yet small enough for a desktop to train models on it overnight.

A. The Model

This project when fully completed will have 3 fully implemented models. Previously I was only working with VGG 16 and couldn't implement the Inception Model, the project now has successfully been implemented on VGG16 and Inception. Soon enough I'll create a 3rd model on top of these two and see the effect on performance and accuracy .

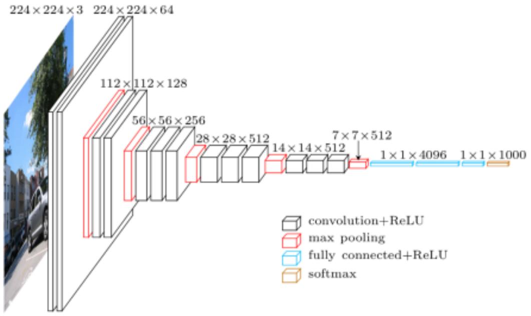


Figure 1VGG16 Network Diagram

This above model is the VGG 16 layered model and is a great model that fits the purpose of our project. Its weight and biases are pretrained and only needs new dataset to create and PKL file.

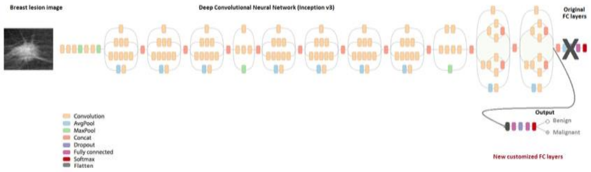
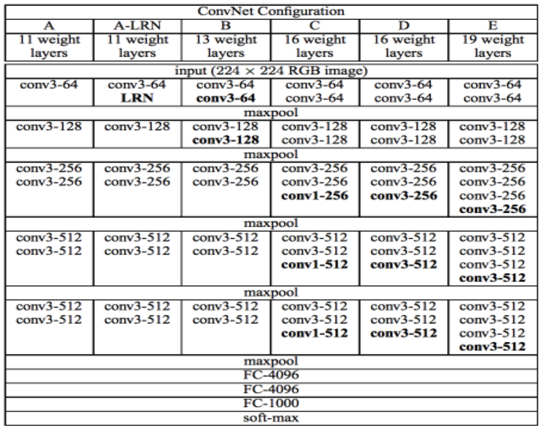


Figure 2Inception V3 Diagram

The model above is Google's Inception V3 model. It is also a pretrained model with weights and biases adjusted. Since these two have now been implemented and the results for the same test image with different models have been shared later in the paper, we will now have a final model on top of these two. The output of these two will go to the 3rd model which will get better result.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3 VGG16

B. Dataset

- The whole data base is labeled. The labels are in a text file which contains name of the image files followed by captions of the pictures.
- The Flickr8K data set is composed of mixed pictures and are a total of 8 thousand images divided in to a test and train part.

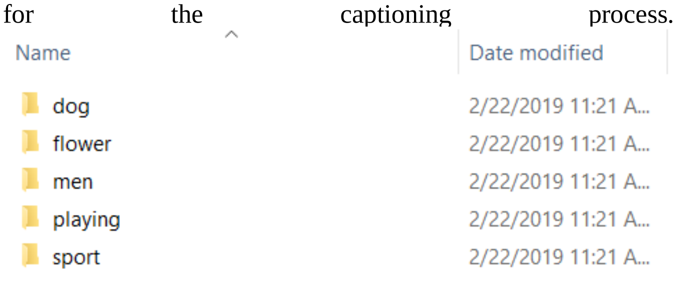
C. Data Preprocessing

I separated the whole data base into small classes containing only 5 keywords. Dogs, Men, Playing, Game and Sport.

The Data was Classified initially but due to issues of miss classification I am going with the whole Flickr8K dataset.

The database was separated by using 5 conditional statements and checking for keyword hits. Since Flickr8K data base, each picture has 5 captions then certain pictures had overlapping labels. So, we received a total of 5432 pictures in all these categories. We then went with using the whole data base since the difference was not much.

Once the pictures were separated then the Natural Language processing came to play. NLTK was used for the Text Stemming and Lemmatization. It was used to create Tokenizer later which will be used with BLEU score and then later generate Sentences for the captioning process.



Name	Date modified
dog	2/22/2019 11:21 A...
flower	2/22/2019 11:21 A...
men	2/22/2019 11:21 A...
playing	2/22/2019 11:21 A...
sport	2/22/2019 11:21 A...

Figure 4Classification

D. Some Common Mistakes

- The common mistakes that usually happen are in the preprocessing part. Classes inter mix, my men class contained images of dogs.
- Text lemmatization can usually crop words by cutting a bit too much from the end and then the word does not make sense.
- Stop words gets counted extra. So, a text file containing all possible stop words should be created and with help of that they should be removed. NLTK has a built function too but make it dynamic so that other unwanted words can be removed easily.
- These mistakes pushed me to go from classified data into original Flicker Dataset.

E. Algorithm and Pseudocode

- Pictures were reshaped into 224x224 sized images for the VGG16 model requirements.
- Pictures were reshaped into 299x299 sized images for the Inception V3 model requirements.
- NLTK was used to preprocess the text labels. This created Description files. Convert all words to lowercase and remove all punctuation. Remove all words that are one character or less in length and also which contain numbers.
- This label data and the pictures features are used to create tokenizer. Since we have two model now thus we have 2 tokenizers now.
- Now time to introduce the model of this project here.

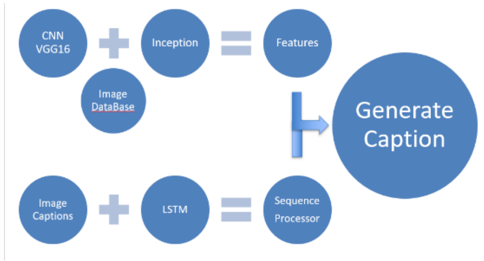


Figure 5Model

IV. Implementation and Evaluation

This project took 14 hours on a laptop containing GTX 1060 6GB and had 16GB 1.8GHz Ram. This project is heavy on a laptop but still light enough to work seamlessly without requiring a desktop or cloud computing. The above-mentioned time duration was by mistake for CPU powered Laptop. I actually didn't know that tensorflow was not working on the GPU but instead on the CPU. Thus, it took 14 hours initially. I successfully configured the GPU based TensorFlow and the time was reduced to mere 4 hours. It is still a lot time because when run on 2 models it takes almost 9 hours.

So, the Whole caption generation model is a combination of CNN and RNN.

CNN is a convolutional Neural Network mostly implemented on image dataset and extracted features.

RNN is a Recurrent Neural Network. Both are Neural network but one works across Space (CNN) whereas the other works across time (RNN).

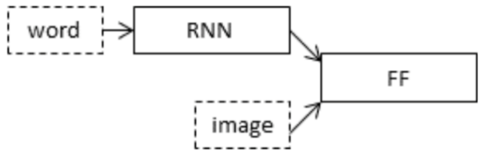


Figure 6captioning model

It works as follows on a training and testing picture.

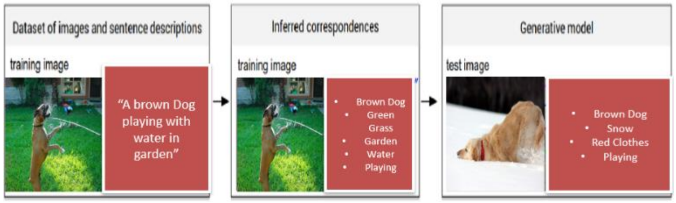
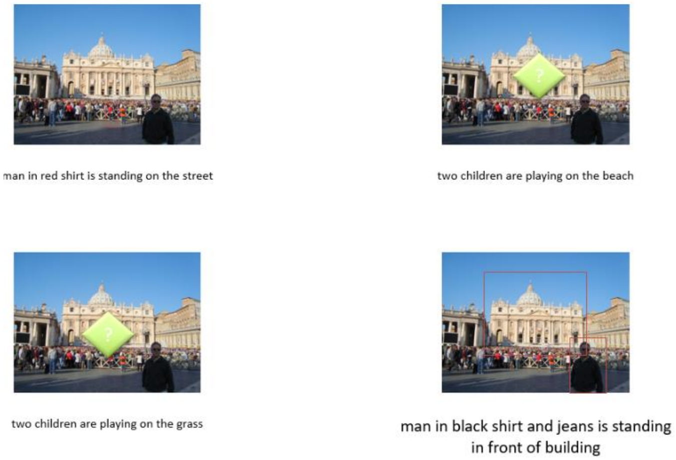


Figure 7 Training and Testing

Since I used a Laptop to train the model thus, I used sequential or batch processing. Instead of running the whole dataset for several epochs I divided the data set into 20 sets and trained models on the segmented data. So, I have 20 Models instead of 1. Some give same result where as some give different but overall, I get almost 5 to 6 unique captions.

Same goes for the Inception V3 model now, I was able to get a few unique captions.

My aim for now is to combine these segregated models into one and then do the same for Inception and later combine those models too. Once the two models have been trained and combined, I will use its outputs to a 3rd model. Which will work on top of these two and generate the best caption by taking suggestions from the two already most prevailing models.



man in red shirt is standing on the street

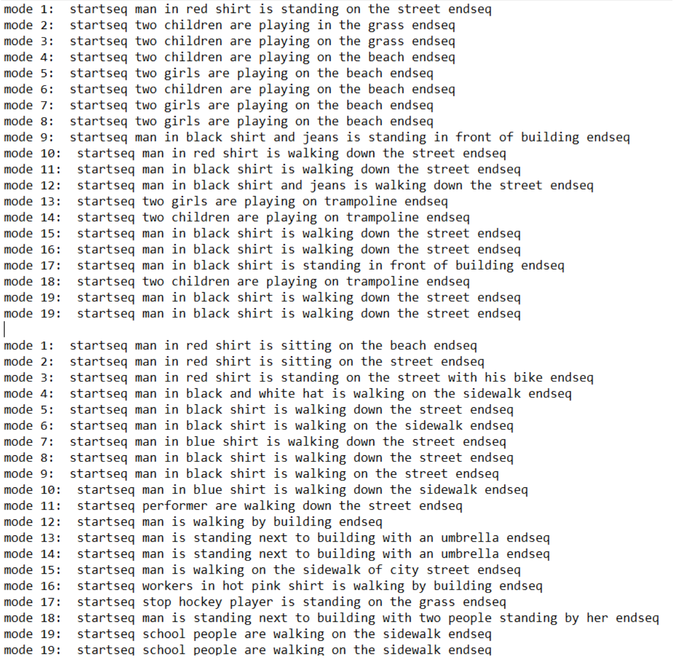
two children are playing on the beach

two children are playing on the grass

man in black shirt and jeans is standing in front of building

Figure 8 unique outputs of my 20-model system

These are unique results from the 20 models that were generated as a result of sequential data processing.
I ran the models on a single image and generated a total of 40 captions.

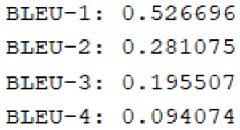


```
mode 1:  startseq man in red shirt is standing on the street endseq
mode 2:  startseq two children are playing in the grass endseq
mode 3:  startseq two children are playing on the grass endseq
mode 4:  startseq two children are playing on the beach endseq
mode 5:  startseq two girls are playing on the beach endseq
mode 6:  startseq two children are playing on the beach endseq
mode 7:  startseq two girls are playing on the beach endseq
mode 8:  startseq two girls are playing on the beach endseq
mode 9:  startseq man in black shirt and jeans is standing in front of building endseq
mode 10:  startseq man in red shirt is walking down the street endseq
mode 11:  startseq man in black shirt is walking down the street endseq
mode 12:  startseq man in black shirt and jeans is walking down the street endseq
mode 13:  startseq two girls are playing on trampoline endseq
mode 14:  startseq two children are playing on trampoline endseq
mode 15:  startseq man in black shirt is walking down the street endseq
mode 16:  startseq man in black shirt is walking down the street endseq
mode 17:  startseq man in black shirt is standing in front of building endseq
mode 18:  startseq two children are playing on trampoline endseq
mode 19:  startseq man in black shirt is walking down the street endseq
mode 19:  startseq man in black shirt is walking down the street endseq

mode 1:  startseq man in red shirt is sitting on the beach endseq
mode 2:  startseq man in red shirt is sitting on the street endseq
mode 3:  startseq man in red shirt is standing on the street with his bike endseq
mode 4:  startseq man in black and white hat is walking on the sidewalk endseq
mode 5:  startseq man in black shirt is walking down the street endseq
mode 6:  startseq man in black shirt is walking on the sidewalk endseq
mode 7:  startseq man in blue shirt is walking down the street endseq
mode 8:  startseq man in black shirt is walking down the street endseq
mode 9:  startseq man in black shirt is walking on the street endseq
mode 10:  startseq man in blue shirt is walking down the sidewalk endseq
mode 11:  startseq performer are walking down the street endseq
mode 12:  startseq man is walking by building endseq
mode 13:  startseq man is standing next to building with an umbrella endseq
mode 14:  startseq man is standing next to building with an umbrella endseq
mode 15:  startseq man is walking on the sidewalk of city street endseq
mode 16:  startseq workers in hot pink shirt is walking by building endseq
mode 17:  startseq stop hockey player is standing on the grass endseq
mode 18:  startseq man is standing next to building with two people standing by her endseq
mode 19:  startseq school people are walking on the sidewalk endseq
mode 19:  startseq school people are walking on the sidewalk endseq
```

The first 20 captions are for the VGG 16 layered model where as the following 20 captions are for the Inception V3 Model. We can see that some captions are quite similar where as others are different, Now the trick is to get a sensible caption out of these 40 different captions. Maybe create a new caption by taking into account these 40 captions.

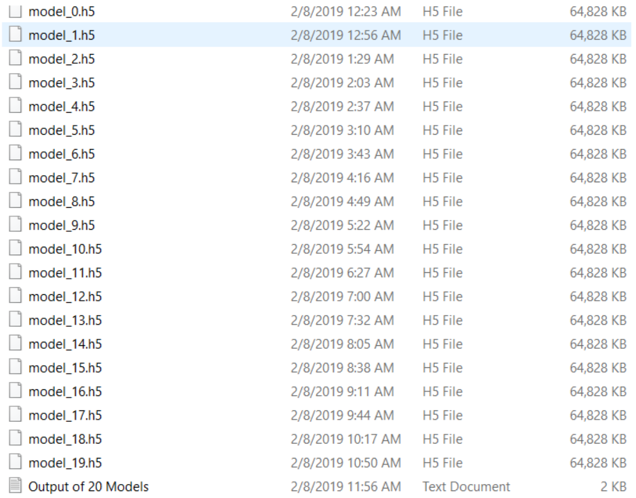
The BLEU score for the Project came out as below:



```
BLEU-1: 0.526696
BLEU-2: 0.281075
BLEU-3: 0.195507
BLEU-4: 0.094074
```

Once the Third model is in the works, I will try to implement CIDER and ROGUE also.

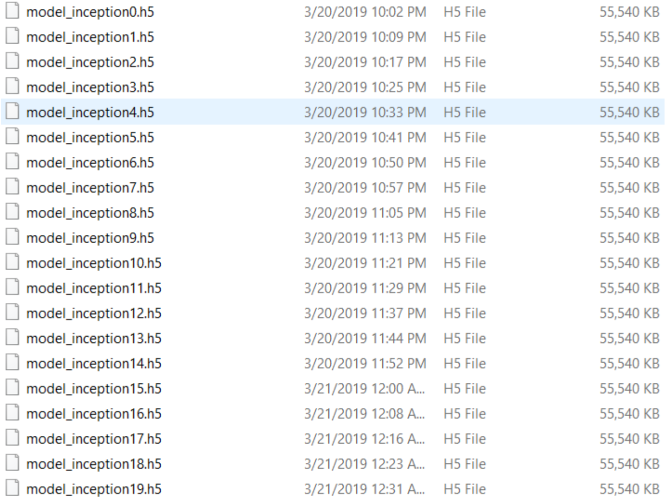
Models for VGG16 are shown below:



model_0.h5	2/8/2019 12:23 AM	H5 File	64,828 KB
model_1.h5	2/8/2019 12:56 AM	H5 File	64,828 KB
model_2.h5	2/8/2019 1:29 AM	H5 File	64,828 KB
model_3.h5	2/8/2019 2:03 AM	H5 File	64,828 KB
model_4.h5	2/8/2019 2:37 AM	H5 File	64,828 KB
model_5.h5	2/8/2019 3:10 AM	H5 File	64,828 KB
model_6.h5	2/8/2019 3:43 AM	H5 File	64,828 KB
model_7.h5	2/8/2019 4:16 AM	H5 File	64,828 KB
model_8.h5	2/8/2019 4:49 AM	H5 File	64,828 KB
model_9.h5	2/8/2019 5:22 AM	H5 File	64,828 KB
model_10.h5	2/8/2019 5:54 AM	H5 File	64,828 KB
model_11.h5	2/8/2019 6:27 AM	H5 File	64,828 KB
model_12.h5	2/8/2019 7:00 AM	H5 File	64,828 KB
model_13.h5	2/8/2019 7:32 AM	H5 File	64,828 KB
model_14.h5	2/8/2019 8:05 AM	H5 File	64,828 KB
model_15.h5	2/8/2019 8:38 AM	H5 File	64,828 KB
model_16.h5	2/8/2019 9:11 AM	H5 File	64,828 KB
model_17.h5	2/8/2019 9:44 AM	H5 File	64,828 KB
model_18.h5	2/8/2019 10:17 AM	H5 File	64,828 KB
model_19.h5	2/8/2019 10:50 AM	H5 File	64,828 KB
Output of 20 Models	2/8/2019 11:56 AM	Text Document	2 KB

Figure 9 Sequential 20-Model

Models for Inception V3 are shown below:

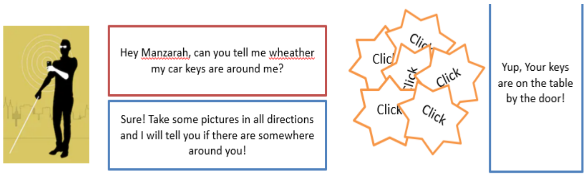


model_inception0.h5	3/20/2019 10:02 PM	H5 File	55,540 KB
model_inception1.h5	3/20/2019 10:09 PM	H5 File	55,540 KB
model_inception2.h5	3/20/2019 10:17 PM	H5 File	55,540 KB
model_inception3.h5	3/20/2019 10:25 PM	H5 File	55,540 KB
model_inception4.h5	3/20/2019 10:33 PM	H5 File	55,540 KB
model_inception5.h5	3/20/2019 10:41 PM	H5 File	55,540 KB
model_inception6.h5	3/20/2019 10:50 PM	H5 File	55,540 KB
model_inception7.h5	3/20/2019 10:57 PM	H5 File	55,540 KB
model_inception8.h5	3/20/2019 11:05 PM	H5 File	55,540 KB
model_inception9.h5	3/20/2019 11:13 PM	H5 File	55,540 KB
model_inception10.h5	3/20/2019 11:21 PM	H5 File	55,540 KB
model_inception11.h5	3/20/2019 11:29 PM	H5 File	55,540 KB
model_inception12.h5	3/20/2019 11:37 PM	H5 File	55,540 KB
model_inception13.h5	3/20/2019 11:44 PM	H5 File	55,540 KB
model_inception14.h5	3/20/2019 11:52 PM	H5 File	55,540 KB
model_inception15.h5	3/21/2019 12:00 A...	H5 File	55,540 KB
model_inception16.h5	3/21/2019 12:08 A...	H5 File	55,540 KB
model_inception17.h5	3/21/2019 12:16 A...	H5 File	55,540 KB
model_inception18.h5	3/21/2019 12:23 A...	H5 File	55,540 KB
model_inception19.h5	3/21/2019 12:31 A...	H5 File	55,540 KB

CONCLUSION

The project is unique in its nature that it is not a computer telling you what it can see but it is more like an assistant that answers to what you ask and the possibilities are limitless since with the slight tilt and change in color condition the features drastically change. Thus, for each image we have a new response vector.

The assistant is there to facilitate the visually impaired by using the already existing CNN and RNN. Furthermore, I will try if I am able to use the new images taken to be inculcated to the database and after sufficient data has been accumulated then the data base is merged with the original and the model is retrained when the user is not actively using it. Thus, increasing accuracy and making it to learn as it works. Inducing such conscious can only increase accuracy of the model and think about its accuracy after 5 years of use and 10 years of use. With increased computational power the process will become more seamless.



Hey Manzarah, can you tell me wheather my car keys are around me?

Sure! Take some pictures in all directions and I will tell you if there are somewhere around you!

Yup, Your keys are on the table by the door!

Figure 10 Manzarah

References

The paper could, in no way, not have been written without the previous work done in the field by the notables. They all require special mention and due citation.

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

[2] Ding, G., Chen, M., Zhao, S. et al. Cogn Comput (2018). https://doi.org/10.1007/s12559-018-9581-x

[3] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV'10, pages 15–29, Berlin, Heidelberg, 2010. Springer-Verlag.

[4] Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. CoRR, abs/1411.2539, 2014.

[5] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. CoRR, abs/1411.4555, 2014.

[6] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. CoRR, abs/1411.4389, 2014.

[7] Xinlei Chen and C. Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. CoRR, abs/1411.5654, 2014.

[8] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. CoRR, abs/1412.2306, 2014

[9] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. CoRR, abs/1506.07285, 2015

[10] Alex Graves. Generating sequences with recurrent neural networks. CoRR, abs/1308.0850, 2013

[11] Karol Gregor, Ivo Danihelka, Alex Graves, and Daan Wierstra. DRAW: A recurrent neural network for image generation. CoRR, abs/1502.04623, 2015