Human Action Recognition

|  |  |  |  |
| --- | --- | --- | --- |
| S Shiv Mohith  Electronics & Communications Department  PES University  *sshivmohith@pesu.pes.edu* | S Vijay  Electronics & Communications Department  PES University  *svijay@pesu.pes.edu* | Sanjana V  Electronics & Communications Department  PES University  *sanjanav@pesu.pes.edu* | Dr. Niranjana Krupa  Professor in Electronics & Communication Department  PES University  *bnkrupa@pes.edu* |

*Abstract*—The estimated number of visually impaired people in the world is 285 million in which 39 million people are blind and 246 million people have low vision. This paper aims at assisting visually impaired people through Deep Learning (DL) by providing a system which can describe the surrounding as well as answer questions about the surrounding of the user. The paper proposes two models, a VQA (Visual Question Answering) model and an image captioning model. The Image caption model is a CNN and RNN architecture that incorporates a form of attention while captioning. VQA is a CNN and LSTM based model that answers questions concerning the input image. These two models are integrated to form a single system. The user will be provided with an option to choose between describing the surroundings or ask a question about the surroundings and he/she will be provided with a description or an answer respectively.

*Keywords* – Deep Learning, Visual Question Answering, Image Captioning, Real Time

# INTRODUCTION

Visually impaired is the reduced capability to see to a point that causes difficulties that aren't rectifiable by most of the means available, like glasses or canes. Visual impairment can cause people to face difficulties with mundane tasks like walking, reading, driving, or socializing. This disability creates dependence and decreased access to productive activities. To address this issue several technical products are available in the market. In the current market, all products available solely focus on identifying and reading text and describing their surroundings to the user. But they do not provide an option of answering the questions that the user might have. We have implemented Visual Question Answering alongside Image Captioning to present a full-fledged experience to the users.

There has been an increase in research in image captioning and Visual Question Answering in the recent years that combines Computer Vision (CV), Natural Language Processing (NLP), and Knowledge Representation & Reasoning (KR).

Image captioning involves object detection as well as expressing relationships between the objects and describing it in natural language. In this model we also try to incorporate a form of attention to parts of the image while captioning to provide meaningful captions.

In this paper, we have implemented open-ended Visual Question Answering (VQA). A VQA system accepts an image and a natural language question as the inputs and predicts a natural language answer for the question. A Multi-Layer Perceptron (MLP) model and a Long Short Term Memory (LSTM) based model have been implemented for the VQA task.

The main contribution of our work is building a product that integrates both the image captioning model and the VQA model unlike any other product in the current market. Speech to Text and Text to Speech modules have also been constructed to enable real-time demonstration.

# RELATED WORK

Recent work has shown significant progress in the field of Computer Vision and Natural language processing. There has been a lot of study and research in the field of VQA as well [1], [2], [3], [4]. The work in [1] involves an operated device that generates binary questions for the images from a pre-defined template and a limited set of vocabulary. In [2], a Bayesian framework has been implemented for the prediction where the answers come from a pre-defined set like colors or objects.

The recent effort in VQA involves LSTM and CNN models which are implemented on the VQA dataset. LSTM models are used for extracting the question features and for the linguistic content of the answers and CNN model for extracting the image features are fused to generate an answer in [3]. In [4], LSTM for the question representation and the CNN for the image features are combined such that the LSTM question representation is conditioned on the CNN image features at each time step, and the final LSTM hidden state is used to sequentially decode the answer phrase. Our work closely relates to the work done in [6], where the CNN image features and the LSTM question representation which are computed independently are fused and provided to a softmax layer for prediction. Another approach has been followed in [5], where the question embedding and the CNN image features are passed to an MLP followed by a softmax layer.

In [6], a talking application for mobile phones has been developed where the dataset consists of pictures captured by blind people which makes it more user-friendly for the visually impaired and the blind. A new approach has been included in [7], where the semantic representation of the question is used as a query to search for regions in an image that are related to the answer. A large number of applications and concepts emerge from Visual Question Answering.

There has been extensive work done in the image caption generation task. A significant contribution was by Farhadi [8], where three different spaces and the triplet mapping was used to generate sentences. There were also several retrieval based approaches for caption generation [9]. A joint embedding model of words and images was developed by [10]. There were several drawbacks to these traditional approaches.

With the advent of Neural networks being employed in computer vision problems. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) were successfully used in image captioning tasks. As in [11] where they used a CNN along with a log-Bilinear model to generate captions. Then came multimodal RNN (m-RNN) [12] which used a CNN and an RNN in a multimodal space to generate novel image captions. Several others had started to experiment with RNN and LSTMs to generate captions like [13,14]. In [15] they proposed a novel model of a CNN and a Bidirectional RNN to represent images and words respectively and map them to the same multimodal embedding space. A completely radical approach was by [16] which had a fully convolutional localization network architecture that could be used not only on images but on videos for description. Then a novel encoder-decoder framework, where CNN acted as the image encoder and RNN as the sentence decoder approach was employed in [17] who coined the term Neural Image Caption (NIC).

There had been some work in incorporating attention into neural networks for computer vision tasks like [18, 19]. A novel approach was proposed by [20] where an adaptive attention model via a visual sentinel was used to generate image captions. The most successful attempt in employing attention in machine translation was by [21] which employs a novel RNN encoder-

decoder approach which is coined as RNNsearch to generate captions. The image captioning model in this work is an implementation of [22], which is an encoder-decoder framework [17] which incorporates attention like that in [21] but with a variation to generate captions. In the future, we would like to implement [22] as our base image caption model which uses a control signal in the form of a sequence of words or image regions to generate constrained region level image captions

# DATASET

## The Microsoft Common Objects in COntext (MS COCO) Dataset

The image captioning model has been trained on the MS COCO dataset. It covers over 90 common object categories. The dataset consists of instances that can be labelled easily like a person, chair, and car. The dataset is classified into - iconic object images, iconic scene images, and non-iconic images. Each image has five captions generated using Amazon Mechanical Turks. The training and validation datasets together consist of 616,435 captions for 123,287 images and the testing dataset consists of 379,249 captions for 40,775 images.

## Visual Question Answering Dataset

The dataset contains 82,783 images and 2,48,349 Question-Answer pairs. The images have been selected from the MS-COCO (Microsoft Common Objects in Context) dataset. The dataset has been split into a training dataset, validation dataset, and test dataset. The test dataset includes test-dev, test-standard, test-challenge, and test-reserve. A user interface was created to collect three questions for every image. Ten answers were gathered from different subjects for each question to deal with these variations and confusion. All answers responded were made lowercase, articles and punctuations were erased, and numbers were converted to digits.

# PROPOSED SYSTEM

## A. Image Captioning

**Pre-processing**

The image has to be resized to 299x299 which is the defined input size of Inceptionv3. The pixel values are normalized in the range -1 to 1 which matches the format that inceptionv3 was trained on. The image features are taken from the last convolution layer which has a size 8x8x2048. All captions are annotated with a <start> and <end> token. Utilizing a tokenizer, all the captions are tokenized, split the sentences into words, unnecessary characters are removed, and based on occurrence numbers are assigned to them, thereby converting a sequence of words into a sequence of numbers, and simultaneously vocabulary of unique words are created. The size of the vocabulary is limited based on a threshold occurrence of a word, from which the top 10000 words are considered. Any word not present in this vocabulary is assigned a special token <UNK>. Then word-to-index and index-to-word mappings are created. Finally, all the sequences are padded with zeros to the same max length to keep the length uniform.

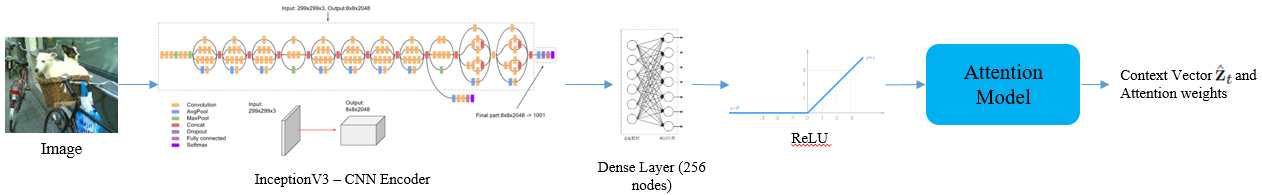


Figure 1: Image Caption – CNN Encoder & Attention Module

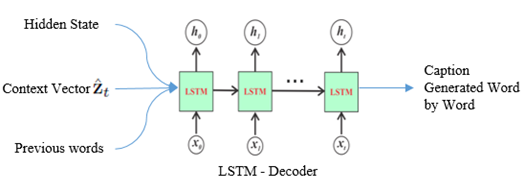


Figure 2: Image Captioning – LSTM Decoder

**Model**

As depicted in Figure 1, an image is passed to an Inception V3 network which acts as a CNN Encoder. The features extracted by the CNN Encoder are a set of vectors, each of which is a D-dimensional representation of a part of the image. They are fed to a dense layer with 256 nodes and then passed to the attention model to generate the context vector, Zt

The context vector, is computed using the input feature vector. For each input vector, which corresponds to a part of the image, a weight is computed which denotes the attention given to that location. The weights are computed using the attention model, founded by Bahdanau [21], for which an MLP model with the previous hidden state as input is used.

The hidden and memory state of the LSTM are computed using the annotation vectors fed to two MLPs.

At every time step, the network focuses on different parts of the image as the hidden state varies with the time steps. Once the weights have been computed, the context vector is calculated using the attention function.

Finally, the context vector, previous hidden state, and previously generated words are provided as inputs to the LSTM network, which acts as the LSTM Decoder, as shown in Figure 2, which produces a caption by generating one word at every time step.

## Visual Question Answering

**Pre-processing**

The input images are resized to 224X224X3 and converted to a suitable range acceptable by the VGG-16.VGG-16 pre-trained with ImageNet weights is used to extract image features. The top 1000 frequent answers among the 17000 different answers present in the dataset are chosen and considered as 1000 different classes converting it to a classification problem.

**MLP based model**

As given in Figure 3, the image features are extracted using CNN (VGG-16) and a 4096-dimensional image vector is extracted. The words are converted to vectors using the Word2Vec ***en\_core\_web\_md*** from the Spacy library and a 300-dimensional question vector is generated. The input to the first dense layer is the concatenation of the question and the image vectors. The output of the first dense layer is again given to a second dense layer. Each dense layer consists of 1024 nodes with tanh as the activation function. The output layer consists of 1000 nodes. A softmax layer is utilized to compute a probability distribution over all the answers. The loss is calculated using categorical cross-entropy and the weights are updated using the backpropagation algorithm.

**LSTM based model**

As depicted in Figure 4, this model utilizes VGG16 in extracting image features and multi-layer LSTM to extract question features. The VGG16 pre-trained with ImageNet weights is utilized to obtain the 4096-dimensional feature vector of an image from the final hidden layer of VGG-16. Then this image feature vector is given to a dense layer with 1024 nodes. The array of 300-dimensional word vectors, obtained after the question is passed through the word embedding, is fed to a multi-layer LSTM. Each 300-dimensional vector (1 word) is given to an LSTM cell. Each LSTM cell has a hidden state with 512 nodes. The hidden state of the last LSTM cell is taken as the question feature vector. This question feature vector is given to a dense layer with 1024 nodes.

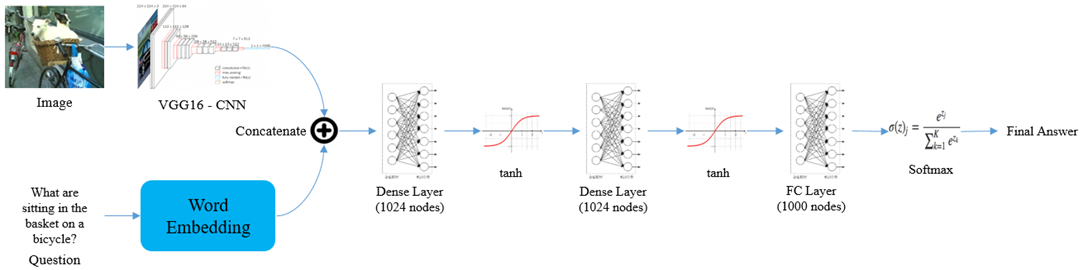


Figure 3: MLP based VQA Architecture

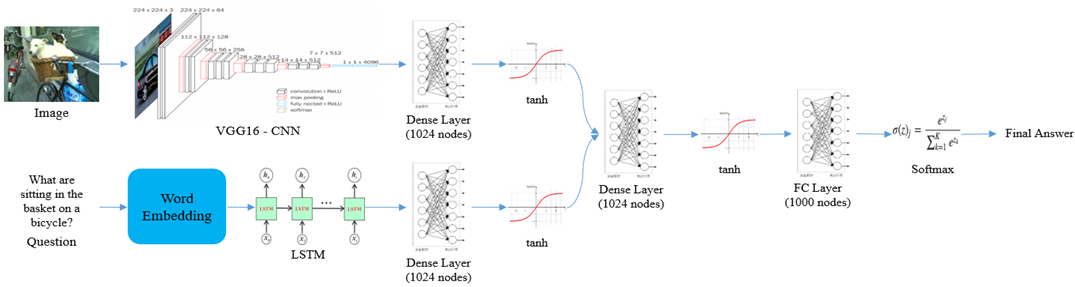


Figure 4: CNN & LSTM based VQA Architecture

Following this, the output of the image dense layer and question dense layer are concatenated and given to another dense layer with 1024 hidden nodes (all the dense layers are followed by a tanh activation function). This output of the latter is given to a fully-connected layer with 1000 output nodes. Next, a softmax layer is introduced to obtain the predicted probabilities.

The loss is calculated using categorical cross-entropy and the weights are updated using BPTT (Back Propagation through Time).

## Text to Speech

This component has been built to convert the answer generated by the VQA model and the caption generated by the image captioning model to audio for the user to listen to. It was achieved using a Python library which is a modifiable and personalized speech-specific sentence tokenizer. This module can read of text any lengths, while maintaining appropriate annotation, abbreviations, and decimals. This ensures that the speech converted is properly annotated and no information is lost during the conversion from text to speech.

## Speech to Text

The user’s question is converted to text for the VQA model. Python's extensive libraries aids in converting the speech to text. The energy thresholds have been set according to the eternal noise level to ensure accurate speech recognition. It can recognize different dialects as well, which makes it user friendly.

# METHODOLOGY

Text to Speech

Image Captioning

User

Camera Image

Question

Speech to Text

VQA

The above block diagram shows how different components discussed so far are integrated to form the complete product. Once the image is captured, it is given to the image captioning component which generates the caption. This generated caption is converted to audio to play it to the user. Once the user asks a question, this question is converted to text. This text and the captured image are given to the VQA component. The predicted answer is converted to audio to play it to the user.

The below flowchart represents how a user would use the product. The user is provided with 2 options, either to know about the surroundings or ask any question about the surroundings.

USER

Clicks a button to know about

the surroundings

Clicks a button to ask a question about the surroundings

Image of the surrounding is captured

Image is captured and user asks questions

Image is given to the image captioning model

Asked questions converted to text using text to speech module

Caption is generated

Question and Image given to the VQA model

Caption (text) to speech

Predicted answer (text) converted to Speech

# MODEL CONFIGURATION AND TRAINING

The image captioning model was trained on 1,00,000 samples. The model was trained using Adam optimizer. The batch size was set to 64. A buffer size of 1000 was used and the embedding dimension was set to 512. Sigmoid and tanh were the activation functions used. The loss was calculated using sparse categorical entropy.

The Visual Question Answering model was trained with 2,15,407 samples. The model was trained using the RMSProp optimizer with a learning rate of 0.001 and batch size 256. Tanh was the activation function used and the dropout layer of rate 0.5 was introduced after every dense layer. The loss was calculated using categorical entropy.

# EXPERIMENTAL RESULTS

**Image Captioning**

An example output of the image captioning model is shown in Figure 5.

The evaluation of the image captioning task was performed using the Bi-Lingual Evaluation Understudy (BLEU). The model was evaluated on 100 images from the test dataset where each BLEU score is calculated by comparing the generated caption with the reference captions of the corresponding image.

Table 1: Evaluation of Image Captioning Model

|  |  |  |
| --- | --- | --- |
| Type | Average | Highest |
| BLEU – 1 | 0.57 | 0.75 |
| BLEU – 2 | 0.39 | 0.46 |
| BLEU – 3 | 0.23 | 0.33 |

The average and the highest score is mentioned in Table 1. The average is taken across 100 images. The BLEU scores are calculated for different n-grams.

**Visual Question Answering (VQA)**

A few example outputs of both MLP and LSTM based VQA models are shown in Figures 7 and 8 respectively.

The two VQA models, MLP based model and LSTM based model, were evaluated by calculating the accuracy, precision, recall, and F1 score. The results were calculated for each type of question as well as for the overall accuracy. The different type of questions are the following,

* Yes or No type
* Numerical type
* Others

Table 2: Results of MLP based VQA model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Count | Accuracy | Precision | Recall | F1 score |
| Overall | 105175 | 38.0% | 0.25 | 0.38 | 0.26 |
| Yes or No | 45176 | 57.0% | 0.38 | 0.59 | 0.41 |
| Number | 13717 | 27.0% | 0.13 | 0.26 | 0.17 |
| Others | 46282 | 20.0% | 0.18 | 0.20 | 0.16 |

Table 3: Results of LSTM based VQA model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type | Count | Accuracy | Precision | Recall | F1 score |
| Overall | 105175 | 47.1% | 0.44 | 0.47 | 0.44 |
| Yes or No | 45176 | 64.0% | 0.63 | 0.64 | 0.63 |
| Number | 13717 | 28.0% | 0.17 | 0.28 | 0.17 |
| Others | 46282 | 36.0% | 0.33 | 0.36 | 0.36 |

Tables 2 and 3 mentions the accuracy, precision, recall and F1 score for each type of question and the overall accuracy of the MLP and LSTM based VQA models respectively.

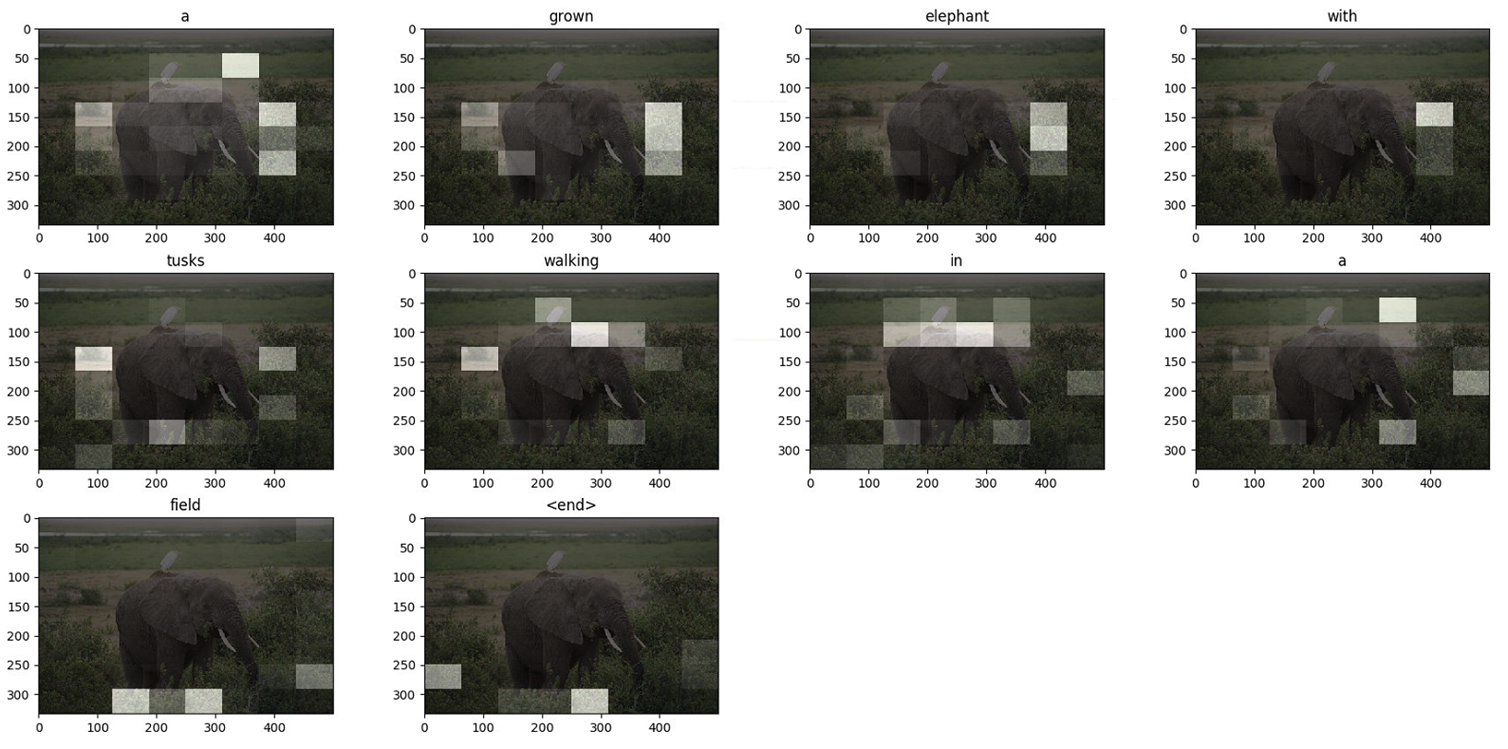


Figure 5: Output of image captioning model with attention at every time step for an example image

|  |  |
| --- | --- |
| Yes or No Type    Question: Are the people taking a selfie?  Predicted Answer: Yes  True Answer: Yes | Numerical Type    Question: How many objects in the picture are red?  Predicted Answer: 3  True Answer: 3 |
| Others Type    Question: What are these people waiting for?  Predicted Answer: Train  True Answer: Train | Failure Case    Question: What is pictured in the foreground?  Predicted Answer: Motor Cycle  True Answer: Parking meter |

Figure 6: Output of MLP based VQA model for each type of question and a failure case

|  |  |
| --- | --- |
| Yes or No Type    Question: Are right turned allowed?  Predicted Answer: No  True Answer: No | Numerical Type    Question: How many people are riding on each bike?  Predicted Answer: 2  True Answer: 2 |
| Others Type    Question: What is the table made of?  Predicted Answer: Wood  True Answer: Wood | Failure Case    Question: What color is the linoleum?  Predicted Answer: White  True Answer: Blue |

Figure 7: Output of MLP based VQA model for each type of question and a failure case

# CONCLUSION

The image captioning model is successfully implemented using CNN and RNN, to generate captions and describe the surroundings. The VQA (Visual Question Answering) model has been successfully built and two models have been developed for the same - MLP model and CNN, LSTM model.

The VQA model answers any questions that the user has regarding his/her surroundings.

The above mentioned two models have been integrated into a single product, with speech to text and text to speech components, which will assist the visually impaired and the blind in understanding and identifying their surroundings as well as answering their questions about the surroundings in their day-to-day life.

The product works successfully in real-time, which has been implemented using a webcam. The user is provided with two options either to choose a description of the surroundings or ask questions about the surroundings.

# FUTURE WORK

Recent work on image captioning, incorporate stacked LSTM networks and a control mechanism to the attention model. These approaches have proven to give a better description of the image. We would like to implement this approach to improve the accuracy of the image captioning model.

We would like to incorporate attention layers in the model to further focus only on the regions related to the question. The images in the MS-COCO dataset are very well taken and good quality images so these images lose practicality as visually impaired or the blind will not be able to take images with good quality. We would like to train our model on VizWiz dataset as the images are taken by the blind people and the questions are also asked by the blind.

# REFERENCES

[1] Geman, Donald, et al, "Visual turing test for computer vision systems.", Proceedings of the National Academy of Sciences 112.12, 2015.

[2] M. Malinowski and M. Fritz, “A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input”, In NIPS, 2014.

[3] Gao, Haoyuan, et al. "Are you talking to a machine? dataset and methods for multilingual image question.", Advances in neural information processing systems, 2015.

[4] M. Malinowski, M. Rohrbach, and M. Fritz., “Ask your neurons: A neural-based approach to answering questions about images”, In ICCV, 2015.

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “VQA: Visual question answering,” in The IEEE International Conference on Computer Vision (ICCV), 2015.

[6] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh., “VizWiz: Nearly Realtime Answers to Visual Questions”, User Interface Software and Technology, 2010.

[7] Yang, Zichao, et al., "Stacked attention networks for image question answering", Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[8] Farhadi A. et al., “Every Picture Tells a Story: Generating Sentences from Images”, Daniilidis K., Maragos P., Paragios N. (eds) Computer Vision – ECCV, 2010.

[9] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng., “Grounded compositional semantics for finding and describing images with sentences.”, TACL, 2014.

[10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean,

T. Mikolov, et al., “Devise: A deep visual-semantic embedding

Model”, NIPS, 2013.

[11] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel,

“Multimodal neural language models”, Proceedings of the 31st International Conference on Machine Learning (ICML-14). 595–603, 2014.

[12] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille, “Explain images with multimodal recurrent neural networks”, arXiv preprint arXiv:1410.1090, 2014.

[13] Xinlei Chen and C Lawrence Zitnick, “Mind’s eye: A recurrent visual representation for image caption generation”, Proceedings of the IEEE conference on computer vision and pattern recognition. 2422–2431, 2015.

[14] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille, “Deep captioning with multimodal recurrent neural networks (m-rnn)”, International Conference on Learning Representations (ICLR), 2015.

[15] Andrej Karpathy and Li Fei-Fei, “Deep visual-semantic alignments for generating image descriptions”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3128–3137. 2015.

[16] Justin Johnson, Andrej Karpathy, and Li Fei-Fei, “Densecap: Fully convolutional localization networks for dense captioning”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 4565–4574, 2016.

[17] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 652-663, 1 April 2015.

[18] Larochelle, Hugo and Hinton, Geoffrey E, “Learning to combine foveal glimpses with a third-order boltzmann machine”, NIPS, pp. 1243–1251, 2010.

[19] Denil, Misha, Bazzani, Loris, Larochelle, Hugo, and de Freitas, Nando, “Learning where to attend with deep architectures for image tracking”, Neural Computation, 2012.

[20] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher, “Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning”, IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[21] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate, “International Conference on Learning Representations (ICLR), 2015.

[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel and Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, arXiv:1502.03044, April 2016.

[23] Marcella Cornia and Lorenzo Baraldi and Rita Cucchiara, “Show, Control and Tell: A Framework for Generating Controllable and Grounded Captions”, arXiv:1811.10652, May 2019.