Sports Image Captioning Using CNN-RNN Architecture

Shin Thant
Data Science and AI Program
Asian Institute of Technology
Pathum Thani, Thailand
st121493@ait.asia

Phway Thant Thant Soe Lin
Data Science and AI Program
Asian Institute of Technology
Pathum Thani, Thailand
st121494@ait.asia

Rafifa Islam
Data Science and AI Program
Asian Institute of Technology
Pathum Thani, Thailand
st121707@ait.asia

Alok Aryal
Information Management
Asian Institute of Technology
Pathum Thani, Thailand
st121289@ait.asia

***Abstract*—Image captioning has been in limelight for sometime now. Image captioning can be done in two ways, bottom-up and top-down approach. We have implemented a top-down approach for image captioning. This paper aims at generating an image caption focused on sport images. As the younger generation are both into technology and sports, like the creation of sports blogs, blog pages, etc. it will be easier for people to just feed the images for social media and get an automated sports related image caption to them. We provide the images to a neural network after making all image sizes equal. We use CNN for encoder and RNN for decoder. Here, we also use the Bilingual Evaluation Understudy also known as BLEU which is used for accuracy calculation of the text generated.**

***Index Terms*—Keywords: image captions, CNN, RNN, BLEU**

I. Introduction

In the area of image or video recognition and interpretation, computer vision plays a crucial role. It deals with the computer's comprehension of pictures and photographs. Machines are capable enough these days to detect objects, images, specific events, etc. To distinguish objects, to sense their surroundings, people use their eyes and brain. Computer vision is a field that provides a system with a similar capability. Human Activity Recognition (HAR), the most significant function of computer vision is played. The computer is then trained in such a manner that it becomes adequately capable of detecting human gestures. The motivation behind the computer vision lies at the core of reproducing HAR. At the heart of imitating HAR lies the inspiration behind computer vision. It attempts to discern different human behaviour, such as tossing a pitch, driving, hitting the ball, playing sports, and many more, by certain kinds of observations in a given environment.

In this paper, computer vision applications are used to identify the different types of sports activities and then caption them. Sports play an essential role in everyone's life. All enjoy playing games or watching them. Many learn about sports at a very early age. Many people post sports-related images in their social media. While a lot of media outlets have news sections dedicated to sports, in which they need to caption the images with their news constantly. So, in this study, our goal is to caption sports-related images which can be used to help people to learn more about different types of sports as well as write captions for their sports-related images.

Two major ways to image captioning are: bottom-up and top-down. Bottom-up approaches produce objects detected and then aim to incorporate the items described into a caption [1]. On the other hand, top-down use different methods to have a semantic representation of an image that is then decoded into a caption using architectures, such as recurrent neural networks [2]. Our approach is the top-down image generation models. We use a deep convolutional neural network to create a vectorized representation of an image which is fed into a Long-Short-Term Memory (LSTM) network that generates captions of the images.

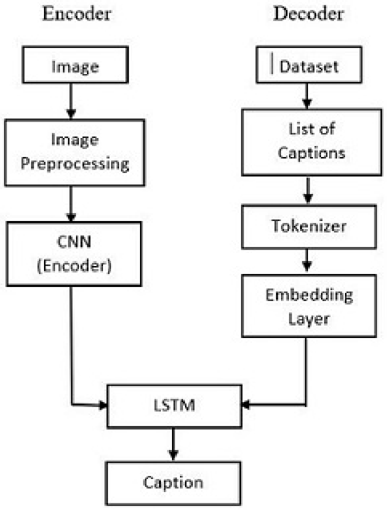


Fig. 1. Diagram

II. Literature Review

Over the years, different Natural Language Processing models have been used for automatic image captioning. In this section we have discussed two of the popular models.

CNN+RNN model: Many methods have been suggested to create image captions automatically, and CNN+RNN is the most common framework among them. To increase efficiency, much of the work aims to change the language model and the relation between the vision and language models. In this network, the deep convolutional network is used to classify images and sentence generation is done by a powerful Recurrent

Neural Network which is trained with the visual input so that RNN can keep track of the objects explained by the text. The m-RNN model uses a vanilla RNN combined with different CNNs [3]. The RNN hidden state and the CNN output are fed into a multimodal block at each time point to integrate the image and language features, and the next term is predicted by a softmax layer [3]. The model for neural image captioning (NIC) uses LSTM as a decoder [4]. The NIC model takes the image function vector as an input at the outset, and then the visual data is transferred along the recurrent direction [4]. An image is represented by a single vector in both m-RNN and NIC, which excludes multiple areas and artefacts in the image [4]. In the image captioning model, a spatial focus function is implemented, which enables the model to pay attention to various areas at each stage of time [4]. This model mostly used Flicker30k and COCO dataset.

CNN+CNN model: In this model, there are four modules: (1) vision module is adopted to observe the images; (2) language module is to model sentences; (3) attention module is for connecting the vision module with the language module; (4) prediction module takes input the visual features from the attention module and concepts from the language module and predicts the next word. The vision CNN extracts features from the image, and the language CNN is applied to create the sentence [5]. Meanwhile, the attention module and prediction module fuse the information from image and word contexts [5]. To allow feed-forward sentence generation, the convolutions in the CNN language they have used causal filters (depending only on the current and past words). They have also used k-1 zero vectors for each convolution layer in the language module to pad the sentence matrix along the length axis, where k denotes the size of the kernel [5]. A CNN without pooling is the basis of the language model, which is very distinct from the traditional RNN-based system. To memorize the context, RNNs follow a recurrent path, while CNN's use kernels and stack several layers to model the context [5]. For this experiment, they used Flicker30k and COCO dataset.

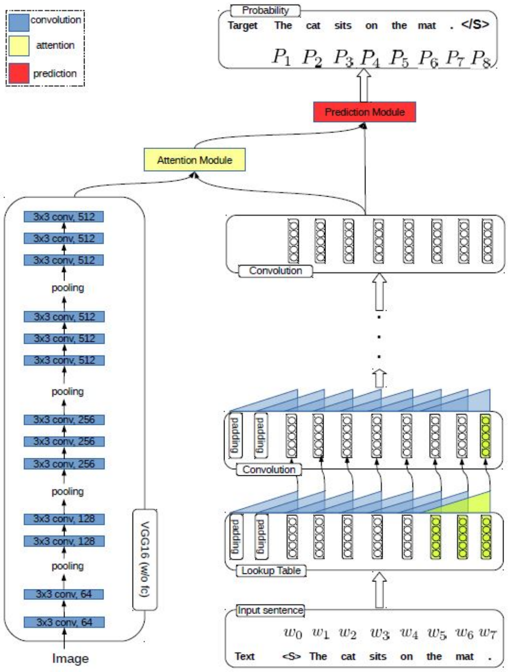


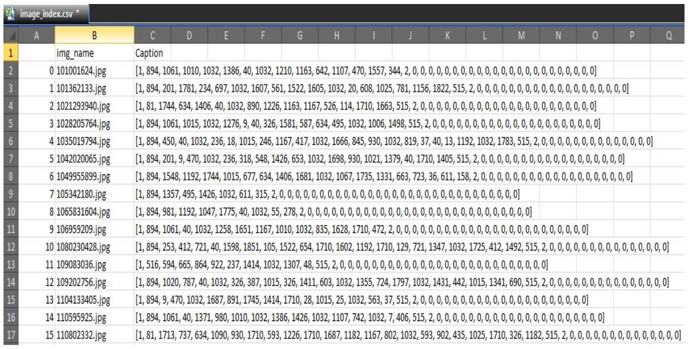
Fig. 2. CNN+CNN model

III. Methodology

A. Vectorization

Many machine learning algorithms require the input to be represented as a fixed-length feature vector. After reading the captions from the training csv file, all unique words are vectorized, as in the example sentence assuming start tag as $< s >$ = 1, end tag = $< e >$ = 2 and null tag = $< null >$ = 0. Example: A tennis player is trying to hit the ball . Captions in index = [1, 12, 35, 47, ...,30, 2 , 0, 0, 0]. Here, $< s >$ 1, $< e >$ 2, $< null >$ 0, a $< 12 >$, tennis $< 35 >$, player $< 47 >$, ... , ball $< 30 >$

A forward dictionary with key of 'word' and value of 'index' is made. A reverse dictionary with key of 'index' and value of 'word' is also made. A dictionary is saved as csv file including image name and index list of respective caption.



	A	B	C
1		img_name	Caption
2	0	101001624.jpg	[1, 894, 1061, 1010, 1032, 1386, 40, 1032, 1210, 1163, 642, 1107, 470, 1557, 344, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
3	1	101362133.jpg	[1, 894, 201, 1781, 234, 697, 1032, 1607, 561, 1522, 1605, 1032, 20, 608, 1025, 781, 1156, 1822, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
4	2	1021293940.jpg	[1, 81, 1744, 634, 1406, 40, 1032, 890, 1226, 1163, 1167, 526, 114, 1710, 1663, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
5	3	1028205764.jpg	[1, 894, 1061, 1015, 1032, 1276, 9, 40, 326, 1581, 587, 634, 495, 1032, 1006, 1498, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
6	4	1035019794.jpg	[1, 894, 450, 40, 1032, 236, 18, 1015, 246, 1167, 417, 1032, 1666, 845, 930, 1032, 819, 37, 40, 13, 1192, 1032, 1783, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
7	5	1042020065.jpg	[1, 894, 201, 9, 470, 1032, 236, 318, 548, 1426, 653, 1032, 1698, 930, 1021, 1379, 40, 1710, 1405, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
8	6	1049955899.jpg	[1, 894, 1548, 1192, 1744, 1015, 677, 634, 1406, 1681, 1032, 1067, 1735, 1331, 663, 723, 36, 611, 158, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
9	7	105342180.jpg	[1, 894, 1357, 495, 1426, 1032, 611, 315, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
10	8	1065831604.jpg	[1, 894, 981, 1192, 1047, 1775, 40, 1032, 55, 278, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
11	9	106959209.jpg	[1, 894, 1061, 40, 1032, 1258, 1651, 1167, 1010, 1032, 835, 1628, 1710, 472, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
12	10	1080230428.jpg	[1, 894, 253, 412, 721, 40, 1598, 1851, 105, 1522, 654, 1710, 1602, 1192, 1710, 129, 721, 1347, 1032, 1725, 412, 1492, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
13	11	109083036.jpg	[1, 516, 594, 665, 864, 922, 237, 1414, 1032, 1307, 48, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
14	12	109202756.jpg	[1, 894, 1020, 787, 40, 1032, 326, 387, 1015, 326, 1411, 603, 1032, 1355, 724, 1797, 1032, 1431, 442, 1015, 1341, 690, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
15	13	1104133405.jpg	[1, 894, 9, 470, 1032, 1687, 891, 1745, 1414, 1710, 28, 1015, 25, 1032, 563, 37, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
16	14	110595925.jpg	[1, 894, 1061, 40, 1371, 980, 1010, 1032, 1386, 1426, 1032, 1107, 742, 1032, 7, 406, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
17	15	110802332.jpg	[1, 81, 1713, 737, 634, 1090, 930, 1710, 593, 1226, 1710, 1687, 1182, 1167, 802, 1032, 593, 902, 435, 1025, 1710, 326, 1182, 515, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Fig. 3. csv file including image name and index list of respective caption

B. Encoder-Decoder

We adopted a transfer learning methodology to produce automatic captions for any given image. The encoder used in this model is the pre-trained ResNet50 model. This model uses a Convolutional Neural Network that encodes a fixed dimensional vector with the variable length input, which is used to decode the desired output sentence. The LSTM units is called "thought" vector which is connected to the vector containing the output of the fully connected layer in ResNet50. The encoder is used to retrieve the image's thought vector, which defines the content of the image. The size of the output of final layer of the 50 layers of ResNet is 2048 x 1, which is denoted as g(I) for an image I. We train a linear transformation of g(I) that maps it into the 300 x 1 input dimensions expected by our LSTM network. The representation is represented by:

$$CNN(I) = W(I)g(I) + b(I) \tag{1}$$

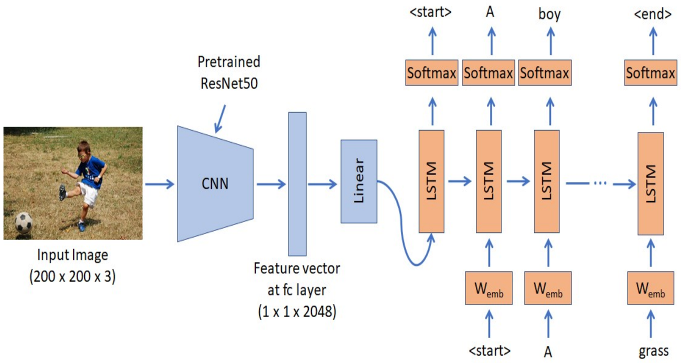


Fig. 4. Encoder and Decoder: CNN-RNN Based Framework

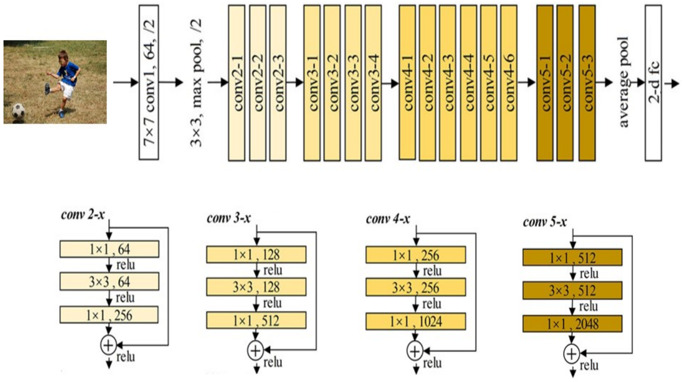


Fig. 5. ResNet50

We initialize a recurrent neural network with initial state equal to zero. We then feed the image representation CNN(I) in as the first input of a dynamic length LSTM, i.e. $x_{-1} = CNN(I)$. Subsequent inputs are the start of sentence token and all the words in the sentence, denoted by $x_t = W_e S_t$ for t = 0. . . N-1 where S_i is a $|V|$x1 one hot vector representing word i, So and S_N are one hot vector representing special start of sentence and end of sentence tokens, and We is a 512x$|V|$ word embedding matrix. Each hidden state of the LSTM emits a prediction for the next word in the sentence, denoted by $p_{t+1} = LSTM(x_t)$ for t = 0. . . . N-1. The model is fully described by the set of equations:

$$x_{-1} = CNN(I) \tag{2}$$

$$x_t = W_e S_t \tag{3}$$

$$p_{t+1} = LSTM(x_t) \tag{4}$$

for t = 0. . . . N-1.

Finally, we evaluate the parameters of the model at each iteration using the cross-entropy loss of the predictions on each sentence. The optimizer used is Adam optimizer for better results. The activation used in the last dense layer is linear activation.The loss function minimized is therefore:

$$J(S|I,\theta) = \sum_{t=1}^{N} log_{pt}(S_t|I,\theta) \tag{5}$$

where $p_t(S_t)$ is the probability of observing the correct word St at time t. This loss is minimized with regards to parameters in the set θ, which are all the parameters of the LSTM above, the parameters of the CNN and the word embeddings. The LSTM function can be described by the following equations where $LSTM(x_t)$ returns p_{t+1} and the tuple (m_t, c_t) is passed as the current hidden state to the next hidden state.

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \tag{6}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \tag{7}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{cx}x_t + W_{cm}m_{t-1}) \tag{9}$$

$$m_t = o_t \odot c_t \tag{10}$$

$$p_{t+1} = Softmax(m_t) \tag{11}$$

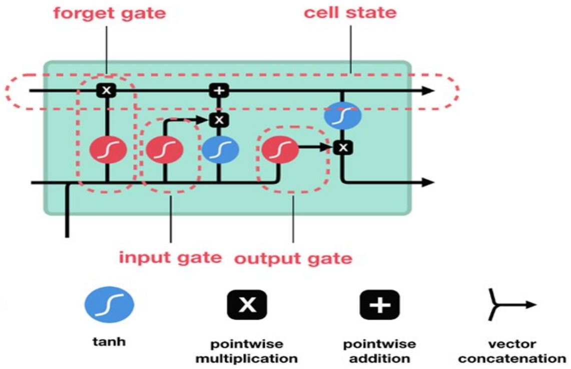
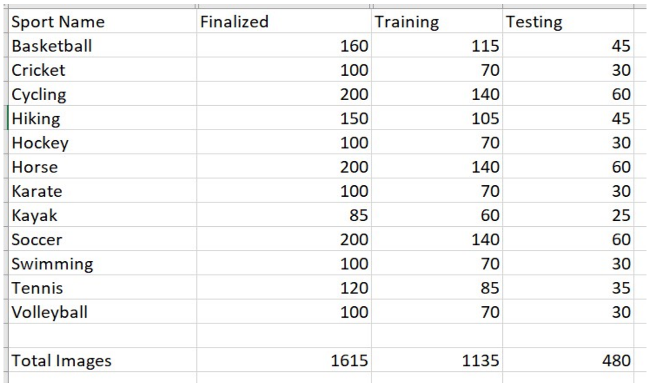


Fig. 6. Basic LSTM Framework

The forgotten gates f_t allows past memory cell states to be selectively ignored by the model, and the input gates will enable the model to selectively disregard portions of the present input. The o_t output gate then helps the model to filter its final secret state for the current memory cell. The combination of these gates includes the ability to learn and reset long-term dependencies dependent on specific inputs, and the ability to prevent gradients from disappearing and bursting.

Images: To represent images, we used a convolutional neural network to map images I to fixed-length vector representations. Deep convolutional neural networks have achieved the state-of-the-art performances in image classification in recent years. Specifically, we use the architecture of ResNet. Moreover, x = CNN(I) is a D_i x 1 vector, in which Di is the fixed dimension of any inputs to the LSTM. Words: We have used a word embedding of size D_i x $|V|$ where $|V|$ is the size of the vocabulary to represent the words.

Experiment: We measure the performance of this architecture on the dataset in which some of the data collected from Flicker30k while the rest is done manually. Our dataset comprises 1135 training images and 480 validation images. In the training data, there are, on average, 4 captions for each image. We have used BLEU-4 to measure the success of our models. Bilingual Evaluation Understudy (BLEU) measure how good the machine translation system is. BLEU compares n-grams of the reference translation and count the number of matches. As long as the machine is generated pretty close to any of the references provided by human, the BLEU score will be high [6].



Sport Name	Finalized	Training	Testing
Basketball	160	115	45
Cricket	100	70	30
Cycling	200	140	60
Hiking	150	105	45
Hockey	100	70	30
Horse	200	140	60
Karate	100	70	30
Kayak	85	60	25
Soccer	200	140	60
Swimming	100	70	30
Tennis	120	85	35
Volleyball	100	70	30
Total Images	1615	1135	480

Fig. 7. Dataset

IV. RESULTS AND ANALYSIS

Using ResNet50 model as an encoder with 50 hidden layers and LSTM network (using 1 LSTM layer) as decoder with number of epochs set to 200 for optimum performance taking 1135 images from the dataset as training data and vocabulary size of 1742 unique words.

Then we test the system with 480 images from the testing dataset. The loss is around 0.0014 and the average BLEU score is 0.21835.

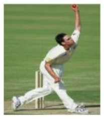
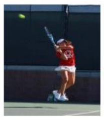
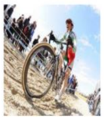
Image			
Predicted Caption	A man in a white athletic suit is throwing a ball in a cricket game .	A tennis player is running across the court with a racquet in his hand .	A little boy wearing a green shirt is riding on the back of a bicycle built for two with
Actual Caption	A cricketer in a white dress standing behind the line is about to bowl in a cricket game	A woman in a red shirt and white skirt playing tennis with a blue racket .	A middle-aged man in a bicycle race trying his hardest on the sand surrounded by watching crowd .
BELU Score	0.44554	0.44573	0.31720

Fig. 8. Prediction of images in test dataset and BLEU score

Figure 8 shows the images, the predicted caption, the first original caption and the BLEU score of the image from the testing dataset. Figure 9 shows the images, the predicted caption, the first original caption and the BLEU score of the image where caption is wrong but BLEU score is high from the testing dataset. Figure 10 shows the images, the predicted caption, the first original caption and the BLEU score of the image where caption is correct but BLEU score is low.

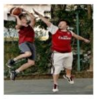
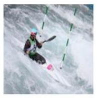
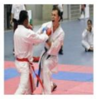
Image			
Predicted Caption	A young boy is beginning to shoot a basketball as a crowd watches the basketball court .	A boy with a blue and helmet swimming in a pool wearing a yellow flotation device .	A man in a white uniform who is the bowler is about to thrown the ball .
Actual Caption	Two guys playing basketball both midair one blocking while the other is attempting a jump shot .	A man in a kayak wearing a bib with 5 on it and going through wave of water in river	A man wearing a red belt and a another man wearing a blue belt are having a game of karate .
Note	Wrong caption but high score	Moderately correct/ Good BLEU	Wrong caption/ bad BLEU
BELU Score	0.36215	0.37142	0.15204

Fig. 9. Prediction of images in test dataset where caption is wrong but BLEU score is high

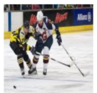
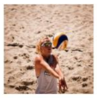
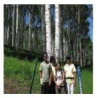
Image			
Predicted Caption	Two hockey players are skating on the ice .	A female basketball player wearing a red uniform is holding a basketball and looking up while an arm is	A person with a backpack and walking stick
Actual Caption	Two hockey players one in yellow and one in white competing against each other to gain control of the hockey puck .	A blond volleyball player in a gray sleeveless shirt and sunglasses bumps a blue	Group of four people in hiking gear standing in front of a forest of trees .
Note	Correct caption/ Bad BLEU	Moderately wrong caption/ moderate BLEU	Moderately correct caption/ Bad BLEU
BELU Score	0.13967	0.26304	0.09672

Fig. 10. Prediction of images in test dataset where caption is correct but BLEU score is low

V. Conclusion

We have implemented a caption generator for pictures that are all related to different kinds of sports. We have tried doing it with twelve different sports. It gives a description of the image in English language. With the use of convolution neural network which is used as an encoder for image input and later a recurrent neural network which helps in generating text for the image. We have used ResNet50 as our encoder and for the decoder we have chosen LSTM. We have collected a total of 1615 images which we have divided into 1135 training images and 480 testing images. The use of BLEU score is implemented to check the accuracy. The average BLEU score we got is 0.2183.For future work, we could add more intricate step by step images of different sport's actions. In our BLEU calculation, only the first caption is taken into consideration. We want to improve the BLEU calculation to be able to take all the caption into consideration. We believe this will improve the BLEU score as most of the generated captions are correct.

References

[1] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, ”Every picture tells a story: Generating sentences from images,” In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, Berlin, Heidelberg, 2010, pp. 15–29.

[2] X. Chen and C. L. Zitnick, ”Learning a recurrent visual representation for image caption generation,” *CoRR*, abs/1411.5654, 2014.

[3] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, ”Deep captioning with multimodal recurrent neural networks (m-rnn),” *ICLR*, 2015.

[4] Q. Wu, C. Shen, L. Liu, A. Dick, and A. v. d. Hengel, ”What value do explicit high-level concepts have in vision to language problems?,” *CVPR*, 2016.

[5] Q. B. Wang and A. B. Chan, “CNN Hengelo: Convolutional Decoders for Image Captioning,” *Computer Vision and Pattern Recognition*, May 2018. Available doi: arXiv:1805.09019.

[6] K. Papineni, S. Roukos, T. Ward, and W. Zhu,”BLEU: A Method for Automatic Evaluation of Machine Translation,” In*Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, 2002, pp. 311-318.