Natural Language Description of Videos

Omkar Acharya
Pune Institute of Computer Technology
Pune, India
acharyaomkar01@gmail.com

Gurnur Wadhwani
Pune Institute of Computer Technology
Pune, India
gurnursmailbox@gmail.com

Parag Ahivale
Pune Institute of Computer Technology
Pune, India
ahivale_parag@yahoo.com

Nehal Belgamwar
Pune Institute of Computer Technology
Pune, India
nehal.belgamwar@yahoo.com

Abstract*— Giving the user the choice to watch the video or read its description even without playing it, or both, we present a model that generates natural language description of videos in English language. For this purpose, we are using deep learning algorithms with both convolutional and recurrent structure. Our system extracts features from the video frames using a pre-trained Convolutional Neural Network viz. VGGNET (16 layer) that is trained on ImageNet dataset. Caffe's python module is used for the same. We feed these features to a LSTM (Long Short Term Memory) to generate the description for the input video. We trained the LSTM model on MSCOCO dataset and used Chainer framework for the related processing. Our system can find its application in video search engines that could provide better search results using video description instead of the existing systems that mainly rely on video titles. Also such a system can help the blind to comprehend video content.

Keywords*— Deep Learning, Convolutional Neural Network, Recurrent Neural Network, LSTM, Caffe, Chainer.

I. INTRODUCTION

Language is the only medium of interaction in our everyday lives and Human Language Technologies [5] have helped to boost cultural growth. Audio visual interactivity combined with the above techniques appear to be advancing in the field of communication with the help of artificial intelligence [1]. A visually descriptive language as a medium supports the above cause. Image captioning or recognizing and describing videos solves the visual symbol grounding problem. A machine doing this job may more cost as to doing it manually. This paper works on the same guidelines to build an effective system to automatically generate natural language description of videos into English language. It creates an analogy between the video content and the sentence generation process. This can be obtained by learning from the words' relations and applying them to form the sentence.

Here, we propose Convolutional Neural Network (CNN) [1, 2, 6], a novel architecture for recognizing features from images. CNN is made up of neurons having weights and biases. Also it comprises of more than one convolutional layer, pooling layers and one or two fully connected layers. CNNs are easier to train with fewer parameters.

Input video:

Expected output: *A man in white T shirt with a rod in hand.*
Human Interpretation: *A baseball player is having a shot. / A man in white jersey is playing baseball. / A man in white shirt with a red helmet is holding a baseball bat.*
Fig. 1: Our system is generating natural language description for the input video in English as above.

Recurrent Neural Networks (RNNs) are used here to form a sentence [3, 5]. They form the next word based on its understanding of previous words. They contain at least one feedback connection and make use of sequential information. For predicting next word in a sentence we need to know the word before it. RNNS then perform the same operation over a series of vectors, hence the name recurrent. They solve the problems of speech recognition, translation, language modeling, image captioning, etc. Here, we use Long Short Term Memory (LSTM), a very special type of recurrent neural network which works better than the standard version. LSTMs prove to be beneficial where RNNs create long-term dependency problem [7]. They can remember information for long periods of time. Specifically, we show that LSTM-type models provide an improved description of image recognition on videos. Each frame of the video is analyzed by the convolutional network pretrained on 1.2M+ images from ImageNet dataset with their respective captions. We trained the LSTM model on 122 thousand images of the MSCOCO dataset and vocabulary of 8000 words to produce the image description.

II. RELATED WORK

The research on this topic is fundamentally based on Natural Language Processing (NLP). [9] Includes recognizing objects in images with a fixed set of categories. They work on the subject verb object mechanism. Advancement to this is finding inter-phrase semantic similarity in sentences [4], which can be done using generalized latent semantic analysis. But it is less accurate. The use of Conditional Random Field (CRF) to predict the label for an input image [1] uses Decoding using language models for surface realization. Some systems have their scope limited to specific images, e.g. news images [16]. In content selection phase, it tells the article topic and decides how to verbalize the content by surface realization. To leverage the strengths of neural networks, Long Term Recurrent Neural Networks (LRCN) model is used to predict image features [3]. It applies to time varying inputs and outputs. But advancement leads to increase in model size and hence high running time.

Our work is one step ahead of above efforts by providing description in sentential form. The primary challenge towards this goal is to provide relevant one to one correspondence from images to vocabulary. For this purpose, we dig insight into Deep Learning. Deep Neural Networks provide efficient outputs with powerful learning algorithms.

III. METHOD OVERVIEW

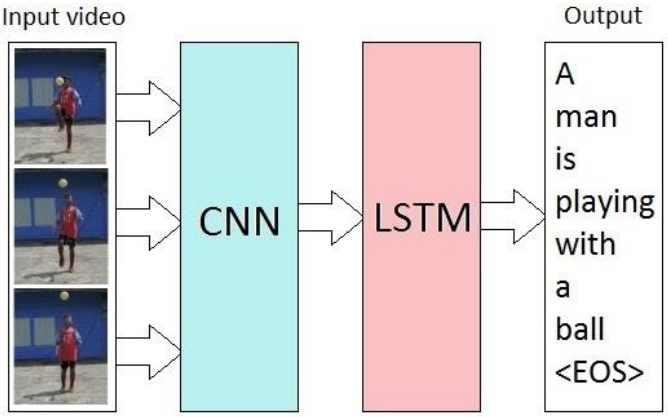

Fig.2 represents the overview of our system.

For an input video:

1. Frames are generated for each newly detected scene.
2. These frames are fed to the CNN for feature extraction.
3. Extracted features are then provided as an input to the LSTM model.
4. The description of all the frames as generated by LSTM is combined to get the final natural language description of the video.

The detailed explanation for the above process has been elaborated in the following sections. FFMPEG is used for frame generation (Section IV.i.). Details about CNN and LSTM are covered in (Section IV.ii.) and (Section IV.iv.) respectively.

IV. OUR APPROACH

IV.i. Frame Generation

For detection of scene change in the input video as well as for frame generation, we use FFMPEG utility. We say there is a scene change, when there is at least 30% dissimilarity between two consecutive frames. Using this criterion by setting the threshold, we are able to automatically discard similar scenes to avoid generating description for similar scenes. FFMPEG *select='gt (scene\, 0.4)'* flag is used for the same.

IV.ii. Convolutional Neural Network

To extract features from the frames we use a pre-trained VGG-16 Caffe model. Though this model is pretty heavy it gave us better accuracy on test videos as compared to Alex-Net, Google-Net, Reference Caffe-Net, and hence this was our model of choice. We feed all the frames generated from the input video one-by-one sequentially to the VGG-16 model. The output of VGG-16 is a 4096 dimensional feature vector corresponding to the input frame. Most of the high level features are captured by this vector, which is then given as an input the LSTM module.

ConvNet is made up of layers like convoultional layer, ReLU layer, pooling layer and fully connected layer. It converts a 3D image from its original pixel values to class scores [11, 12]. Variable size images from the MSCOCO dataset are resized to dimension 224x224x3(width, height and depth respectively). CONV layer computes the dot product of the input neurons with the region they are connected to. ReLU (Rectified Linear Unit) will apply an activation function. Down sampling is done using POOL layer. Lastly, class scores are computed by Fully Connected Layer (FC), resulting in a vector of dimension [1x1x4096]. In this way, layer by layer transformation takes place on a total of 138M number of parameters of this network.

IV.iii. Recurrent Neural Network

RNNs learn complex temporal dynamics by mapping input sequence to output state via hidden states [3]. LSTM is a type of Recurrent Neural Network with the following recurrence equations [5]:

$$h_t = f(W_{zh}x_t + W_{hh}h_{t-1})$$

$$z_t = g(W_{zh}h_t)$$

Where, z_t is the output sequence, (x_1 ...xt) is the input sequence and (h_1...h_t) is the hidden sequence.

The functions f and g is sigmoid or hyperbolic tangent. W_{ij} are the weights connecting two layers of neurons. One of the problems with RNN is that it requires fixed length.

IV.iv. Long Short Term Memory

Sometimes we need very less past information to perform the present task to predict the next word in the sentence "The color of the grass is ... *green*", we dont need any further context. RNNs can do this task easily. But eventually we need more contexts in real life scenarios. For e.g., "The boy is native of India...He is fluent in *Hindi*", to predict the last word in the above sentence we need more information from further

back. LSTMs help to store this long term information. They have a simple structure consisting of neural network layers interacting in a special way. Inspired from [17], LSTM architecture is shown below:

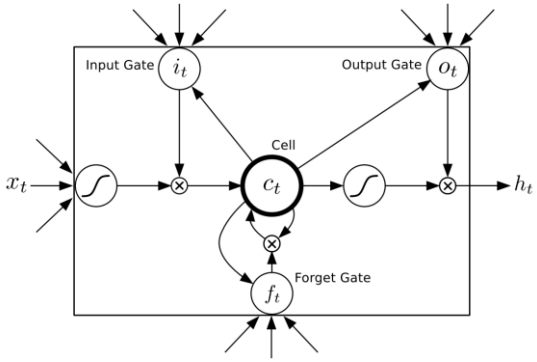

Fig. 3: Single LSTM cell (LSTM architecture)

The core unit of LSTM is a memory unit known as cell (c_t). It has three gates, namely input (i_t), output gate (O_t) and forget gate (f_t). It firstly decides which information to throw away from the cell state. This is done by forget gate. Forget gate analyses the input from input gate and xt and gives the output as 0(discard) or 1(save) for the cell state Ct-1. Next step is to decide the new information to store and update the old one in the cell state. Finally, the output will be based on a filtered version of the cell state.

To generate the description for the input video we need a sentence generation model. We decided to use LSTM which an advancement over Recurrent Neural Networks and has been found to be useful for sequencing tasks. Our LSTM model is 2 layered, with each layer consisting of 512 units. The 4096 dimensional feature vector is the input for the first layer of the LSTM model, the output of which is fed as an input to the second LSTM layer. This second layer outputs one word at a time to generate a description for the frame under consideration. We trained our LSTM model using Chainer framework in python.

IV.v. Experimental Analysis

For the evaluation measures in our experiment, we have performed both automatic as well as human evaluation for performance analysis.

IV.v.i. Human Evaluation

We performed a survey wherein our associates watched each video and described each scene distinctively. The sentences then obtained were evaluated on grammatical correctness. The results then obtained were compared with automatic evaluation by our system.

IV.v.ii. Automatic Evaluation

For this purpose, we tested three mode CNN models along with VGGNet. All the models were trained on the same training dataset. We used a different test dataset to compare the results.

VGG-16 is pretrained with ImageNet dataset and provides high accuracy. For high computation purposes, Nvidia GeForce GTX GPU is used to train LSTM.

Fig. 4: An image used for testing purpose.

Following is a comparison of the captions generated by various models for the above test image:

Sr. No.	CNN Model	Caption	Accuracy	Speed
1.	AlexNet	Swimming trunk, dumb bell, power drill, punch bag	Moderate	Fast
2.	GoogleNet	Vat, tub, laptop, notebook	High	Fast
3.	VGGNet	Notebook, laptop, turnstile, carton	High	Slow
4.	Reference CaffeNet	Lollipop, cellphone, groom, abaya	Least	Very Fast

Table 1. Relative comparison of the CNN models where GoggleNet and VGGNet outperform the rest of the models.

The table suggests that GoogleNet and VGGNet are most accurate for our test image, but VGGNet turned out to be accurate on a wider set of test images. On a contrary GoogleNet produced higher precision as compared to VGGNet over a short set of testing data. We have tested both the models with appropriate testing data set and came up with the conclusion that GoogleNet provides more than average results in short period of time, whereas If VGGNet is trained for more than 100 iterations of training data, one can get better results over the previous one. Table 2 shows the comparative accuracy of these two models based on the extent of testing dataset.

Sr. No.	CNN model	Data set size	Accuracy (In %)
1.	GoogleNet	Large	46.16
	Google-Net	Small	72.73
2.	VGGNet	Large	75.07
	VGGnet	Small	32.67

Table 2. Numeric precision of GoogleNet and VGGNet over data size.

IV.vi. Limitations

The training data available for video description is quite limited, which reduces the accuracy. Also the above experiments are performed by our model which has completed 30 iterations over training data. State of the art results are obtained after completing 100 or more iterations. Also, our LSTM model can only generate description over feature vector at a fixed resolution. This remains an open problem to be solved.

V. CONCLUSIONS

In this paper, we proposed a novel approach to video description. In advancement to related work, we construct descriptions using Deep Learning approach, where frames are first analysed sequentially and then converted to text sequentially. For this purpose we have used state of the art models like Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). Our experiments have proved that our approach generates better sentences than the previous results. The future scope includes integration of face recognition APIs to recognise people, landmarks, monuments, streets or important objects with their names for better service. Our system finds its applications in the fields such as human-robot interaction, automatic scene description instead of manual evaluation, classification of news videos or photos based on the topic, content based video or image search, etc. Also our system can be fruitful for blind people to guide them navigation by converting the generated text to audio. We will release our implementation as well as the models and generated sentences.

ACKNOWLEDGMENT

We would like to thank the following, Prof. S. S. Sonawane and Dr. Parag Kulkarni for their fruitful guidance. We also appreciate the inputs from Naga Sandeep Ramachandruni and T. Satoshi which helped us in implementing our system.

REFERENCES

[1] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, Tamara L. Berg, “BabyTalk: Understanding and Generating Simple Image Descriptions”, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 12, December 2013, pp. 2891-2903.

[2] Karpathy A., Li Fei-Fei, "Deep visual-semantic alignments for generating image descriptions”, in Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conferenc, pp. 3128-3137.

[3] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, Trevor Darrell, “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”, arXiv:1411.4389v3 [cs.CV] 17 Feb 2015.

[4] Yashaswi Verma, Ankush Gupta, Prashanth Mannem, C. V. Jawahar, “Generating Image Descriptions Using Semantic Similarities in the Output Space”, in 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops.

[5] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, Kate Saenko, “Translating Videos to Natural Language using Deep Recurrent Neural Networks” in Proceedings the 2015 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2015), pp. 1494-1504, Denver, Colorado, June 2015.

[6] Subhashini Venugopalan, Marcus Rohrbach, Trevor Darrell, Jeff Donahue, Kate Saenko, Raymond Mooney, ”Sequence to Sequence – Video to Text”.

[7] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Alan L. Yuille, “Explain Images with Multimodal Recurrent Neural Networks”, arXiv:1410.1090v1 [cs.CV] 4 Oct 2014.

[8] Ryan Kiros, Ruslan Salakhutdinov, Richard S. Zemel, “Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models”, arXiv: 1411.2539v1 [cs.LG] 10 Nov 2014.

[9] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, “Show and Tell: A Neural Image Caption Generator”, arXiv: 1411.4555v2 [cs.CV] 20 Apr 2015.

[10] Hao Fang, Li Deng, Margaret Mitchell, Saurabh Gupta, Piotr Dollar,John C. Platt, Forrest Iandola, Jianfeng Gao, C. Lawrence Zitnick, Rupesh K. Srivastava, Xiaodong He, Geoffrey Zweig, “From Captions to Visual Concepts and Back”, arXiv:1411.4952v3 [cs.CV] 14 Apr 2015.

[11] Xinlei Chen, C. Lawrence Zitnick, “Learning a Recurrent Visual Representation for Image Caption Generation”, arXiv: 1411.5654v1 [cs.CV] 20 Nov 2014.

[12] Junhua Mao, Wei Xu ,Yi Yang, Jiang Wang, Zhiheng Huang Alan Yuille, “Deep Captioning with Multimodal Recurrent Neural Networks (M -RNN)”, arXiv: 1412.6632v5 [cs.CV] 11 Jun 2015.

[13] Remi Lebret, Pedro O. Pinheiro, Ronan Collobert, “Phrase-based Image Captioning”, arXiv: 1502.03671v2 [cs.CL] 9 Apr 2015.

[14] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, Aaron Courville, “Describing Videos by Exploiting Temporal Structure”, arXiv:1502.08029v5 [stat.ML] 1 Oct 2015.

[15] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, Alan L. Yuille, “Learning like a Child:Fast Novel Visual

Concept Learning from Sentence Descriptions of Images", arXiv:1504.06692v2 [cs.CV] 2 Oct 2015.

[16] "Automatic Caption Generation for News Images", in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No.4, April 2013.

[17] Joe Yue-Hei Ng, Matthew Hausknecht, Audheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, George Toderici from University of Maryland, University of Texas at Austin, Google,Inc.,"Beyond Short Snippets: Deep Networks for Video Classification", arXiv:1503.08909v2 [cs.CV] 13 Apr 2015.