Captioning for Motion Detection for video surveillance Applications using Deep Learning

M. Nivedita, Asnath Victy Phamila Y, Harsh P.V

Abstract: ***Video surveillance has become a major tool in Security and has become sophisticated and fool-proof. Recent developments in the field of Image processing and object recognition have enabled us to integrate with the existing technology of surveillance. Today, the most popular type of video surveillance system is CCTV. But there are a lot of de-merits to this system which are mentioned in the below section. We are aiming at improving the existing technology by drastically increasing its reliability by using motion detection and image captioning to detect the moving object and alert the user by describing about that in the form of image captioning. We have developed motion detection using the build-in functions of OpenCV and Image processing and an Image captioning system using Neural Networks like Convolutional Neural Networks (CNN) and Recurrent Neural Network-Long Short-Term Memory (RNN- LSTM) etc. This generated caption is sent to the user for analysis of the situation.***

Index Terms: CNN, Image Captioning, LSTM, motion detection, Video Surveillance

I. INTRODUCTION

Vision is the sense of morality. It gives us the clear distinction between right and wrong. Humans are capable of understanding and analyzing images and its features. But the same image when shown to a computer is nothing more than a matrix with a bunch of numbers. Now the real challenge is for the computer is caption images when it detects motion in the video frame and analyzing the features of the frame at the time of change of motion has occurred. This data should be captured in different environmental and lighting conditions.

There are different techniques of motion detection. They are Gaussian mixture model-based foreground and background segmentation. Newer versions of OpenCV supports Bayesian based foreground and background segmentation. In this application we have use the standard deviation-based motion detection which will measure the stander deviation of the detected regions us calculated and is compared with a preset threshold value. When the SD is more than the threshold value, motion in the foreground is detected. Captioning is a process by which objects and features in a image is detected and put forward in form of a sentence.

Captioning detects regions of interest in an image and generates a small caption which best describes the object. When natural language description is included, dense captioning describes images with detail than sentence captioning as shown in Fig.1. The computer must learn to identify the different objects that are present that a frame. To achieve this, we use a model which breaks images by detecting objects and areas of interest. Hence, we use Convolutional neural network (CNN) to identify and recognize the features and objects in an image and use these to generate the captions. Generating paragraphs for images is very difficult. First the fine-grained images without any distortion have to be fed into the system and the system has to learn from the given set of images and extract the useful features from the test images. To achieve this, we use a large dataset of images (MS COCO) to train, test and validate the images. The MS COCO dataset consists of the 164K images of different categories with different background and foreground. These images are divided into train, test and validation classes. We allocate 75% images for training and 25% for testing and validation purposes.

Sentences:
1) A man is holding a white dog
2) A man is holding women
3) A man is walking holding a white dog
4) The women are walking
5) The women is walking on grass holding man's arm

Fig.1. Sample Image and the relevant captions

For many years, caption generation is done by translating the features directly and reordering them. Due to recent advances in Machine Leaning, we can generate the caption easily using RNN (Recurrent Neural Networks). Encoder-decoder LSTM is a deep learning library in Keras which is used for caption generation. It works by generating the next sentence based on the previously given inputs.

Generating paragraphs from images is a tedious task. Instead the images are broken into the semantic meaningful pieces by detecting the objects and the regions in the image and decomposing the sentences into the paragraphs. The rest of this paper is organized as follows: Sec.1.2 overviews related work in the area of image captioning and RNNs, in Sec.1.3 the system architecture is discussed and in Sec 1.4 implementation of the system is done and in Sec 1.5 deals with the results and discussion followed by conclusion in Sec 1.6.

II. RELATED WORK

In [1] they implemented a multi- model neural network which automatically learns to features from the images and generate the captions. The image is trained using COCO dataset where the dataset is split into test, train and validation sets. First the images are tokenized and kept. The images are then fed into a CNN network. A convolutional neural network can be used to create a dense feature vector. This dense vector, also called an embedding, can be used as feature input into other algorithms or networks. For an image caption model, this embedding becomes a dense representation of the image and will be used as the initial state of the LSTM. LSTM is a advanced recurrent neural network architecture with long term memory cell that is commonly used. The previous states information is captured and is used for the current prediction through the memory cell states. It consists of 3 components i.e. the forget gate, input gate and the output gate. The beam search algorithm used will focus on the outputs of most promising nodes. It generates the best fitting captions only. Hence, we get better suited captions instead of random captions. In [2] CNN and LSTM have been used to obtain the features and caption it. They have improved the speed of the network by retuning the hyper-parameters. They have used MS COCO dataset. First the dataset is fed into the CNN and then to LSTM to generate the captions. In [3] they use SIMNET (Stepwise image topic merging network) that uses two types of attention at the same time. According to the generated context, the decoder merges the information in the extracted topics and the image, so that the visual information and the semantic information can be effectively combined. In [4] Fully Convolutional Localization Network (FCLN) architecture is used to process an image in one single forward pass efficiently. Here we don't need any external regions. This architecture consists of CNN and RNN which is used to get the images features and image captioning respectively. The dataset used here Visual genome dataset which consists of 4100000 region grounded captions. In [5] the VDSA consist of both visual attention model and semantic attention model to extract both the image features and the semantic relationship between them. Here the most relevant word is aligned with the current word by which captions are generated for the image. The proposed models have been tested on Flickr30K and MS COCO datasets. In [6] a multimodal RNN is used for generating the captions. It works based on probability i.e. the best fitting word that can be fitted after the previous word. It consists of CNN-RNN architecture. They together form the M-RNN. They have tested the model on IAPR TC-12, Flickr 8K, Flickr 30K and MS COCO datasets. It has significant performance improvement over the others.

In [8] they have proposed a new model which consists of a policy and value network. They do not follow the conventional architecture of encoder-decoder. The policy network provides guidance in predicting the next word and the value network works as a look-ahead guidance by evaluating all possible extensions of the current state. Its focus is to predict words around the ground truth. A reinforcement type of training is performed on the MS COCO dataset. In [10] the image captioning system can caption any type of images i.e. people, objects, places, etc. it is capable of handing out of the domain data like captioning celebrities name even if the picture is not included in the dataset. It out performs other deep learning models which are trained on datasets like (MS COCO, Flickr30K), etc. the dataset is trained on MSCOCO dataset and images crawled from commercial search engines. It uses regular CNN architecture. For caption generation we use MELM (maximum entropy language model) with DMSM (deep multimodal similarity model). In [11] a model which uses the standard CNN-LSTM model for capturing the images features and captioning them is proposed. The MSCOCO dataset is first split into test, train and validation classes, and then it is sent to CNN for generating the image features and then it is passes to LSTM for sentence generation. This is also known as "Show and Tell image captioning". In [12] the model uses an end to end trainable bidirectional LSTM model for image captioning. This model consists of CNN and two bi-directional LSTMS. With the help of previous and future context information the captions are generated at high level semantic space. We prevent overfitting by using Data augmentation and this also helps us get more training data. The model is tested on datasets like Flickr8K, Flickr30K and MSCOCO. In [13] the authors proposed a two-stage procedure for training such an attribute-based approach: in the first stage, we mine several keywords from the training sentences and learn the mapping from images to those attributes with a CNN. In the second stage, we learn the mapping from detected attribute occurrence likelihoods to sentence description using LSTM. Datasets like Flickr8k, Flickr30K and MS COCO are used here. [14] proposed a model that is a modification to the already existing CNN-LSTM model. The CNN model is used to extract features from the image and the LSTM model is used to generate the caption. Here a question-answer type of model is implemented. This model asks questions to the images about its contents even though the image does not have a complete picture. By doing so, the system learns new features by answering the questions. In [15] the model uses joint interference and context fusion. They have used a R-CNN type model to identify the features and used the structure of VGG-16 for the convolutional layers. For generating the captions, we use the LSTM model. This model is tested on Visual Genome. In [16] they have introduced a new model known as LSTM-C which is LSTM with copying mechanism. This is done because training pairs with image and captions are very

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

limited.

It incorporates copying mechanism into CNN and RNN models. Word by word sentence generation given by the decoder RNN is integrated by LSTM-C with copying mechanism that will select words from novel objects in the output sentence. This model is trained with datasets like MS COCO, etc. In [17] they used a modified version of LSTM known as LSTM-A which is LSTM with attributes. Training is done in end to end manner in this image captioning framework and so that the attributes can be introduced to the CNN and RNN easily. Different variants of architecture are constructed to incorporate attributes into RNNS by feeding image representations in various ways to explore mutual and fuzzy relationship between them. They have been tested using the MS COCO dataset. In [18] they have proposed a new model which includes the top-down approach, bottom-up approach with sematic attention. The semantic concept region proposals will be learnt by the algorithm and fuse them into the hidden states and process and outputs the sentence from the Recurrent Neural Networks. The selection and fusion connect both the top-down and bottom-up approaches. This model has been tested on the MS COCO dataset. In [19] stride operations in deconvNet based visualizations was introduced to address the images. A dataset known as 8Flower which was specifically designed for objective quantitative evaluation of methods for visual explanation was used. The method used in this paper produced detailed explanations with extraction of relevant features of the classes of interest. It was experimented on the MNIST, ILSVRC12, Fashion144k and an 8Flower datasets. In [21] they have implemented the use of R-CNN which is suited for large scale visual learning. This model is end to end trainable. Learning long term dependencies is possible when non-linearities are introduced into the network. Hence, they have incorporated long term RNN models that can directly map video frames to the natural language text. They can be used as a model to complex temporal dynamics. The RNN which is directly connected to visual convolutional network models are trained to learn both temporal dynamics and convolutional perceptual representations. This model has been trained on MS COCO dataset. In [22] attention is calculated using the top down and bottom up mechanisms at object level. The bottom up approach uses a faster R-CNN to propose image regions and the top bottom approach is used to determine the feature weights. The dataset used here is MS COCO and they have received a BLEU score of 39.9.

III. PROPOSED SYSTEM ARCHITECTURE

A. *Motion Detection:*

Motion detection is the process of detecting a change in the position of an object relative to the environment. For motion detection we would require a camera as an external hardware. We get the live video from the external camera as shown in the Fig.2. The motion detector will find the change in difference between the two consecutive frames. When there is a difference, we can say that there is a motion in the foreground. But this motion can also be false, hence we find the standard deviation of the frame. This will trigger an alert when a motion of an object takes places in the foreground. This is doubly ensuring that there is no false motion detected by the system. First, we set up a minimum threshold value above which a motion has been detected. We calculate the Pythagorean distance between the three BGR layers of both the consecutive frames. This gives the difference between both the consecutive frames. Then we shift frames and apply Gaussian blur. This will reduce the image noise and detail. Hence it is easier to identify change when external noise has been removed. Then a binary mapping has been made to retrieve the location of motion in the frame. Then we calculate the standard deviation. Finally, we compare the calculated standard deviation with the threshold standard deviation that we had initially set. If the value is greater than the threshold, then we capture the frame and store it in a folder. If not then, the program will continue to run until a motion is detected.

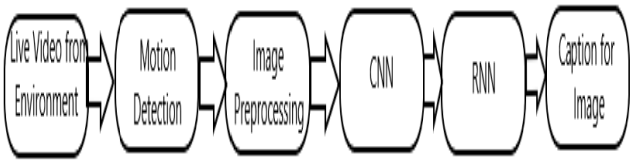

Fig.2. System Architecture

B. *Video Preprocessing:*

Video preprocessing was done in the previous module. We have applied Gaussian blur that reduces background blur. Here we do additional resizing of the images so that it is easier to train the images at the training phase. The images are shaped into a size of 224x224x3 using PIL libraries.

C. *CNN:*

For encoding the CNN is used. It is a type of neural network which will process the pixel data.
It has input layer, one or more hidden layers and output layer. The hidden layers include multiple layers like convolution, pooling, fully connected and normalization layers. The image features will automatically be extracted by the CNN and by doing transfer learning the features are fed to LSTM for decoding.

D. *LSTM:*

For decoding we have two options. One is Gated Recurrent unit and the other id long short-term memory. For this application we have used the latter one. The LSTM part helps in generating the captions from the 1024-dimensional output vector. LSTM consist of long- and short-term memory. LSTM consists of certain cells which perform certain operations to generate the captions. The cell under optimization must find the correct weights to accommodate the dictionary of words. We feed the words

as inputs and get output as a probability of words from the entire dictionary. LSTM predicts the next best fitting word from the set of previous words that it has predicted. For this application we will need a dictionary.

The problem is sequence models do not understand symbolic languages. Hence images have to be represented in the form of tensors. To present this we have to map the words to the integers. We obtain the embeddings of the integer representations using a built embedded layer in Pytorch. Every word is going to get embedded in a high dimensional real number space. They help us to examine the word or character manifold once it is mapped to a 2D space. We use the teacher-forcer algorithm for converging the RNN. It works in the following manner:

1. Tokenize the sentence with a <start> and an <end> tag
2. Feed a sentence to the LSTM cell
3. Find the word having the most probability for occurring
4. Feed the next token to the LSTM network
5. Loop this procedure until <end> tag has reached.

IV. SYSTEM IMPLEMENTATION

We are using the COCO (Common objects in context) dataset. COCO is a large-scale object detection, segmentation, and captioning dataset. Its dataset consists of everyday objects and natural images which helps the machine to identify images even under very low lighting and environment quality. COCO has several features. It consists of 330K images out of which 200K are labelled. It consists of 1.5 million object instances which are classified into 80 object classes and 91 stuff categories. We have 5 captions per images where LSTM can use the best suited caption for the given test input image. It consists of 250000 people with key points. The entire dataset is around 19GB large. For our purpose we have split the dataset into test, train and validation sets. The split is done as 20:80 respectively. COCO also offers API which assists in loading, parsing, and visualizing annotations in COCO.

We have used Pytorch to implement CNN and RNN. The torch visions- models package comes with a classifier of 1000 classes. We have replaced the classifier with a parametric RELU activation function and applied dropout to reduce overfitting. We have used a pretrained model instead of training the model as it helped in reducing the amount of computational power and resources. The result of the network will be a 1024- dimensional vector in the latent space. Then the LSTM takes that as input and generates meaningful captions by using beam search algorithm as shown in Fig.3. A beam search algorithm will efficiently select the top-N highest scoring responses from among the very large set of possible messages that a LSTM can generate. It uses breadth-first search to build its search tree, but only keeps top N (beam size) nodes at each level in memory. The next level will then be expanded from these N nodes.

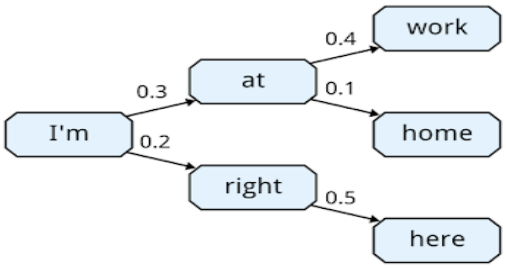

Fig.3. Example of a beam search tree

V. RESULTS AND DISCUSSIONS

The images shown in Fig.4.a) to d) are captured by a camera when any motion is detected. The detected frame is stored in a system in a specific location that is given as input image to the cautioner to generate the captions of the motion that is detected. The ground truth caption and the system generated captions are given below for all the four images. The caption generated is evaluated against the ground truth by using the BLEU score.

Generated Caption: <start> a man is holding a cell phone in front of a television . <end>

Base Truth: A man is holding a cellphone and laughing

a)

Generated Caption: <start> a person holding a pink umbrella in front of a building . <end>

Base Truth: A man is holding a blue umbrella

b)

Generated Caption: <start> a person is holding a laptop in front of a laptop . <end>

Base Truth: A man is holding a black laptop

c)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

d)

Fig.4. a) to d) Motion Detection Frames and Their Appropriate captions generated by the proposed system

VI. EVALUATION METRIC

A. *Blue score:*

It is also known as Bilingual Evaluation Understudy. It is a metric for comparing a candidate translation of text to one or more references translations. Its main use is to evaluate the correctness of the text generated by the system. It has a range from 0 to 1. It is 0 when the sentence generated by the system is totally irrelevant. It is 1 when the sentence generated by the system is exact to that of the base truth. The features of bleu score are:

1. It is quick and inexpensive
2. It is language independent
3. It is widely adopted

It works by counting matching n-grams in the computer-generated text with the n-grams in the reference text. The comparison is regardless of word order. It also takes in account of the similar words and not the amount of relevant words. In python, we import bleu score import *sentence_bleu* from the *nltk.translate* library. We pass the generated and reference text as parameters. This function returns the bleu score value that is shown in the TABLE I.

TABLE I: Images evaluated using BLEU Score

Image	Ground truth caption	Generated caption	BLEU caption
Fig.4.a)	A man is holding a cellphone and laughing.	A man is holding a cellphone in front of a television	0.51566
Fig.4.b)	A person is holding a blue umbrella	A person is holding a pink umbrella in front of a building	0.33180
Fig.4.c)	A man is holding a laptop	A man is holding a laptop in front of a laptop	0.46173
Fig.4.d)	A man is holding a remote in his hand	A man is holding a remote in his hand	1.0000

VII. CONCLUSION

We have made a model which captions images when any motion is detected in a live video stream. Our approach features quick and faster Real time image captioning. We have implemented the use of Motion detection, CNN and LSTM. The practical application for this model is endless. However, the major focus here is about smart video surveillance. This model can be the future of CCTV- based applications. From our BLEU score we have showed that our model's prediction is pretty good. We have got a score in the range 0.4 to 1. In future developments, we can integrate this application with a cloud-based framework where any suspicious activity in the live stream can be automatically detected by the system and informed to the police. This will help in reducing the number of crimes and make the lives of people safer.

REFERENCES

1. Zhongliang Yang, Yu-Jin Zhang, Sadaqat ur Rehman, Yongfeng Huang ,"Image Captioning with Object Detection and Localization" arXiv preprint arXiv:1706.02430 (2017)
2. Jyoti Aneja, Aditya Deshpande, and Alexander GSchwing.2018.Convolutional image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.5561–5570..
3. Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, and Tat-Seng Chua.2017. SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).6298–6306.
4. Justin Johnson, Andrej Karpathy, and Li Fei-Fei.2016. Dense cap: Fully convolutional localization networks for dense captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.4565–4574.
5. Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher.2017. Knowing when to look: Adaptive attention via A visual sentinel for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).3242–3250..
6. Junhua Mao,Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille.2015. Deep captioning with multimodal recurrent neural networks (m-rnn). In International Conference on Learning Representations (ICLR).
7. Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek.2017. Areas of Attention for Image Captioning. In Proceedings of the IEEE international conference on computer vision.1251–1259.
8. Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-JiaLi.2017. Deep Reinforcement Learning-based Image Captioning with Embedding Reward. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).1151–1159.
9. Hamed R Tavakoli, Rakshith Shetty, Ali Borji, and Jorma Laaksonen.2017.Paying Attention to Descriptions Generated by Image Captioning Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.2487– 2496.
10. Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, and Chris Sienkiewicz.2016. Rich image captioning in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.49–56.
11. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan.2017. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. IEEE transactions on pattern analysis and machine intelligence 39,4(2017), 652–663.
12. Cheng Wang, Haojin Yang, Christian Bartz, and

Christoph Meinel.2016. Image captioning with deep bidirectional LSTMs. In Proceedings of the 2016 ACM on Multimedia Conference.ACM,988–997.

13. QiWu, Chunhua Shen, Antonvanden Hengel, Lingqiao Liu, and Anthony Dick.2015. Image captioning with an intermediate attributes layer. arXiv preprint arXiv:1506.01144 (2015).
14. Linjie Yang, Kevin Tang, Jianchao Yang, and Li-JiaLi.2016. Dense Captioning with Joint Inference and Visual Context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).1978–1987.
15. Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2017. In corporating copying mechanism in image captioning for learningnovelobjects.In2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).IEEE,5263–5271.
16. Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei.2017. Boosting image captioning with attributes. In IEEE International Conference on Computer Vision (ICCV).4904–4912.
17. Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and JieboLuo.2016.Image captioning with semantic attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.4651–4659.

18. A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in IEEE Transactions on Pattern Analysis and Machine Intelligence, April 2017, pp. 664–676.
19. X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding long short term memory for image caption generation," in Proc. IEEE Int. Conf. Comp. Vis., 2015, pp. 2407–2415
20. J. Donahue, et al., "Long-term recurrent convolutional networks for visual recognition and description," in IEEE Transactions on Pattern Analysis and Machine Intelligence, April 2017, pp. 677–691.
21. Peter Anderson, Xiao dong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang.2017. Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 (2017).