Blind Mood System For Video Explanation Using Image Captioning

Mahmoud Gamal Aboelfotoh
201601022
s-mahmoud_gamal@zewailcity.edu.eg

Computer vision final project report
CIE 552

***Abstract*—Blind mod system is a system used for illustrating a video for blind people by using image captioning. Image captioning is mainly used for illustrating an image. In this project we tried to us this merit and extend it to be used for a video by using some video manipulation techniques.Blind mood system is divided into two parts : image captioning and video manipulation. Firstly , image captioning is used every scene change in the video and turn it into voice . Secondly , video manipulation is used for combining the voice generated by image captioning with the original video in a proper way. Image captioning have a lot of applications . blind mood . Image captioning , video manipulation.**

I. Introduction

Image captioning is about making a textual description of the content of an image. Image captioning has a various applications in various fields like : recommendation in editing systems , virtual assistant systems , etc. In this project we tried to make some virtual assistant system that help blind people to get the content of the video as easy as possible.

II. System Design :

A. Image Captioning Network : [1]

1) Network design:

The image captioning is a combination between two major fields in artificial intelligence : computer vision and natural language processing , so the image captioning network has two parts the encoder and the decoder:

The encoder : The encoder use a convolution neural network , usually a pretrained one we used resnet50, to extract features from the image. This feature map is transmitted to the decoder.

The decoder : The decoder is a recurrent neural network ,we used LSTM , receive the feature map from the encoder and do a language modelling up to the word level on it.

2) Training :

Data set: COCO data set.

The COCO data set has two parts : images and annotations,contain the captions. During the training , the image go through encoder and decoder , then the output of the decoder is compared with the caption of the input image from the annotations.Then , the output loss is used with some optimization techniques to develop our network.:

3) Testing : The testing part follow the same procedure of training as it use the COCO data set images and annotations to proceed in the testing part.

B. Video Manipulation :

1) Audio caption :

a) Get Prediction: Get image from a video and pass it through the trained image captioning network and get the caption as an output.

b) Get Text from image: In case there is any text in the image , this function will get it and add it to the caption. function (image_to_string) from (pytesseract) library. This library is an OCR tool for python which use LSTM as kind of RNN. [2]

c) From text to Audio: After adding the image caption and the text from image. We use (gtts) to convert text to audio.

2) Video creating :

a) Scene detection: At each major change in the scene , the function find scenes give the begin and the end of this change beside the frame number of that change. This information is used to cut the main video to sub videos. The algorithm of scene detection was mainly inspired from the function find scenes on scenedetect library website [3]

b) Video from audio and image : The audio from the Audio captioning function is combined with frame of change to make a video cation. he function (VideoFileClip) is used from the (moviepy) library to do this function. [4]

c) Video Composite: Mixing the channels of voice of audio of the main video with the audio of the caption. The function (CompositeVideoClip) is used from the (moviepy) library to do this function. [4]

d) Video Concatenate: This used for combine the sub videos to reconstruct the original video with the caption added to it. The function (concatenate_videoclips) is used from the (moviepy) library to do this function. [4]

III. Related work

a) The image captioning system was inspired from the image captioning project in the computer vision nano degree offered by Udacity and the github link for this project is: https://github.com/udacity/CVND---Image-Captioning-Project :

b) The scene change detection was inspired from find scene function in the documentation of PySceneDetect library. The link for this function is : https://pyscenedetect.readthedocs.io/projects/Manual/en/stable/api/scene_manager.html :

IV. Methodology

If you want to lead the algorithm, design it so Algorithm 1.

Algorithm 1 Video captioning algorithm

0: Get the frame list from the function find scene , this frame list is a csv that contain : frame rate , every frame number detected , its begin time and its end time.
0: **for** $i \leftarrow 1$ to $number\_of\_frames\_in\_the\_frame\_list$ **do**
0: Get *Frame_of_detection*
0: *Audio* = Audio_caption(Frame_of_detection)
0: *caption_video* = Video_caption (Audio , Fram_of_detection)
0: *sub_clip* = The_original_video [begin time : end time]
0: *video_composited* = CompositeVideoClip (caption_video , sub_clip)
0: *The_final_video* = concatenate_videoclips(The_final_video , video_composited)
0: **end for**=0

V. Experimental results

A. Image captioning results :

a) : At the first test for the whole system the image captioning wasn't giving any close idea about the image that was according to the small number epochs in the training.

b) : Then , after increasing the number of epochs , the results was much better giving a closer explanation which make more sense.But, that wasn't very promising.

c) : The final trick that appeared to has a major impact was transforming the input image with the same transformations used for the dataloader

B. Between video Compositing and video concatenating

a) The first though about reconstructing the video was by compositing the videos , this showed up a very disappointing results as it composite all the sub videos and captions in the same time. In another words , it converts a 30 seconds video to 4 seconds with mixing all the voices together. :

b) Then ,concatenating sub videos and clips together make the results much better , as the caption was said while the original video was freezed , then the video replayed again. However , this technique gives the first promising results and was showing how the final format of the video shall be , it wasn't meeting the expectation. Freezing the video makes the listener lose the mood of the video in some way and make the video much longer for example it converts a 30 seconds video to 70 seconds video:

c) Finally , combing the two techniques together was the last and the best trial. The video compositing was used in combing the caption video and the sub video related to this caption. Then, video concatenating was used to merge all the generated videos together. This result in a video almost in the same duration, not losing the video mood and sounds more familiar:

VI. Conclusion

Image captioning is very useful in revealing the context of an image , but in the video case the lack of following the context appears clearly. Of course it will give some insights to the listener about the content of the video , but it will not always give him a full understanding to the whole video.

REFERENCES

[1] Udacity, "udacity/cvnd—image-captioning-project," Jul 2018. [Online]. Available: https://github.com/udacity/CVND---Image-Captioning-Project

[2] "pytesseract." [Online]. Available: https://pypi.org/project/pytesseract/

[3] "Scenemanager." [Online]. Available: https://pyscenedetect.readthedocs.io/projects/Manual/en/stable/api/scene_manager.html

[4] "moviepy." [Online]. Available: https://zulko.github.io/moviepy/examples/examples.html