DCU ADAPT at TRECVID 2021: Video Summarization - Keeping It Simple

Anastasia Potyagalova and **Gareth J. F. Jones**
ADAPT Centre, School of Computing, Dublin City University, Dublin 9, Ireland
anastasia.potyagalova2@mail.dcu.ie
Gareth.Jones@dcu.ie

Abstract

We describe details of our participation in the TRECVID 2021 Video Summarization task. We adopt a straightforward approach to the task with the objective of understanding the potential for such a strategy to address the challenges of the task. The main strategy of our approach is based on detecting sub-clips containing selected characters using a neural network; the approach relies on face-detection algorithms and keyword search in short clips. Our solution to the Video Summarization subtask uses the same stages as the main task, with the only difference being that for each character's question had a separate pool of keywords which were searched for. Our results show that our method provides a reasonable solution to the main task, but is less successful for the subtask.

1 Introduction

We describe details of our participation in The TRECVID 2021 Video Summarization task [2]. This task aimed to foster research in the field of video summarization by asking participants to automatically summarize the major life events of selected characters over a number of weeks of programming of the BBC EastEnders TV series. Task participants were required to submit 4 summaries with 5, 10, 15 and 20 automatically selected shots for each of five different characters of the series. For our participation in the task we detected episodes containing the selected characters using the provided videos, scripts, master shot boundaries, and fan-made short videos. The generated summaries were evaluated by the TRECVID assessors according to their tempo, contextuality and redundancy, as well as with regard to how well they answered a set of questions unknown to the participants before submission. For our participation in the task, we adopted a straightforward technical approach, to examine its potential for addressing the task. We trained a face recognition system for BBC EastEnders characters, and used a single method for our runs with varying constraints on shots and the maximum summary duration.

In the next section we give details of our approach to the task, with results and conclusions in the following sections.

2 Approach

Our main approach to the video summarization task is based on detecting sub-clips containing selected characters using a neural network. Our methods rely on face-detection algorithms, and keyword search in short clips. Our solution to the subtask of the main video summarization task uses the same stages as the main task, with the only difference being that for each character's question had a separate pool of keywords, which were searched for. The first stage of our methods consists of data preparation. Short fan videos from YouTube and archived videos were used for image dataset preparation. We first cut frames with the presence of characters who are declared in the task, then, using OpenCV [7] methods, we sample only frames with faces. Then, using Keras [3] augmentation methods, several datasets were created with different augmentation options. The final dataset contains 4000 images for training for each class and 1000 images for validation and testing. In general, this method of creating datasets is quite simple to perform but in future, could be replaced by a method using Generative Adversarial Networks [5] to create more accurate augmented images by changing angles and lighting.

After completing pre-processing of the dataset, we began creating and training the neural network. This is a standard deep convolutional neural network, based on

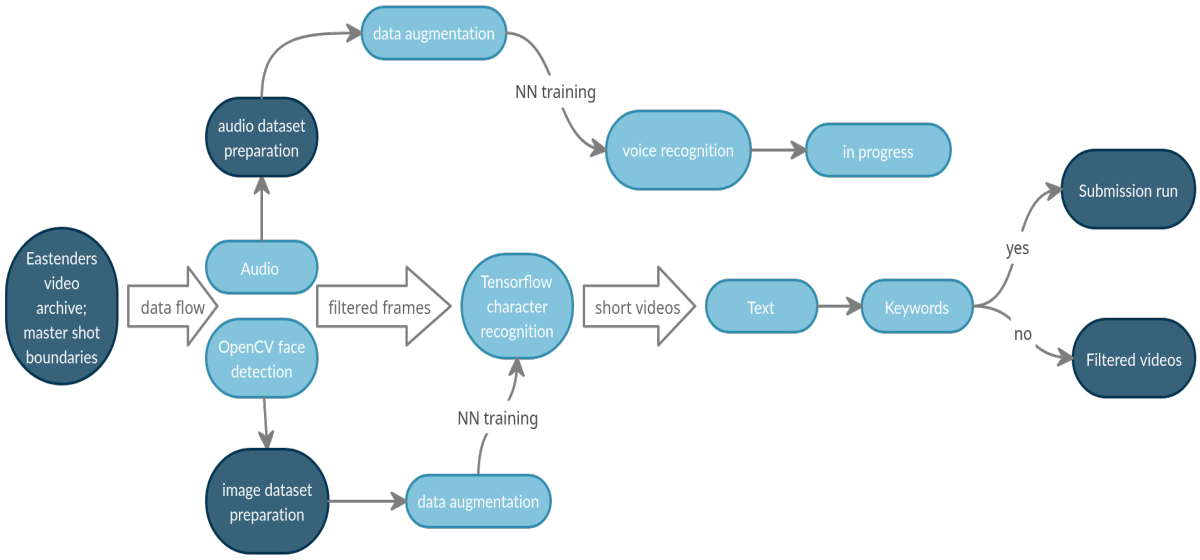

Figure 1: Phases of approach

the VGG16 principles.

Here we have started with initialising the model by specifying that the model is a sequential model. After initialising the model levels were added:

- 2 x convolution layer of 64 channel of 3x3 kernel
- 1 x maxpool layer of 2x2 pool size and stride 2x2
- 2 x convolution layer of 128 channel of 3x3 kernel
- 1 x maxpool layer of 2x2 pool size and stride 2x2
- 2 x convolution layer of 256 channel of 3x3 kernel
- 1 x maxpool layer of 2x2 pool size and stride 2x2
- 2 x convolution layer of 512 channel of 3x3 kernel
- 1 x maxpool layer of 2x2 pool size and stride 2x2

The next layers contain RELU (Rectified Linear Unit) activation to each layer so that all the negative values are not passed to the next layer. After creating all the convolution, we passed the data to the dense layer so for that we flattened the vector which comes out of the convolutions and added:

- 1 x Dense layer of 4096 units
- 1 x Dense layer of 4096 units

The Tensorflow API [1] was used for model development. After making the first models and checking them on the first videos, we observed a problem of overfitting. Regularization methods, such as kernel, bias and activity regulazier, from the Keras API were added to the dense layer to solve this problem. Selection of optimal settings for the neural network improved detection quality and selection of hyperparameters tuner from Keras API helped to reduce the time for neural network training. The next phase was to detect all faces in the frames using OpenCV methods to filter out those frames which do not contain characters. Next characters were detected in the filtered frames using a previously prepared Tensorflow model. As a result, after this stage, we had a set in which each episode consists of episodes containing instance of the selected characters.

The next stage of our algorithm consists of scraping synopses from video metadata and fansites. Our hypothesis was that if a character is not mentioned in the episode synopsis, there will be no important events for that character. This helps to filter out some episodes as irrelevant and reduce the time spent on the rest of the episodes. The mapping between the episodes and their dates is in files provided by the challenge organizers.

For this, an audio track was extracted from the clip, and speech in the audio was transcribed into text using DeepSpeech [4]. Also, an audio dataset for voice recognition was prepared based on extracted audio track. Then, using Librosa [6] augmentation methods, several datasets were created with different augmentation options. The final dataset contained 500 audio chunks for training for each class and 100 images for

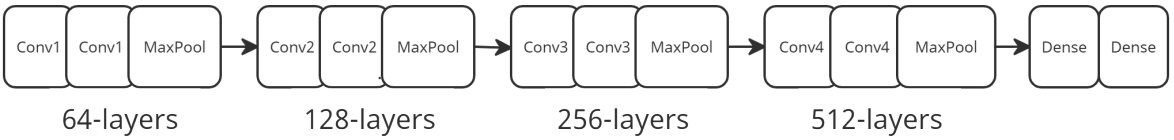

Figure 2: Neural network architecture

Query	Percentage
Archie_Run_1	62%
Archie_Run_2	79%
Archie_Run_3	30%
Archie_Run_4	31%
Jack_Run_1	17%
Jack_Run_2	16%
Jack_Run_3	30%
Jack_Run_4	14%
Max_Run_1	27%
Max_Run_2	8%
Max_Run_3	8%
Max_Run_4	10%
Peggy_Run_1	26%
Peggy_Run_2	26%
Peggy_Run_3	25%
Peggy_Run_4	42%
Tanya_Run_1	24%
Tanya_Run_2	43%
Tanya_Run_3	44%
Tanya_Run_4	42%

Table 1: Main task detailed results

Query	Percentage
Archie_Run_1	12%
Archie_Run_2	15%
Archie_Run_3	15%
Archie_Run_4	36%
Jack_Run_1	9%
Jack_Run_2	9%
Jack_Run_3	13%
Jack_Run_4	32%
Max_Run_1	12%
Max_Run_2	12%
Max_Run_3	12%
Max_Run_4	11%
Peggy_Run_1	12%
Peggy_Run_2	12%
Peggy_Run_3	12%
Peggy_Run_4	12%
Tanya_Run_1	33%
Tanya_Run_2	9%
Tanya_Run_3	33%
Tanya_Run_4	33%

Table 2: Sub task detailed results

testing. Further, specific keywords were searched for in the text file of each clip, which can serve as a flag to determine the importance of the episode. If the necessary words were found, this clip was marked as a key clip for the specified character and added to the final submission run.

3 Results

The average results for the team and for each selected character are presented in Table 1 (main task) and Table 2 (sub task). Table 1 contains results for the each run for each selected character for the main task and Table 2 contains results for the each run for each selected character for the sub task.

Unfortunately, our approach, using keywords search, is not very reliable for text analysis and it could not be used for detection of special events. Probably, it could be improved by using the bigger video dataset with selected characters, because it may provide more potential major video fragments for subtask questions.

4 Conclusions

Despite its straightforwardness, our approach gives a reasonably good result. To improve this result, a more accurate neural network could be used, probably with a more extensive training base using GAN algorithms or a more sophisticated structure. Also, accurate voice detection could be added for selected characters; our current voice recognition results with SincNet [8] tools were not accurate enough to include them in the final submission.

It may be beneficial to improve handling of questions related to individual characters and to perform a more detailed analysis of subscripts. Unfortunately, we found that the character-detection approach is not accurate enough for answering the subtask questions.

5 Acknowledgement

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224, and partially as part of the ADAPT Centre at DCU (Grant No. 13/RC/2106_P2) (www.adaptcentre.ie). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/

[2] G. Awad, A. A. Butt, K. Curtis, J. Fiscus, A. Godil, Y. Lee, A. Delgado, J. Zhang, E. Godard, B. Chocot, L. Diduch, J. Liu, Y. Graham, G. J. F. Jones, G. Quénot, Evaluating multiple video understanding and retrieval tasks at TRECVID 2021, in: Proceedings of TRECVID 2021, NIST, USA, 2021.

[3] F. Chollet, et al., Keras (2015).
URL https://github.com/fchollet/keras

[4] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, A. Y. Ng, Deep speech: Scaling up end-to-end speech recognition, cite arxiv:1412.5567 (2014).
URL http://arxiv.org/abs/1412.5567

[5] M. Luo, J. Cao, X. Ma, X. Zhang, R. He, Fa-gan: Face augmentation gan for deformation-invariant face recognition, IEEE Transactions on Information Forensics and Security 16 (2021) 2341–2355.

[6] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, A. Weiss, D. Hereñú, F.-R. Stöter, P. Friesch, M. Vollrath, T. Kim, Thassilo, librosa/librosa: 0.9.1 (Feb. 2022).
URL https://doi.org/10.5281/zenodo.6097378

[7] OpenCV, Open source computer vision library (2015).

[8] M. Ravanelli, T. Parcollet, Y. Bengio, The pytorch-kaldi speech recognition toolkit, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019, IEEE, 2019, pp. 6465–6469.
URL https://doi.org/10.1109/ICASSP.2019.8683713