Nagaoka University of Technology Submissions to TRECVID 2023 Video to Text Task

Mutsuki Ishii, Shungo Kubosaka, Takashi Yukawa

Nagaoka University of Technology, Niigata, Japan

Abstract

The kslab team participated in the VTT task of TRECVID 2023. Our system is composed of three phases: keyframe extraction, captioning, and caption aggregation. This year, we developed a new method for keyframe extraction and introduced a new phase that utilizes audio, built upon our traditional system. These components were incorporated into our previous system, resulting in the submission of four runs generated using different approaches.

The new keyframe extraction method uses boundary detection to remove noisy keyframes. In the previous method, the video was divided into short segments and the frames with large changes in RGB value features were extracted from each frame. The new method adds two filtering operations to the previous one to detect and remove frames that are on the boundaries of the video. The first operation is based on gray-scaled frame features, and the second filter takes into account the distribution of RGB values throughout the video. The resulting removal of blurred frames led to a reduction in errors during the captioning phase and improved scores on all assessments except BLEU. Therefore, this method was successful in improving captioning accuracy.

Furthermore, our new phase uses audio. Environmental sounds are extracted from audio data and determine the type of sound they represent. This phase verifies if the words included in the captions match the situations depicted in the video. Systems using this phase tended to have higher BLEU scores than those not using this phase but did not show improvements in other evaluation metrics. The analysis revealed that the improvement in BLEU scores resulted from the grammatical adjustments that were added within the phase. The environmental sound classification phase was not sufficiently effective in discriminating between audio types and did not enhance the meaning of the output text.

1. Introduction

The kslab team participated in the VTT task of TRECVID 2023. Our system consists of three phases: frame extraction from the video, captioning for each frame, and aggregation of the captions, as shown in Figure 1. In previous years, the method [1] that was proposed by Shibata et al. has been used for frame extraction. It uses only part of the frames, called keyframes [2], which are located at the start or end of a scene or transition. The OFA model [3] has been used for captioning, and Lexrank [4] has been used for aggregation. However, this frame extraction method sometimes selects blurred frames at scene changes, which leads to false object detection in the captioning phase. The conventional caption aggregation phase also has a problem in selecting captions that contain unnecessary words. To solve this problem, we propose a frame extraction method that combines two scene change detections and a new phase that uses audio. Therefore, this paper compares this year's system with the previous system and describes whether the new methods and phases contribute to achieving a more accurate depiction.

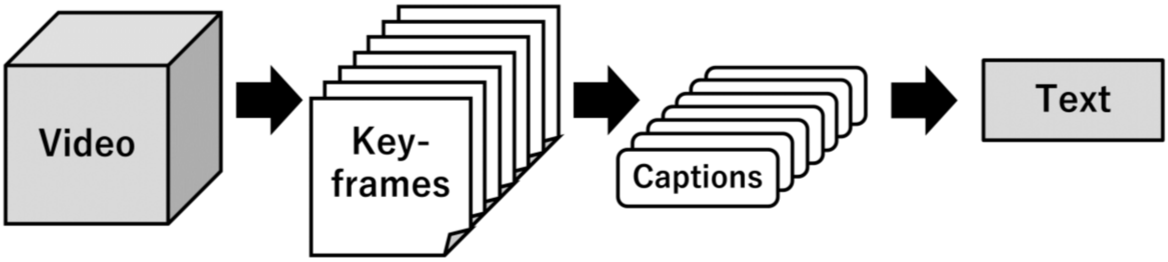


Figure 1. Overview of the keyframe method

2. Keyframe Extraction Method and Environmental Sound Classification Phase

2.1 Keyframe extraction method

In the keyframe extraction phase, conventionally, GoogLeNet [5] is used to extract the feature amount for each frame of the video. Kernel Temporal Segmentation (KTS) [6] is then used to extract seven images as keyframes by calculating the sum of five frames with large feature amounts extracted by GoogLeNet and including the first and last frames. The conventional system is shown in Figure 2. However, a problem with conventional systems is that frames during transitions, such as dissolves, might be extracted as keyframes, which could result in undesirable text generation. Therefore, as a solution to this problem, we propose a system that extracts frames during transitions, such as dissolves, and removes them from the keyframes.

Figure 3 illustrates the proposed keyframe extraction system, incorporating the consideration of frames during transitions, such as dissolves. In this method, keyframes are extracted using GoogLeNet and KTS as in the conventional method. After that, frames that are identified as dissolved scenes are removed from the extracted keyframes. During the keyframe extraction phase, it is necessary to ensure that an adequate number of keyframes are extracted, as some frames may be removed. To achieve this, we extract a total of 10 frames, including 8 frames with large features and 2 frames at the beginning and end, so that there are more keyframes than in conventional methods. Dissolved frames are removed from these keyframes.

A detection system for removing dissolved frames will be described. As a method for detecting dissolved frames, we used a method that combines (1) Ioka's "Detection of Dissolve Scene" method [7] and (2) Nagasaka and Tanaka's "Scene-Change Detection" method [8]. Method (1) for the detection of a dissolve scene is a technique that calculates and compares the amount of change in each pixel before and after each frame of a video converted to grayscale. Among these changes, the number of pixels exhibiting a positive change, such as increased brightness or intensity, when comparing consecutive frames is used as a feature to detect a dissolve scene. Nonetheless, this approach exhibited a limitation in its propensity to overemphasize feature detection when substantial motion occurred within the video or when zoom effects were introduced. On the other hand, the scene-change detection method of (2) divides each frame of the video into 4x4 blocks and calculates the feature by performing a chi-square test on the distribution of RGB values per block between consecutive frames. This scene-change detection method does not have the ability to detect dissolve scene, but it can determine whether the preceding and subsequent frame are from the same scene regardless of motion or zoom, based on changes in the RGB distribution across the entire video.Therefore, by combining it with method (1), we believed it would be possible to perform more accurate dissolve scene detection that is robust to movement and zooming, as demonstrated by the whole system shown in Figure 4.

In this system, videos are divided into 30 frames per second, and each frame is extracted using (1) "Detection of Dissolve Scene" method, which identifies frames with a large number of features. Next, the divided frames are analyzed for scene changes using (2) "Scene-Change Detection" method. Based on this result, a filter is created, and the results from (1) are applied to this filter. By extracting frames with values at least as high as the threshold from the obtained results, this system can detect dissolve frames.

We believe that this system will help remove the detected dissolve frames from the keyframes, resulting in more accurate caption generation.

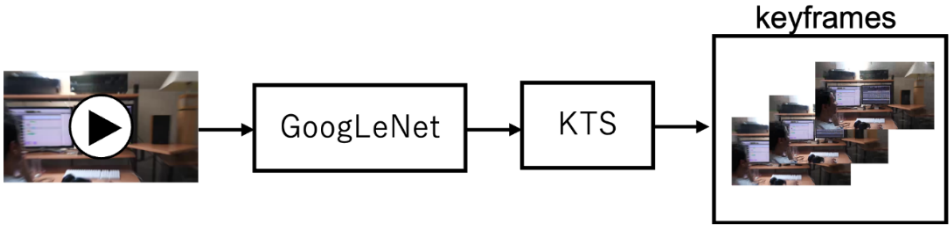


Figure 2. Previous keyframe extraction method

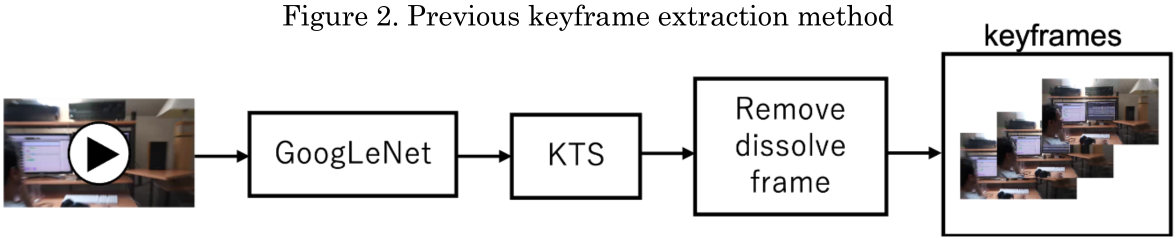


Figure 3. Proposed keyframe extraction method

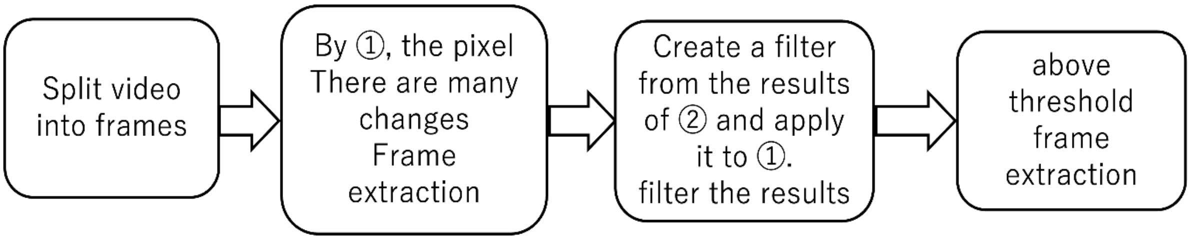

Split video into frames
By ①, the pixel There are many changes Frame extraction
Create a filter from the results of ② and apply it to ①. filter the results
above threshold frame extraction

Figure 4. Proposed Detection of dissolve scene Method

2.2 Environmental sound classification phase

In VTT tasks in TRECVID2018, the PicSOM team proposed using audio in a sentence generation system [9]. The system is based on the show and tell [10] model. Audio data is provided for the feature initialization and the persistent features in the LSTM layer. However, this system scored lower than the system that did not include audio. It is considered that the multi-labeled and large size dataset makes the audio classification difficult.

Therefore, we focused only on environmental sounds in the audio data and developed a new phase with sound classification techniques to determine the correctness of the captions. This phase refers to the environmental sound classification phase. Figure 5 shows the system process that added the environmental sound classification phase to the previous system.

The phase consists of three steps. First, it determines whether environmental sounds are included or excluded. Next, it classifies the environmental sounds. Finally, it calculates a similarity score between the label and the captions. After these three steps, the similarity score is added to the Lexrank score which is calculated in the aggregation phase, and the highest scored sentence is selected as the final sentence. Thus, the environmental audio classification phase has a role in helping to select a sentence that is the most relevant to the video content.

2.2.1 Detection of environmental sound

In this step, CNN is used as machine learning model to classify the audio.

The ESC-50 [11], VoxConverse [12] and free BGM are used as the training datasets.

Table 1 is a breakdown of the training dataset. The training data is processed by looping the audio to create a 16-second audio source. It is converted to a mel spectrogram image. We also prepare an equal amount of audio files with added white noise in order to increase the flexibility of the training. The CNN structure is shown in Figure 6.

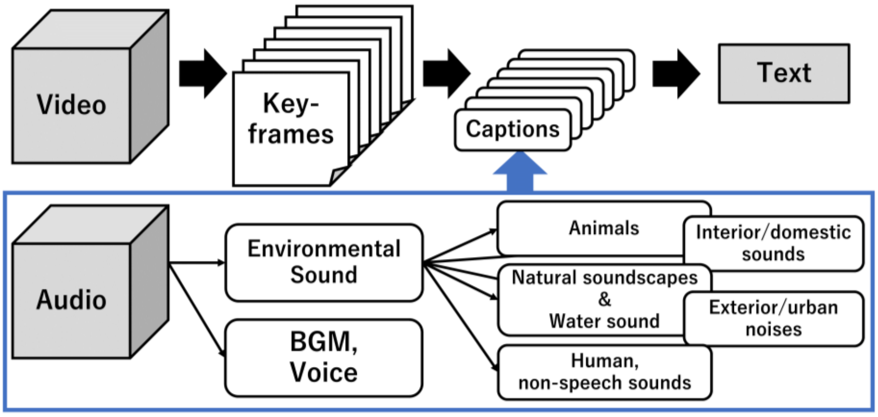


Figure 5. Overview of the system with audio phase

Table 1. Details of the dataset used for model training

Name	Type	Time	Sample
ESC-50	Environmental	5	2000
VoxConverse	Voice	5~15	2240
Free Music Archive	BGM	5~15	2068

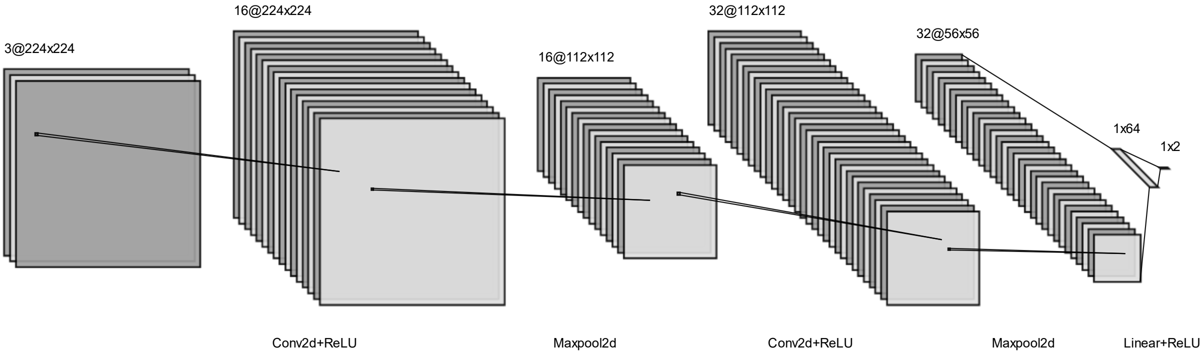


Figure 6. CNN for detecting environmental sound existence

2.2.2 Environmental sound classification

The environmental audio classification step also uses a CNN for as machine learning model. The training dataset is only ESC-50, and the audio processing is the same as in the environmental sound detection step. Figure 7 shows the CNN structure used in this step.

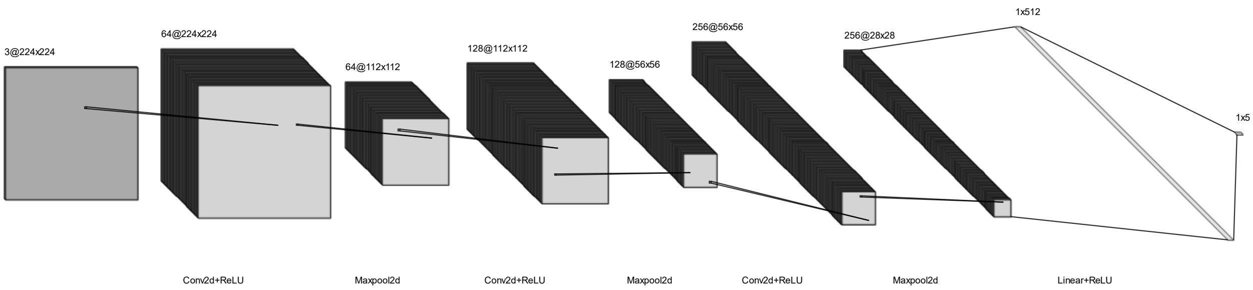


Figure 7. CNN for environmental sound classification

2.2.3 Calculating similarity score

Captions and labels from the environmental audio dataset are vectorized using sentence-transformers. After that, the cosine similarity is calculated, and the similarity is scaled in the range of 0 to 0.5. This scaled value is defined as the similarity score.

2.3 Submitted Runs

The kslab team submitted four runs in this year. Table 2 shows the combination of method and phase to create each run.

Table 2. Names and methods of runs

Run	Key frame extraction	Caption aggregation
TV23_NUT_1	KTS + Dissolve Detection	Text
TV23_NUT_2	KTS	Text
TV23_NUT_3	KTS + Dissolve Detection	Text + Audio
TV23_NUT_4	KTS	Text + Audio

Table 3. Scores for each run

	METEOR	BLEU	CIDEr	CIDEr-D	spice
tv23_NUT_1	0.2274255377	0.0384961812	0.501	0.140	0.078
tv23_NUT_2	0.2248083453	0.0392198399	0.484	0.130	0.076
tv23_NUT_3	0.2255115912	0.0539845496	0.495	0.139	0.077
tv23_NUT_4	0.2232341071	0.0548268463	0.479	0.130	0.076

3. Results and Discussion

Table 3 summarizes the run names and each evaluation metric score.

3.1 Keyframe extraction

tv23_NUT_1 and tv23_NUT_3, which included the proposed method, scored higher than tv23_NUT_2 and tv23_NUT_4 in the four evaluations, excluding BLEU. This confirms the effectiveness of removing frames in transition in the keyframe extraction phase. Since the score difference is small, we aim to enhance this phase's effectiveness. In contrast, there are cases where this method also removed frames that were important in the video. Current dissolve detection systems detect dissolves by comparing all of the pixels in a frame. In other words, if only a portion of the video frame has been edited, the dissolve scene cannot be detected. This has led to some videos scoring lower when the dissolves were removed. To achieve an improvement in scores, we would like to explore a system that can detect fine-grained editing points.

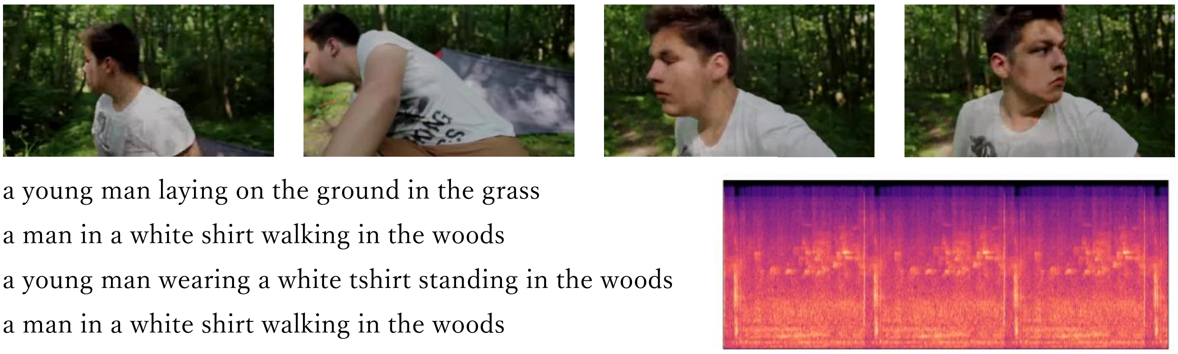
3.2 Sound classification

For running the test data, the video files were converted to wav format in order to work within the environmental sound classification phase. Each WAV file was processed into a 16-second looped audio and transformed into a mel spectrogram image for the same format as the training data. In addition, we adjusted the output text to align sentence beginnings and proper nouns with uppercase letters.

According to Table 2 and Table 3, tv23_NUT_3 and tv23_NUT_4 scored higher in BLEU than the other two runs that did not use the environmental audio classification phase in BLEU. We compared both pairs, tv23_NUT_1 and tv23_NUT_3, as well as tv23_NUT_2 and tv23_NUT_4, finding most of the words in the output were the same. This suggests that the grammatical adjustments were the reason for the increase in BLEU scores. In other cases, the environmental sound classification was very rarely found to be helpful in error handling.

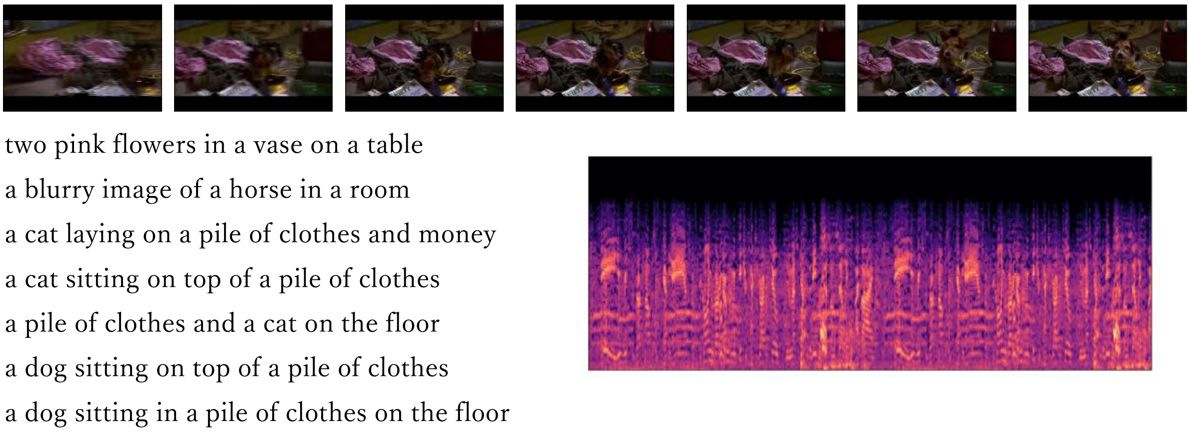
Figure 8 shows three outputs of video ID 484 from the environmental sound classification phase. In tv23_NUT_1, "a young man laying on the ground in the grass" was selected from the captions. On the other hand, in tv23_NUT_3, the caption "a man in a white shirt walking in the woods" was chosen. This is because the audio data for ID 484 was classified as "Natural soundscapes & water sound: Chirping birds," and the sentence "a man in a white shirt walking in the woods" was the closest meaning to the label. This leads to improved scores in all evaluation metrics for tv23_NUT_3.

In contrast, the most significant cause of poor performance in this phase is classification failure. In the example of video ID 1156 in Figure 9, a dog’s bark was misidentified as a cat. Therefore, the incorrect sentence "a cat sitting on top of a pile of clothes" was mistakenly selected instead of the correct caption "a dog sitting on top of a pile of clothes," which is also included in the list of generated captions. Most of the other videos with lower scores resulted from classification failures. One of the contributing factors to this issue appears to be the inadequacy of the dataset labels in encompassing the extensive range of sound categories present in the test data. The ESC-50 dataset comprises 50 labels; however, the test data contains sounds that cannot be classified into one of these labels. In such scenarios, it becomes imperative to assess the classification's reliability or contemplate reassigning the audio to a more limited set of labels, such as "human voice. "



a young man laying on the ground in the grass
a man in a white shirt walking in the woods
a young man wearing a white tshirt standing in the woods
a man in a white shirt walking in the woods

Figure 8. Video ID 484: keyframes, captions and mel spectrogram



two pink flowers in a vase on a table
a blurry image of a horse in a room
a cat laying on a pile of clothes and money
a cat sitting on top of a pile of clothes
a pile of clothes and a cat on the floor
a dog sitting on top of a pile of clothes
a dog sitting in a pile of clothes on the floor

Figure 9. Video ID 1156: keyframes, captions and mel spectrogram

4. Conclusion

The proposed keyframe extraction method is more effective than previous methods. However, it has been observed that when the dissolve effect is only applied for a part of the screen, there is a failure to remove dissolve frames during keyframe extraction. Regarding the environmental sound classification phase, it can be said that the system was not improved by the addition of it. We suspect that limiting the audio types and allowing for proper sound classification will be the key to effective use of this phase in the future.

References

[1] A. Shibata and T. Yukawa, An automatic text generation system for video clips using machine learning technique, In TRECVID 2017 VTT Task paper, Nagaoka University of Technology, 2018.

[2] G. F. Woodman and M. M. Chun, The role of working memory and long-term memory in visual search. Visual Cognition, Vol. 14, No. 4–8, pp. 808–830, 2006.

[3] T. Mashimo and T. Yukawa. Nagaoka University of Technology Submissions to TRECVID 2022 Video to Text Task, In TRECVID 2022 VTT Task paper, Nagaoka University of Technology, 2023.

[4] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, Vol. 22, No. 1, pp. 457–479, 2004.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[6] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, Category-specific video su mmarization, In Proceedings of European Conference on Computer Vision, Vol. 8694 o f Lecture Notes in Computer Science, pp. 540–555, 2014.

[7] M. Ioka, Detection of dissolve scene change in motion picture, In Proceedings of the 51st National Convention of IPSJ, no.6S-8, pp.247-248, Sept.1995

[8] A. Nagasaka, and Y. Tanaka, Automatic scene-change detection method for video works, In Proceedings of the 40th National Convention of IPSJ, no.1Q-5, pp.642-643, Mar.1990

[9] M. Sjoberg, H. R. Tavakoli, H. L. Mantecon, J. Laaksonen and Z. Xu, PicSOM Experiments in TRECVID 2018, In TRECVID 2018 VTT Task paper, Aalto University School of Electrical Engineering and Aalto University School of Science, 2018

[10] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, Show and tell: A neural image caption generator, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[11] K. J. Piczak, ESC: Dataset for Environmental Sound Classification, In Proceedings of the 23rd Annual ACM Conference on Multimedia, Brisbane, Australia, 2015

[12] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, A. Zisserman, Spot the conversation: speaker diarisation in the wild, Interspeech, 2020