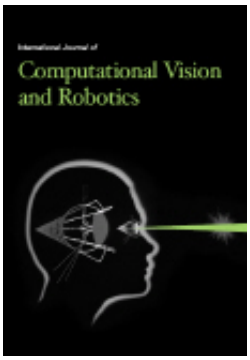


International Journal of Computational Vision and Robotics

ISSN online: 1752-914X - ISSN print: 1752-9131
https://www.inderscience.com/ijcvr

Construction of metadata of video for effective video search

Kitae Hwang, In Hwan Jung, Jae Moon Lee

DOI: 10.1504/IJCVR.2025.10067736

Article History:

Received:	21 October 2022
Last revised:	10 January 2023
Accepted:	23 February 2023
Published online:	02 December 2024

Construction of metadata of video for effective video search

Kitae Hwang, In Hwan Jung and Jae Moon Lee*

School of Computer Engineering,
Hansung University,
389, Samsundong, 2Ga, Sungbukgu, Seoul, South Korea
Email: calafk@hansung.ac.kr
Email: ihjung@hansung.ac.kr
Email: jmlee@hansung.ac.kr
*Corresponding author

Abstract: Video searching in the existing video sharing platform such as YouTube depends on hashtags or basic metadata generated manually when the video is produced. In such systems, the search accuracy is low because there is no detailed information about the video content. In this paper, we implemented an AI-based metadata construction system, the VMeta, which analyses audio and video frames in a video to extract feature data and stores them as 13 useful metadata to increase the accuracy of video search. In VMeta, when a video is uploaded using a web service, it analyses and creates metadata using TensorFlow. The VMeta system provides users with various meta-information about the searched video, such as presenter, keywords, categories, and ten-second text scripts, making it easy for users to make choices and provide users with a variety of information about the video without having to play it.

Keywords: video; search; metadata; ranking; scene; TensorFlow.

Reference to this paper should be made as follows: Hwang, K., Jung, I.H. and Lee, J.M. (2025) 'Construction of metadata of video for effective video search', *Int. J. Computational Vision and Robotics*, Vol. 15, No. 1, pp.19–30.

Biographical notes: Kitae Hwang is currently a Professor at the School of Computer Engineering in Hansung University, Seoul, Korea. He received his BE(CSE), ME(CSE) and PhD(CSE) in Seoul National University, Korea, in 1986, 1988, and 1994, respectively. He is having over 24 years of teaching and research experience. His research interest includes mobile computing, IoT system, and artificial intelligence.

In Hwan Jung received his PhD in Information and Communication Engineering from KAIST in 2000. He is currently a Professor in the Department of Computer Engineering at Hansung University, Seoul, South Korea. His major research interest is focused on topics related to IoT and mobile software.

Jae Moon Lee is a Professor at School of Computer Engineering at the Hansung University, Seoul in Republic of Korea. He received his MS in 1988 and PhD in 1992 from Korea Advanced Institute of Science and Technology (KAIST). His research interests include machine learning, artificial intelligence and computer game.

1 Introduction

With the development of the internet network and the generalisation of smartphones, mobile-centred media content consumption is increasing, and this type of content is also appearing in all fields of industries such as video lectures, TV broadcasting, movies, UCC, and music. Videos, lectures and advertisements (Törhönen et al., 2020; Gimpel, 2015). This trend is further spread by many video sharing platforms such as YouTube, Panopto, Brightcove, and Vimeo (Peer and Ksiazek, 2011). As video sharing platforms become active, more videos are being produced and distributed on the internet, and as e-learning spreads in universities and other educational institutions, more learning videos are being produced.

The increase in the number of videos adds difficulty in finding a video that a user wants. This is because a typical video search relies on very basic metadata such as hashtags or a short description attached to a video by the creator, the video title, the creator's name, or simple information such as subtitles. If there is not enough metadata, the user's ability to search for video narrows and thus the quality of the search deteriorates. Even if videos are recommended through a search, the user cannot determine what features the searched videos have or whether the video they are looking for is correct, so they play the video little by little to determine whether it is a desired video. This adds a lot of time wasted and inconvenience to users (Hanjalic et al., 2012). In this paper, we propose VMeta, a system that builds rich metadata for video to improve video search accuracy and provide users with various information about video. VMeta generates metadata by automatically extracting 13 pieces of information, such as keywords and categories that can represent the content of the video. To extract this information, the voice in the video is converted to text and analysed, and each frame of the video is analysed. When generating metadata, Index provides the video's voice as a text script every ten seconds.

In this paper, the feasibility was verified by implementing VMeta in the form of a web service. When an administrator uploads a video, the web application of the VMeta system analyses the uploaded video and creates metadata, and stores it in the database. A user was allowed to search for a video by four items such as video author, title, keyword, and category, and the VMeta system recommended videos with high matching ranking by comparing them with the metadata of each video using a video ranking algorithm. In addition, all metadata is provided to the user so that the user can view detailed information about the recommended video and easily reselect it.

In this paper, the generation time of metadata was measured to evaluate the performance of the VMeta system. The time taken to generate metadata from the video was measured at 2.7 seconds per MB.

2 Related works

Video search on the vast web uses a web robot-like search engine to find the desired video from a number of random videos on the internet. This is a form of search serviced by Google, Bing, DuckDuckGo, Yahoo, etc. They provide faster searches by an automated web spider that visits a website and makes a copy of it, and indexes it within a search engine (Thong et al., 2002).

In contrast, companies with video-sharing platforms search for a specific video in the multitude of videos stored on their platform. Traditional companies with shared platforms include YouTube, Dalimotion, MetaCafe, and Vimeo. Universities and many e-learning companies and institutions that provide educational video content are also among the types that use video sharing platforms. Video search in video sharing platforms often relies on basic and simple meta information such as title, description, and hash tags contained in video files (Choudhari and Bhalla, 2015; Dimitrova et al., 2002; Lawto et al., 2011; Silber-Varod and Geri, 2014; Gunjan et al., 2012). In this case, there is a limit to providing a ranking in the search results. This is because these metadata are relatively simple pieces of information, such as automatically pasted when creating a video or tagged by video creators to suggest video content and have fundamental limitations in detailed search. A form of video retrieval that does not depend on the basic meta information contained in the video has been continuously studied. A form of video retrieval that does not depend on the basic meta information contained in the video has been continuously studied (Alberti et al., 2009).

Adcock et al. (2010) proposed TalkMiner, which automatically generates metadata directly from lecture videos shot with a video camera (Cooper et al., 2010). TalkMiner, a technology proposed a long time ago, identifies presentation slides in lecture videos and builds a search index from them. It utilises optical character recognition (OCR) technology to isolate the text on each slide of the PPT and store the time information and text as an index. When a search request is received from a user, a video is found from the stored information, and a representative frame of the video, lecture date and title, lecture time, number of slides, etc. are provided from previously stored information. Today's lecture videos are often shot directly with video cameras, but most of them record computer screens and voices, which makes it difficult to apply TalkMiner's techniques to modern video.

The study of extracting metadata from video has been focused on analysing the captions or metadata manually embedded into the video (Pal et al., 2019; Hatirnaz et al., 2020; Xiang et al., 2021; Scherer et al., 2022). However, in contrast to this paper, these studies do not analyse the frame with AI techniques, so they do not create enough metadata.

3 Generation of metadata

3.1 Types of metadata

As shown in Table 1, the number of metadata generated by the VMeta system proposed in this paper is 13 and classified into seven groups.

- Basic information – basic information such as video title, video author, video size, etc.
- Keywords – a list of key words in the voice and image of the video.
- Categories – classification of the content of the video

- Classification of a video into a few adjacent fields among philosophy, religion, social sciences (politics/economics/social/education/law, transportation/administrative/military), nature, science, medicine, IT, games, media, sports, music, fashion, language, literature, history, geography, culinary, and architecture.
- Narrative type – according to the way the presenter speaks in the video, if the purpose is to deliver information, it is divided into description type, and if it is other speaking method, it is divided into application type.
- Presentation type – if the background of the video hardly changes and the content delivery method is a simple narrative method, it is classified as a static type, and if dynamic scenes such as today's weather or news dominate, it is classified as a dynamic type.
- Script text – converts all voices in the video to text and stores them in ten-second increments
- Index – it is text information that has been reduced to meaningful words through syntactic analysis after combining the texts detected from the images in the scene of the video and converting the voice into text. At this time, it is configured in units of ten seconds.

3.2 Metadata creation

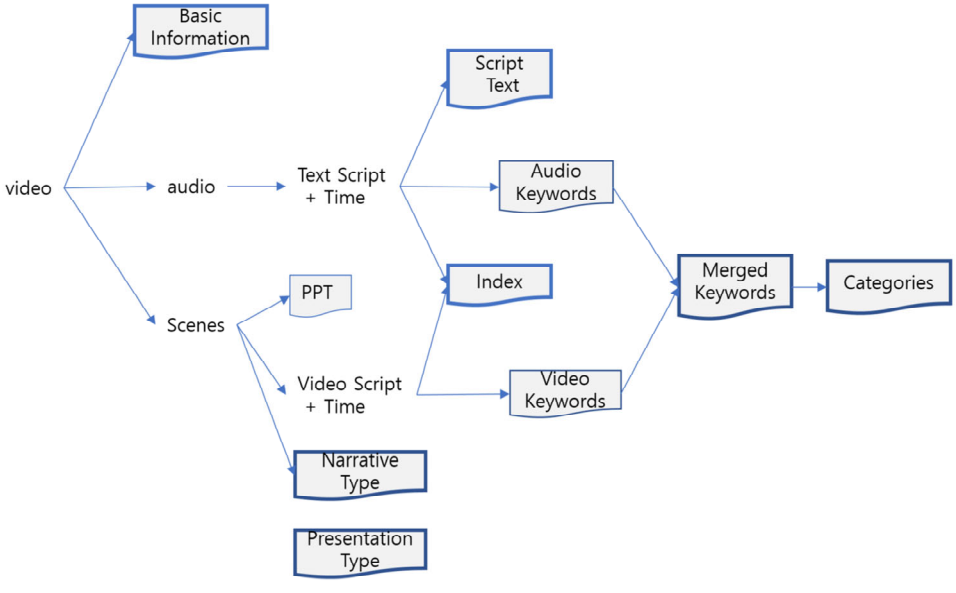
Figure 1 shows the entire process of generating metadata from video. The process of creating each piece of metadata is as follows.

First, read the video file and extract basic information from the video header. It then separates the audio stream from the video. Using the speech to text (STT) technique, the separated audio stream is converted into a text script, and script metadata is configured along with time information. Then, key words are selected from text scripts to compose index metadata in units of ten seconds. Then it finds the most used words in the text script and creates a list of audio keywords. It also analyses the text script to determine the category metadata. After separating the audio stream from the video, the video is split into multiple scenes. A scene is an image of a certain period of time that continues with the same subject or background in the video. For example, in a news video, when news about crime continues and then changes to weather news, the crime news part and the weather news part become scenes respectively. The process of segmenting scenes from a video stream is performed by analysing each frame of an image and extracting an image at a point in time when the frame changes rapidly. The extracted frame is judged as the time when the scene is changed, and the frame at that time is stored as an image and recorded in the database together with time information. In order to detect frame change, frame P is generated with the difference value between frame A and subsequent frames B, and if the signal-to-noise ratio (PSNR) of frame P is calculated and is greater than the set value, it is determined that the scene is switched. If it is judged as a scene change, frame B is saved as the starting point of a new scene.

Table 1 Metadata sample

Type	*Metadata*	*Sample*	*Note*
Basic information	Title	Chapter 4 thread	Video title
	Presenter	Hwang Kitae	
	Video length	00:45:03	
	Video frame	1280*720	
	Video type	MP4	
	Upload date	2022-03-15	
	Video size	80247350	Bytes
Category	Category	IT	
Narrative type	Narrative type	Description	Description/application
Presentation type	Presentation type	Static	Static/dynamic
Index	Index	06:32 Kernel-level thread	Time: Text script
		07:28 Thread library in a process-based operating system	
		10:33 User-level thread	
Keyword	Keyword	Thread; kernel; user; scheduling; library; operating system; process	
Script	Script	Let me talk about thread	Separated by enter key every ten seconds
		It is a run time unit in operating system.	

Figure 1 Metadata created from video (see online version for colours)



To create metadata of presentation type and narrative type, Keras' CNN classification model was used. Input images were classified into one of four labels: *L* (lecture), *N* (news), *P* (ppt), and *A* (application), and frames from various images were saved as image files and used as training data. About 10,000 images were used for training. The lecture format images were labelled *L* and news format images were labelled *N* and PPT-like images were labelled *P* and all other images labelled *A*.

If the probability of the *L* or *P* label, which is the result of classifying the image type by giving each scene image generated from the image as an input, is greater than 0.5, the image expression type is set to static, and otherwise, the image expression type is set to static. Otherwise, the video presentation type is determined dynamically. In the case of the highest probability of label *A*, the narrative type of the video was determined as application, and in other cases, it was determined as description.

Meanwhile, by using the OCR technique, characters in each scene are recognised based on the extracted scene frames, and an index composed of time and text is created. In addition, the index information generated from the audio stream is combined to create a single index metadata. For a user's later request, each scene image generated by the scene segmentation technique is connected to make one PPT.

3.3 Video ranking algorithm

3.3.1 Algorithm overview

The VMeta system provides users with four search options: title, presenter, keyword, and category of the video. The video ranking algorithm of this paper uses these four options as search parameters to recommend videos in the order of the highest match with the metadata and within the number range determined by the user. Recommended videos can be searched in detail according to category, presentation type, and narrative type.

Table 2 shows the four parameters and weight values used in the video ranking algorithm of this paper. p_i is the probability that the corresponding parameter will match the content of the video, which depends on the video. Therefore, for the entire video to be searched, the P_{vi} of video v has to be calculated. As shown in equation (1), the video ranking algorithm calculates R_v for video v using the parameter P_{vi} and weight W_i as shown in the following equation, and recommends a higher rank as the video R_v is larger.

$$R_v = \sum_{i=0}^{3} P_{vi} W_i \text{ for } \sum_{i=0}^{3} W_i = 1 \tag{1}$$

Table 2 Search parameters and weights used in the video ranking algorithm

Parameters	*Title*	*Presenter*	*Keyword*	*Category*
Match probability	P_0	P_1	P_2	P_3
Weight	$W_0 = 0.3$	$W_1 = 0.3$	$W_2 = 0.2$	$W_3 = 0.2$

3.3.2 Description of algorithm

Video Ranking algorithm consists of three steps.

- *Step1. Weight decision:* the initial values of the weight W_i of each search parameter are 0.3, 0.3, 0.2, and 0.2, respectively, as shown in Table 2. If a search parameter is omitted, all search parameter values are modified. When the search parameter is

omitted, the weight of the search parameter is equally divided among the weights of the remaining search parameters. If the nth search parameter is omitted, W_i is recalculated as equation (2).

$$W_i = \begin{matrix} W_i + \frac{W_n}{3}, \text{ for all } W_i (i \neq n) \\ 0 \text{ for } i = n \end{matrix} \tag{2}$$

The weight W_i of the search parameter is a value independent of video.

- *Step2. Determination of P_{vi} for each video v:* calculate values from P_{v0} to P_{v3} for each video v stored in the database. P_{vi} represents the probability that the i^{th} search parameter matches the video v. P_{v0} is 1 when the title parameter matches the title metadata of video v stored in the database, and 0 is determined otherwise. P_{v1} is also determined to be 1 when the presenter parameter matches the presenter metadata of video v, and 0 otherwise. The decision between P_{v2} and P_{v3} is not simple. Keyword metadata is stored in order of importance along with the importance of keywords in the video as shown in Table 3. The importance is expressed as a real number less than or equal to 1, and the importance of the most important keyword is 1. Table 3 shows a sample keyword metadata of one video. Here, the keyword 'operating system' is the most important keyword in the video and has an importance of 1, and the next 'virtual memory' keyword has an importance of 0.4.

Table 3 Keyword sample in metadata of a video

Index	*Keyword*	*Importance(M)*
0	Operating system	1
1	Virtual memory	0.4
2	Linux	0.3
…	…	…

Reflecting the importance value M when the search keyword is found in the metadata of video v, P_{v2} is calculated as equation (3).

$$P_{v2} = P_{v2} \times M \tag{3}$$

If the search keyword is not found in the keyword metadata of video v, P_{v2} is calculated as 0. In the case of category metadata, in the same way as keyword metadata, M values are stored for each field for 18 fields. P_{v3} is also calculated by the same formula as P_{v2} for each video.

- Step 3. *Video ranking calculation:* when P_{vi} for both W_i and video v are determined, using the parameter P_{vi} and the weight W_i, R_v representing the match for the search is calculated as the following equation (4).

$$R_v = \sum P_{vi} W_i, \text{ where } \sum W_i = 1 \tag{4}$$

It is determined that the higher the R_v, the higher the probability that the search parameter matches the video v.

4 Implementation and performance evaluation of VMeta system

4.1 Implementation of VMeta system

The VMeta system is implemented as a web service running on a server computer, and the overall structure and flow of control are shown in Figure 2. The web application was implemented using the Python-based Django framework, and the web server used NGINX. And the web server and Django framework were connected through Gunicorn web server gateway interface and Async server gateway interface.

Figure 2 Structure and control flow of the VMeta system (see online version for colours)

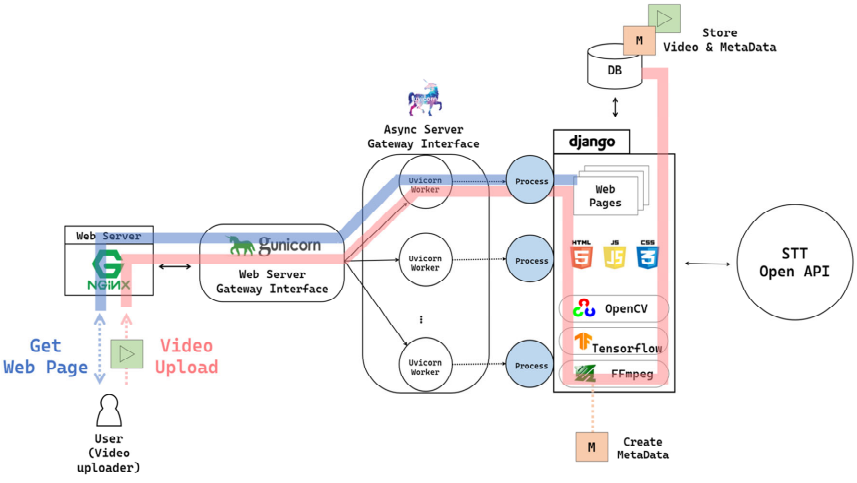
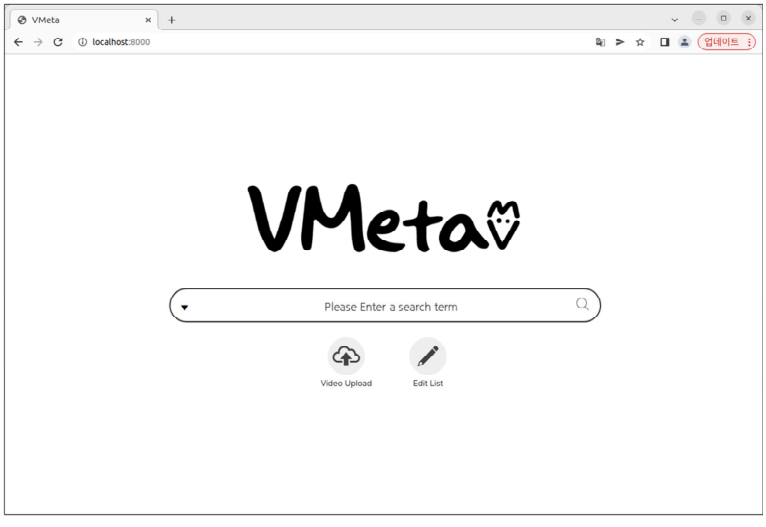
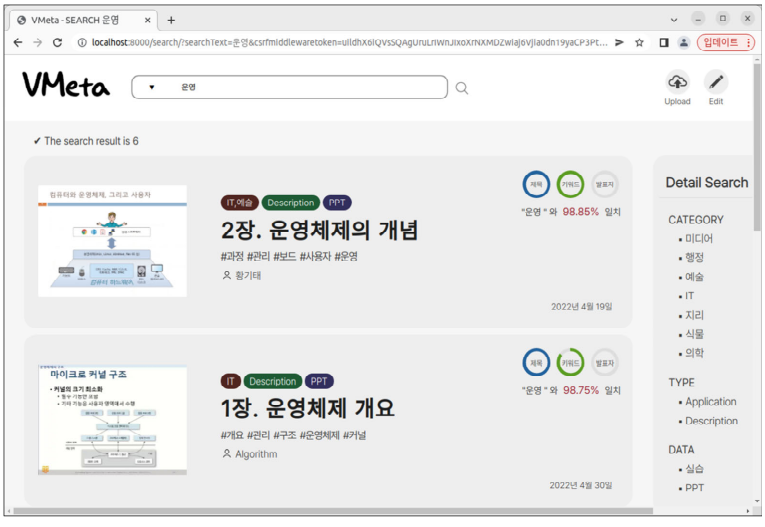


Figure 3 An example of a VMeta system in action, (a) upload screen (b) search results (c) metadata of the retrieved (see online version for colours)

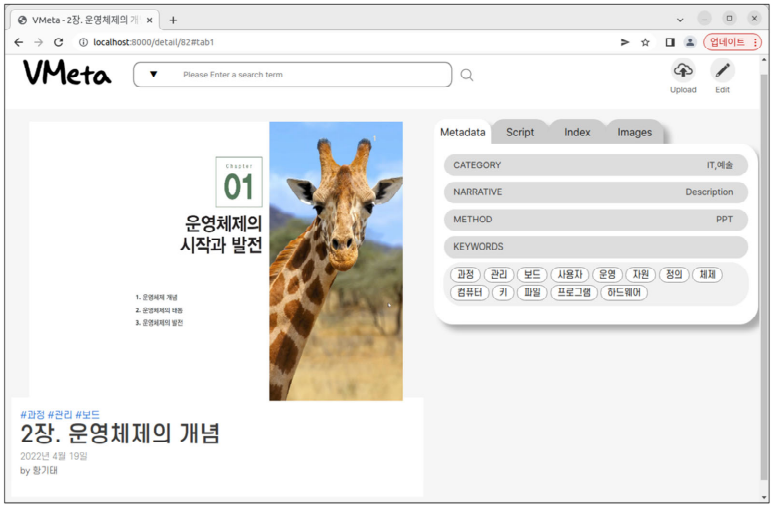


(a)

Figure 3 An example of a VMeta system in action, (a) upload screen (b) search results (c) metadata of the retrieved (continued) (see online version for colours)



(b)



(c)

A MySQL database was used to store video and metadata, and OpenCV was used for image processing, and a cloud service provided by ETRI (Korea Electronics and Telecommunications Research Institute) was used to convert audio into text. To generate metadata of presentation type and narrative type, images were learned and classified using artificial intelligence platforms Tensorflow and Keras. As training images, about 10,000 frames separated from the test video were used as images. Figure 3 shows the implementation example of the implemented VMeta system.

4.2 Performance evaluation

There are two major tasks in the VMeta system implemented in this paper. One is to analyse the video uploaded by the administrator to generate metadata, and the other is to search for the video by receiving a user's request. Therefore, the performance of the VMeta system is related to these two tasks.

4.2.1 Metadata creation time

The process of generating metadata from a video is long and complex and takes a lot of time. In this paper, the time taken to generate metadata for several video samples was measured and shown in Figure 4. As shown in Figure 4, depending on the size of the video, it can take anywhere from a few tens of seconds to a few minutes. Since the video size depends on various variables such as recording time, resolution, frame size, and video compression rate, the video size may not be proportional to the video recording time. Table 4 shows the recording time and frame size of the sample video used in Figure 4. As shown in this table, the recording time of sample 2 is shorter than that of sample 1, but the video size is larger than that of sample 1 because of the large frame size.

Since the time to generate metadata is proportional to the time to analyse the video, it can be predicted that it is proportional to the size of the video rather than the recording time of the video. Therefore, to evaluate the metadata generation performance, it is desirable to evaluate the average generation time of metadata per MB of video. As a result of measurement through experiments, the average metadata generation time was measured to be 2.675 seconds/MB.

Figure 4 Time taken to create metadata (see online version for colours)

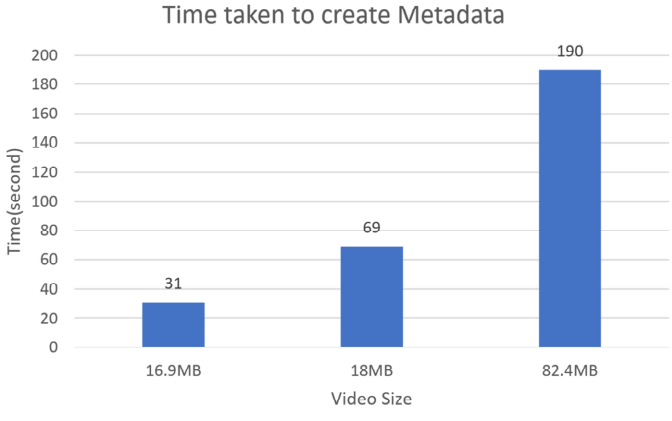


Table 4 Video samples

Video	*Sample 1*	*Sample 2*	*Sample 3*
Video size	16.9 MB	18 MB	82 MB
Video length	13:51	05:32	14:59
Frame size	962 × 720	1,280 × 720	2,208 × 1,668

4.2.2 Discussion

The video search time is generally proportional to the number of search parameters N and the number of stored videos M, and depends on the complexity of the video ranking algorithm. Video search in this paper is a simple task of comparing search parameters with previously constructed metadata, and since the number of parameters N is set to 4 and is small, N does not affect the search time. Figure 4 shows the results of measurements to determine how long the actual search takes. As the VMeta system is currently under experiment, it is not meaningful to measure the current search time because there are only about 100 saved videos. It is unfortunate that the search time cannot be measured in the context of large-scale video, but it is left for future research.

Search accuracy is a major indicator for evaluating search quality. If the video that the user wanted to search for is in the top ranking, it can be evaluated that the search accuracy is high. Therefore, it is accurate to evaluate the accuracy of the search by the satisfaction of the searched user. In addition to the method of asking the user directly, the user's satisfaction may evaluate the accuracy of the search by an indirect method of evaluating the time during which the user plays the video among the searched videos. Since the currently implemented VMeta system is not used in real situations, it is difficult to evaluate the accuracy of the search whether directly or indirectly. We plan to measure the accuracy of the search through future research.

5 Conclusions

In this paper, we proposed and implemented a VMeta system that generates metadata from videos in order to facilitate detailed search for users and provide users with rich data about videos. The metadata of the VMeta system consists of a total of 13 data including keywords, categories, and text scripts in units of ten seconds. The ranking algorithm in the VMeta system receives four parameters, such as keywords and categories, and recommends videos in the order of the highest match. Thus, users can easily understand the characteristics of a video without having to play it. As a result of measuring the metadata creation time of the VMeta system in this paper, it was estimated that it takes about 2.7 seconds per MB. Since the generation of metadata is performed in the background, this time is not significant.

Video retrieval time and retrieval accuracy are also important performance indicators in the VMeta system, but it is not easy to evaluate the retrieval time because VMeta is currently in the testing phase and there are not many loaded videos. In addition, since it is necessary to confirm the user's judgement on the accuracy of the search, their evaluation is left for future research.

Acknowledgements

This research was financially supported by the Hansung University.

References

Adcock, J., Cooper, M., Denoue, L., Pirsiavash, H. and Rowe, L.A. (2010) 'TalkMiner: a lecture webcast search engine', *Proceedings of the 18th ACM International Conference on Multimedia*, October, pp.241–250, https://doi.org/10.1145/1873951.1873986.

Alberti, C. et al. (2009) 'An audio indexing system for election video material', *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.4873–4876, DOI: 10.1109/ICASSP.2009.4960723.

Choudhari, K. and Bhalla, V.K. (2015) 'Video search engine optimization using keyword and feature analysis', *Procedia Computer Science*, Vol. 58, pp.691–697, ISSN: 1877-0509, https://doi.org/10.1016/j.procs.2015.08.089.

Cooper, M., Adcock, J., Denoue, L., Pirsiavash, H. and Rowe, L.A. (2010) 'TalkMiner: a search engine for online lecture video', *Proceedings of the 18th ACM International Conference on Multimedia*, October, pp.1507–1508, https://doi.org/10.1145/1873951.1874263.

Dimitrova, N., Zhang, H-J., Shahraray, B., Sezan, I., Huang, T. and Zakhor, A. (2002) 'Applications of video-content analysis and retrieval', in *IEEE MultiMedia*, July-Sept., Vol. 9, No. 3, pp.42–55, DOI: 10.1109/MMUL.2002.1022858.

Gimpel, G. (2015) 'The future of video platforms: key questions shaping the TV and video industry', *International Journal on Media Management*, Vol. 17, No. 1, pp.25–46, DOI: 10.1080/14241277.2015.1014039.

Gunjan, V.K., Pooja, Kumari, M., Kumar, A. and Rao, A.A. (2012) 'Search engine optimization with Google, *International Journal of Computer Science Issues*, January, Vol. 9, Nos. 1–3, ISSN (Online): 1694-0814.

Hanjalic, A., Kofler, C. and Larson, M. (2012) 'Intent and its discontents: the user at the wheel of the online video search engine' *Proceedings of the 20th ACM International Conference on Multimedia*, October, pp.1239–1248, https://doi.org/10.1145/2393347.2396424.

Hatirnaz, E., Sah, M. and Direkoglu, C. (2020) 'A novel framework and concept-based semantic search Interface for abnormal crowd behaviour analysis in surveillance videos', *Multimedia Tools and Applications*, Vol. 79, No. 25, pp.17579–17617.

Lawto, J., Gauvain, J-L., Lamel, L., Grefenstette, G., Gravier, G., Despres, J., Guinaudeau, C. and Sébillot, P. (2011) *A Scalable Video Search Engine Based on Audio Content Indexing and Topic Segmentation*, CoRR. abs/1111.6265.

Pal, S. et al. (2019) 'A semi-automatic metadata extraction model and method for video-based e-learning contents', *Education and Information Technologies*, Vol. 24, No. 6, pp.3243–3268.

Peer, L. and Ksiazek, T.B. (2011) 'Youtube and the challenge to journalism', *Journalism Studies*, Vol. 12, No. 1, pp.45–63, DOI: 10.1080/1461670X.2010.511951.

Scherer, J., Scherp, A. and Bhowmik, D. (2022) *Semantic Metadata Extraction from Dense Video Captioning*, arXiv preprint arXiv: 2211.02982.

Silber-Varod, V. and Geri, N. (2014) 'Can automatic speech recognition be satisficing for audio/video search? Keyword-focused analysis of Hebrew automatic and manual transcription', *Online Journal of Applied Knowledge Management*, A Publication of the International Institute for Applied Knowledge Management, Vol. 2, No. 1, pp.104–121.

Thong, J., Moreno, P., Logan, B., Fidler, B., Maffey, K. and Moores, M. (2002) 'Speechbot: an experimental speech-based search engine for multimedia content on the web', *IEEE Transactions on Multimedia*, Vol. 4, pp.88–96, DOI: 10.1109/6046.985557.

Törhönen, M., Sjöblom, M., Hassan, L. and Hamari, J. (2020) 'Fame and fortune, or just fun? A study on why people create content on video platforms', *Internet Research*, Vol. 30, No. 1, pp.165–190, https://doi.org/10.1108/INTR-06-2018-0270.

Xiang, Z. et al. (2021) 'Forensic analysis of video files using metadata', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.