IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 25, 2022, accepted June 6, 2022, date of publication June 17, 2022, date of current version June 23, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3184028

Fonts That Fit the Music: A Multimodal Design Trend Analysis of Lyric Videos

DAICHI HARAGUCHI[1], SHOTA SAKAGUCHI[2], JUN KATO[3], MASATAKA GOTO[3], AND SEIICHI UCHIDA[4], (Member, IEEE)

[1]Graduate School of Advanced Information Technology, Kyushu University, Fukuoka 819-0395, Japan
[2]Graduate School of Systems Life Sciences, Kyushu University, Fukuoka 819-0395, Japan
[3]National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan
[4]Department of Advanced Information Technology, Kyushu University, Fukuoka 819-0395, Japan

Corresponding author: Daichi Haraguchi (daichi.haraguchi@human.ait.kyushu-u.ac.jp)

This work was supported by the Japan Society for the Promotion Science (JSPS) KAKENHI under Grant JP17H06100.

ABSTRACT Lyric videos, or kinetic typography videos, are music videos showing lyric text in synchronization with the music. The purpose of this paper is to quantitatively and qualitatively analyze lyric videos to understand their design trends via three modalities: word motion, font style, and music style. These trends will not only be helpful as hints for designing new lyric videos but also be meaningful to quantitatively reveal the thought processes of the video design professionals. To achieve this, we needed to develop or utilize several technologies. First, we developed a lyric word tracking method to capture the motion of individual lyric words. The proposed method uses the lyric text as the guiding information for word tracking to overcome the difficulties arising from the various word appearances and motions. Second, we developed a font style estimator to quantify the appearance of each word as a feature vector. Finally, we employed a music style estimator to quantify the mood of the music, e.g., ''techno'' and ''fast.'' We then analyzed feature vectors of these three style modalities collected at 3,494 time points in 100 lyric videos. After revealing the trend of each modality via k-means, we conducted a co-occurrence analysis to understand the correlation between each modality pair. Our experimental results indicate that such a cluster-wise co-occurrence analysis can capture interesting trends hidden in lyric video designs.

INDEX TERMS Lyric video, lyric word tracking, text motion analysis, video design analysis.

I. INTRODUCTION

Lyric videos (a.k.a., kinetic typography videos) have become a popular approach for promoting songs on video sharing services, such as YouTube and social network services. In lyric videos, the lyric words are displayed and animated synchronously with the music. The display style of the lyric words is very different from that of still video captions. Figure 1 shows a series of video frames taken from a lyric video. In this video, the lyric words are shown in a decorative font style and move dynamically along with the video frames.

Similar to conventional typographic designs, such as book covers, posters, and web advertisements, creating lyric videos requires that the video creator have expertise in graphic design and that the relationship between the graphical and musical expressions be considered. The creators need to carefully choose the font style for the lyric words while considering the style (mood) of the music. Moreover, creators need to design the word motions. For example, lyric words might be shown with fewer motions for quieter music and with more flashy movements for energetic music. Moreover, the motions are often designed to be synchronized with the rhythm (i.e., the beat) and the vocal timing.

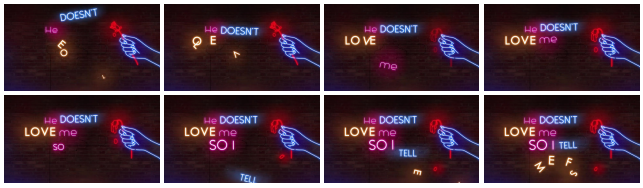


FIGURE 1. Example of video frames captured from an existing lyric video (from upper-left to lower-right).

This characteristic means that lyric words are often displayed in various decorated fonts. Therefore, elaborate visual designs are sometimes hard to read, even for humans. In addition, the background images of the video frames can be photographic images, illustrations, or mixtures of the two, often making it difficult to read the lyrics. As noted later, this means that lyric word detection and recognition for lyric

The associate editor coordinating the review of this manuscript and approving it for publication was Eunil Park.

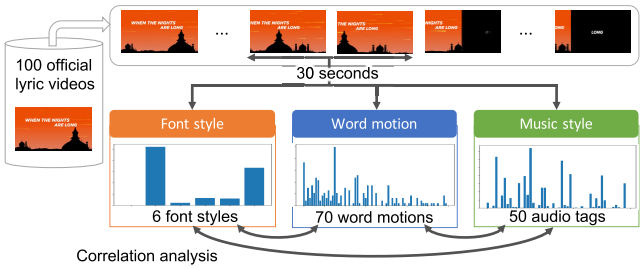


FIGURE 2. **Overview of the proposed lyric video analysis.**

videos is a difficult task, even for state-of-the-art scene text detectors and recognizers.

The purpose of this study is to explore the relationships between three style modalities of lyric videos: font style, word motion, and music style, as shown in Fig. 2. For this study, we need to develop or employ appropriate techniques to quantify these three style modalities. For example, to quantify word motions, we first need to detect and recognize lyric words in individual frames and then track them over multiple frames.

After quantifying the three style modalities, statistical analyses are conducted to reveal the correlations between the modalities. This correlation analysis is meaningful in two ways. First, it will lead to a deeper understanding of the typographic designs of lyric videos. This analysis provides hints as to how experts can use their knowledge of typography in music videos. Second, the relationships discovered by the analysis will help non-experts create lyric videos or help in the development of lyric video creation tools such as TextAlive [1]. The relationships could also be used to suggest suitable font styles for specific music styles.

Despite its meaningfulness, correlation analyses between these three style modalities for lyric videos are underexplored and remain challenging because of the following difficulties.

1) Word motion quantification is not a simple task. Lyric word detection and recognition for lyric videos is a difficult task, even for state-of-the-art scene text detectors and recognizers. Various decorated fonts and background images prevent the accurate detection and recognition of lyric words.
2) Even though quantification of the music style is possible using a standard style estimator, such as musicnn [2], [3], there is no standard tool for quantifying the font style. The font style in lyric videos has wide varieties, and therefore the employed font style estimator needs to be capable of dealing with them.
3) The correlation between the style modalities will likely be very subtle and weak. Styles largely depend on the designer's subjective choices and may undergo multiple artistic and artificial variations. For example, the same font style may be used for music with completely different styles. This indicates that style correlations will not have simple or clear (such as linear) trends or distinctive peaks. In fact, our preliminary regression analysis experiment using XGBoost [4] was unable to capture a clear correlation between the style modalities.
4) Because lyric videos are a relatively new multi-media video resource with typographic artwork, there is not yet a standard video dataset available for analyses. This situation is very different from other well-studied video analysis tasks, such as the Text REtrieval Conference Video Retrieval Evaluation (TRECVID).

To address the first of the above difficulties, we propose a lyric word detection and tracking method, called *lyric-frame matching*. Its key idea is to utilize the lyric word sequence, which is given as metadata, to improve the tracking performance. More specifically, state-of-the-art scene text detectors and recognizers are first applied to each video frame to obtain candidates for the lyric word locations. Then, dynamic programming (DP)-based optimization is applied to determine the optimal matching between the candidates and the lyric word sequences over the frames. The matching result gives a reliable spatio-temporal trajectory for each lyric word in a given sequence.

For the second difficulty, we developed a font style estimator based on a convolutional neural network (CNN). Basically, the estimator is simply realized by training the CNN with a font image dataset where the font style (e.g., "Sans-Serif") is annotated to each font. Because there is no standard font style class definition, we roughly defined six font styles and represented the style of a given word image using a six-dimensional class-probability vector. In addition, because the word images extracted from the lyric video frames have various backgrounds and distortions, we needed to train the CNN not with a clean font image but with synthetic font images that mimic actual lyric word images.

For the third difficulty, we made full use of cluster analysis. Even though clustering is a classic and simple method, it is useful for our correlation analysis task. Clustering involves vector quantization and therefore gives a rough view of the variations in the styles. Moreover, clustering can deal with highly nonlinear style trends because of its non-parametric nature. In this paper, we first apply k-means clustering to each modality independently and then apply a biclustering technique to understand the correlation between two modalities via the co-occurrence of their (quantized) styles.

For the fourth difficulty, we prepared a new lyric video dataset containing 100 lyric videos created by design experts. We manually attached the lyric word bounding boxes to 1,000 video frames to evaluate the accuracy of the lyric word tracking result. A list of the videos and the bounding box data are publicly available at `https://github.com/uchidalab/Lyric-Video`.

The main contributions of this paper can be summarized as follows:

- To the best of the authors' knowledge, this is the first study to analyze the design of lyric videos in a quantitative manner. Because of the design factors specific to lyric videos, we focus on three style modalities: font style, word motion, and music style.

A correlation analysis between these style modalities will provide basic knowledge concerning kinetic typography designs in music videos. In fact, the analysis results reveal interesting trends between the three style modalities; for example, ''Fancy'' fonts tend to be used for ''pop'' and ''guitar'' music, and active motions are often printed in ''Fancy'' and ''Sans-Serif'' fonts.

- This is also the first attempt to detect and then track lyric words in lyric videos. We propose a novel word tracking technique using an optimal lyric-frame matching algorithm based on DP.

A preliminary version of this study, in which only a word motion analysis was conducted, was published in a conference paper [5]. The present paper contains much wider analyses introducing two new style modalities, i.e., font style and music style. The correlation analysis between the three modalities is another novel contribution of this paper.

II. RELATED WORK

Since this paper is the first attempt at a design analysis of lyric videos, there are presently no similar studies. In this section, instead, we review previous attempts to extract or analyze word motion, font style, and music style for more general subjects.

A. WORD MOTION ANALYSIS

There are several tasks involved in detecting and tracking words in video frames. The most typical task is caption detection [6]–[14]. Captions are defined as text superimposed on video frames. Captions, therefore, have characteristics that differ from scene text. Even though most studies have dealt with static captions (i.e., captions without motions), Zedan [11] addressed not only static captions but also moving captions. They referred to the vertical or horizontal scrolling of caption text as moving captions.

Recently, video text tracking [15]–[23] has also been attempted, as reviewed in [24]. Because such methods try to track words in a scene captured by a moving camera, they introduce a common assumption that the words are static in the scene and are captured by the moving camera. Therefore, they assume, for example, that neighboring words will move in similar directions. The paper [25] introduces ''moving MNIST'' for video prediction tasks. This paper focuses on synthetic videos capturing two digits moving with respect to a uniform background.

Our study is very different from these previous attempts with respect to the following three points at least. First, our target words in lyric videos move far more dynamically and freely, invalidating the assumption used in previous studies. Second, we can utilize lyric information during tracking, whereas previous attempts did not include such guiding information.

B. FONT STYLE ESTIMATION

Most previous font image analysis studies have focused on the so-called font identification (or font recognition). This involves identifying the font name (such as ''Helvetica'') of a given text image. Zramdini and Ingold [26] presented a pioneering trial recognizing 10 different fonts. Recently, deep neural networks have also been used for font identification [27]–[29].

In this study, we use font style estimation, which is different from font identification. Font styles are defined as Serif, Sans-Serif, Script, and so on. If a method can estimate the style of an arbitrary font, it can be applied to lyric words printed with rare or even brand-new fonts. However, font style estimation is less common than font identification because font style classes are not well defined.[1] Shinahara *et al.* [31] developed a font style estimation method based on six font classes (Serif, Sans-Serif, Hybrid, Script, Historical Script, and Fancy) defined in a font guidebook [32]. They used simple pattern matching for the classification. In this paper, as described in Section V, we develop our own neural network-based font style estimator, following the same six classes. Note that there have been several classical attempts (see Table 1 of [33]) to classify font images into Roman, bold, and italic classes. We do not use these three classes because they are appropriate for font images from ordinary text documents but not for various font images of lyric words.

C. MUSIC STYLE ESTIMATION

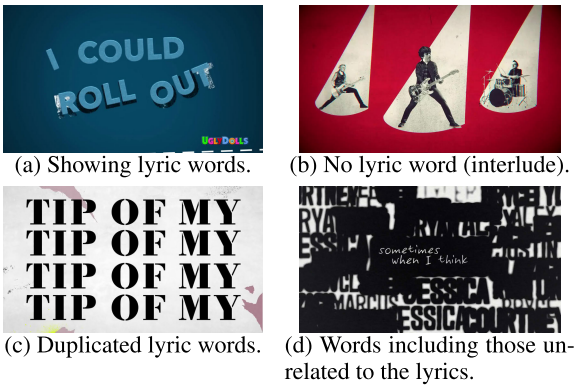
Music audio tagging, including mood/emotion estimation, is a popular research topic in the music information retrieval (MIR) community. Various approaches have already been proposed for music audio tagging [3], [34]–[43]. Recently, Pons and Serra released *musicnn* [2], [3], which can provide a ''taggram'' for each music segment using CNNs. Each taggram is a 50-dimensional vector, and each element corresponds to 1 of 50 tags (defined in the MagnaTagATune (MTT) dataset [44]). This is not a one-hot vector but rather a non-negative real-valued vector. Each value represents the property of the music segment or the corresponding tag.

In this paper, we use the 50-dimensional taggram given by musicnn as the music style. As in the case of the font styles, the music styles do not have any standard definition; this is because music styles are defined by multiple factors, such as instrument types and genres. Fortunately, taggram by musicnn covers these factors. Of the 50 tags, some tags indicate musical instruments (such as ''drums'' and ''guitar''), some indicate vocal types (such as ''male vocal'' and ''choral''), some indicate music genres (such as ''rock'' and ''techno''), and some indicate moods (such as ''loud'' and ''slow'').

III. LYRIC VIDEO DATASET

As the lyric video dataset to be analyzed, 100 videos were collected via the following steps. First, a list of lyric videos was generated by searching YouTube with the keywords ''official

[1]The PANOSE System [30] was expected to be a good standard for font styles; however, most fonts currently do not follow it.



(a) Showing lyric words. (b) No lyric word (interlude). (c) Duplicated lyric words. (d) Words including those unrelated to the lyrics.

FIGURE 3. Variations of lyric video frames.

lyric video'' (on July 18, 2019). The keyword ''official'' was added to find videos with not only long-time availability but also professional quality. The latter is very important because we want to exclude incomplete or thoughtless video designs from our analysis. Then, the videos in the list were manually checked to exclude videos with only static motion words (i.e., videos whose lyric words did not move). Finally, the top-100 videos on the list were selected as our experimental target.[2] The frame image size is 1,920 × 1,080 pixels. The average, maximum, and minimum lengths of the videos in the dataset are 5,471 frames (3 min 38 s), 8,629 frames, and 2,280 frames, respectively. The average, maximum, and minimum numbers of lyric words are 338, 690, and 113, respectively.

Figure 3 shows four examples of lyric video frame variations. Figure 3 (a) depicts a frame showing lyric words. Typically, several words (i.e., a phrase in the song) are shown in a single frame. In the introduction, interlude, and ending parts, frames with no lyrics are often found, as shown in Figure 3 (b). In Figure 3 (c), the same word is duplicated, as in the refrain of a song. Sometimes, as shown in Figure 3 (d), the background image contains words unrelated to the lyrics.

To perform a quantitative evaluation of the word tracking method in Appendix VIII, bounding boxes were manually attached to the lyric words for 10 frames in each video. These frames were selected automatically. Specifically, for each video, the top 10 frames with the most words were selected from the frames sampled at three-second intervals. The lyric words were detected using the method described in Appendix VIII-A, and a bounding box was attached to each word in the lyrics. We attached non-horizontal bounding boxes[3] to the rotated lyric words. Consequently, we obtained $10 \times 100 = 1,000$ ground-truth frames with 7,770-word bounding boxes for the dataset.

[2]A list of all 100 videos and their annotations is published at https://github.com/uchidalab/Lyric-Video.

[3]To attach non-horizontal bounding boxes, we used the labeling tool roLabelImg available at https://github.com/cgvict/roLabelImg.

IV. WORD MOTION STYLE

A. LYRIC WORD TRACKING

We propose a word tracking method for extracting individual word motions and then quantifying their style. The proposed method is specialized to accurately track lyric words while utilizing lyric information (which is available via the metadata of the lyric video). The tracking method has three steps: word detection, word recognition, and lyric-frame matching. In the first step, lyric word candidates are detected and recognized by the method presented in Appendix VIII-A, as shown on the left-hand side of Figure 4 (a).

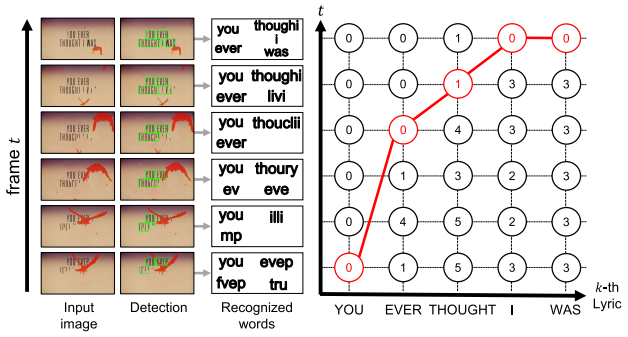
After detection and recognition, lyric-frame matching is conducted to establish the correspondence between the frames and the lyric words. The matching algorithm is based on DP-based optimization and is detailed in Appendix VIII-B. The red path on the right-hand side of Figure 4 (a) represents the optimal correspondence of the frames and lyric words. If the path passes through the grid (k, t), it means that the tth frame is determined to be the most confident frame for the kth lyric word. We then search the frames around the tth frame to find the same kth lyric word. The vertical orange paths in Figure 4 (b) depict the search results for individual lyric words. This search was done not only using simple spatio-temporal closeness but also by evaluating the word similarity of the kth word. As shown in Figure 4 (b), there are many misrecognized words; therefore, we cannot use the exact match with the lyric word in this search. Details are given in Appendix VIII-C.

The vertical orange paths for ''EVER'' and ''THOUGHT'' in Figure 4 (b) include skipped frames. For example, ''EVER'' was not detected in the second frame. Such missed detections occur because of occlusion and severe misrecognition. Therefore, we need to perform the interpolation process shown in Figure 4 (c) to complete the spatio-temporal tracking process of each lyric word. Roughly speaking, if a missed frame is found for a lyric word, the polynomial interpolation process determines the location of the lyric word in that frame. Details are given in Appendix VIII-C. Figure 5 shows the final result of the tracking process for the two lyric words ''YOU'' and ''EVER.''

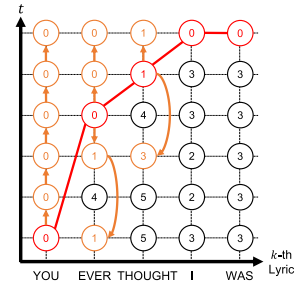
Even though the above tracking method is not perfect because of the various difficulties, the quantitative evaluation uses the ground-truth bounding boxes attached to the video frames. Specifically, as detailed in Appendix VIII-E, the tracked trajectories according to the above method show high precision. Therefore, we believe that the following word motion style analysis based on the tracking result is sufficiently reliable.

B. REPRESENTATIVE WORD MOTIONS

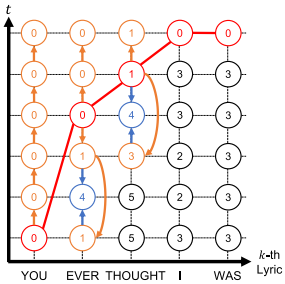
Later, in the correlation analysis, we represent the word motion style of each 30-second time window in a so-called ''bag-of-words'' manner. Specifically, the motion trajectories of all the lyric words in the video are quantized into B representative word motions, and a histogram with B bins is



(a) Detection, recognition, and lyric-frame matching.



(b) Tracking via neighbor search.



(c) Interpolation.

FIGURE 4. Lyric word detection and tracking. The circled number shows the distance $D(k, t)$ between the kth word and the frame t.

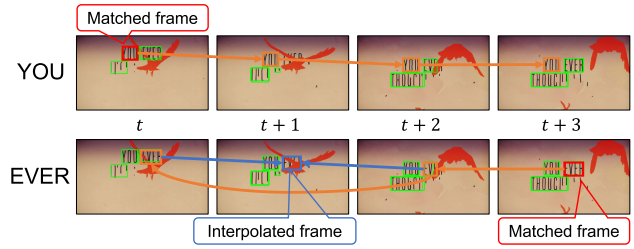


FIGURE 5. Tracking result of "YOU" and "EVER." Especially, interpolation is successfully performed for "ever."

created. Each bin corresponds to one representative motion and shows how many word trajectories are quantized to that motion. We therefore need to select the representative word motions in advance of the word style representation.

The steps to select the $B(= 70)$ representative motion trajectories of all the lyric videos in the dataset are as follows. First, each motion trajectory is represented as a sequence of four-dimensional vectors (x_1, y_1, x_2, y_2), as shown in Figure 6, where (x_1, y_1) represents the location of the center of the word bounding box, and (x_2, y_2) is defined as the upper-right corner of a square whose center is (x_1, y_1) and whose edge length is the bounding-box height. The coordinates (x_2, y_2) indirectly represent the size (word height) and rotation of the bounding box in a manner consistent with (x_1, y_1). Second, each motion trajectory is translated such that its first location (x_1, y_1) becomes $(0, 0)$. Third, the trajectories are grouped by their duration: $0.5 \sim 1.0$s (5,107), $1.0 \sim 1.5$s (4,423), $1.5 \sim 2.0$s (3,581), $2.0 \sim 2.5$s (2,744),

FIGURE 6. Four-dimensional representation of the word location and rotation.

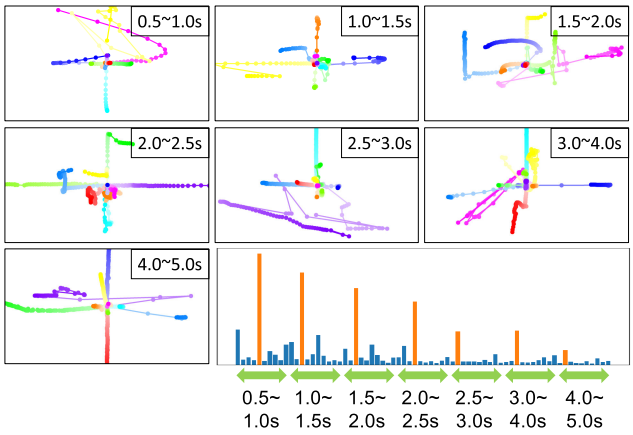


FIGURE 7. The 70 word motions (7 duration groups × 10 k-medoid clusters) and a histogram showing each cluster size. Note that this k-medoid clustering for word motions is different from the cluster analysis used to understand the style modality correlation, as described in Section VII.

$2.5 \sim 3.0$s (1,658), $3.0 \sim 4.0$s (1,742), and $4.0 \sim 5.0$s (973). The numbers in parenthesizes count the trajectories in the individual groups. Extremely short (< 0.5s) and long (> 5.0s) trajectories were rare and were excluded. Finally, k-medoid clustering ($k = 10$) was performed to a dynamic time warping distance metric at each group, and 70 representative word motions were obtained.

Figure 7 shows the 10 word motions (i.e., 10 medoids) in each of the seven duration groups, where (x_2, y_2) is omitted. The center of each plot is the origin $(0, 0)$ (i.e., the starting point of the trajectory), and the change in the color saturation (white to vivid) indicates the transition of time. The majority consists of rather simple motions: vertical, horizontal, or no-motion (i.e., staying at the origin). In addition, we show a histogram of the number of trajectories in each of the 70 clusters for all 100 lyric videos. In each of the seven duration groups, the orange bin indicates the cluster having a no-motion trajectory in which the lyric words do not move; lyric words often appear and disappear without movement.

V. FONT STYLE

To facilitate the correlation analysis performed later, we represent the font style of each video as a likelihood vector of six typical font styles: Serif, Sans-Serif, Hybrid (of Serif and Sans-Serif), Script, Historical Script, and Fancy (i.e., Display). Accordingly, the font style is given as a six-dimensional real-valued vector. For this purpose,

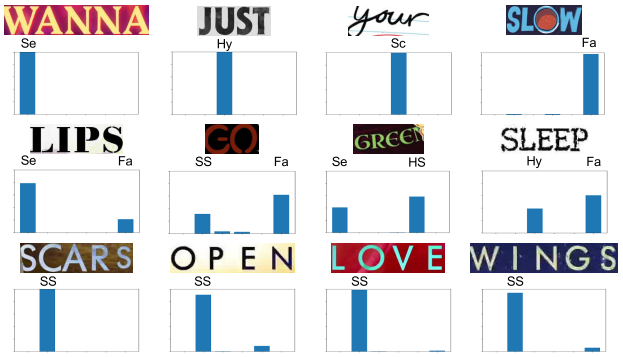


FIGURE 8. Font style estimation results for the lyric words in the lyric videos. The bar chart visualizes the likelihood of the six styles; its horizontal axis corresponds to Serif (Se), Sans-Serif (SS), Hybrid (Hy), Script (Sc), Historical Script (HS), and Fancy (Fa) from the left to right.

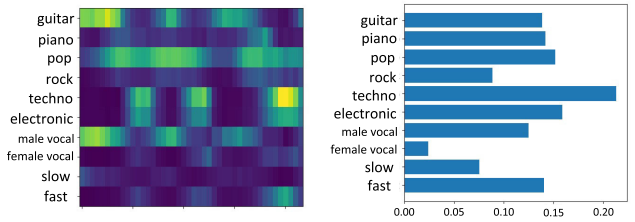
we developed a six-class font style classifier that gives the likelihood vector of each word image. The font style estimates were derived using ResNet18, a CNN trained using a large number of word images synthesized by SynthText [45]. More specifically, we first collected 510, 314, 151, 74, 58, and 704 different fonts for the Serif, Sans-Serif, Hybrid, Script, Historical Script, and Fancy classes, respectively. The class of each font was specified in a font guidebook [32]. We then generated 19,000 synthetic word images for each of the six fonts using SynthText. The images were separated into training (80%), validation (10%), and test (10%) sets, and these sets were font-disjoint. Finally, ResNet was trained as a six-class classifier using the training and validation sets. We used the six-dimensional likelihood vector given before the softmax layer of the trained ResNet as the font style vector. Note that the performance of the brute-force classification (into one of six font classes) by the trained ResNet was 81.10% for the test dataset.

Figure 8 shows the font style vectors for word images extracted from the lyric videos. The horizontal axis corresponds to the six font classes, and the vertical axis indicates their likelihood. The top row shows four cases with a high likelihood only at the correct single class. The middle row shows cases estimated as being a mixture of several font styles. The bottom row shows style vectors for word images taken from the same lyric video in which the font styles were consistent. Note that, in the experiment in Section VII, the font style vectors of all lyric words detected within each 30-second time window were averaged and then used as the font style vector for the time period.

VI. MUSIC STYLE

The music style was obtained as a 50-dimensional vector using musicnn[4] [2], [3]. We used the "MTT_musicnn" model pre-trained on the MTT dataset [44]. For the audio in a 30-second time window, we estimated a 50-dimensional tag likelihood vector corresponding to the 50 MTT tags,

[4]https://github.com/jordipons/musicnn



(a) Transition of music style. (b) Averaged style over the entire song.

FIGURE 9. Music style estimation by musicnn [2] of the lyric video of "Something Just Like This" by The Chainsmokers & Coldplay.

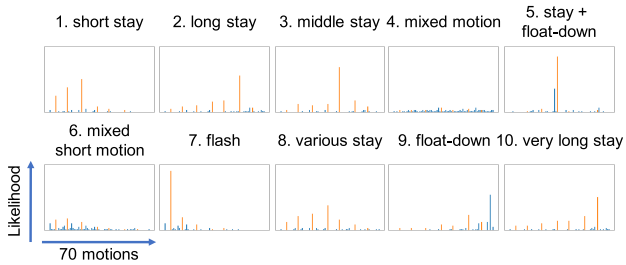


FIGURE 10. Ten word motion style types by k-means clustering. The horizontal axis corresponds to the 70 word motions presented in Figure 7. The orange bars correspond to no motion (i.e., stay) with seven different durations. A brief description, such as "flash" (very short presence), is attached to each type.

including instrument-related tags such as guitar and drums, tempo-related tags such as slow and fast, and vocal-related tags such as male and female. We simply refer to this as the music style vector, even though the estimated tags are not always style-related tags and represent a variety of musical attributes of a song, which is desirable for our research purposes. Each vector was estimated from a 30-second time window, and the estimation was performed every 5 s (at five-second intervals). For example, a sequence of 19 music style vectors can be extracted from a 120-second song $((120 - 30)/5 + 1 = 19)$.

Figure 9 (a) visualizes an actual music style vector sequence as a heatmap, where the horizontal axis indicates time and the vertical axis shows 10 tags of the 50 tags. Yellow indicates the highest value (i.e., 1). In this example, the music style changes along the time axis because of various interludes. Figure 9 (b) shows the averaged style vector over the same song and indicates that this music is sung by a male and has a techno mood with a fast tempo. Note that we do not use the averaged vector in Figure 9 (b) but rather the vector sequence in Figure 9 (a) in the later analysis.

VII. CORRELATION ANALYSIS BETWEEN THE THREE STYLE MODALITIES

A. TEN REPRESENTATIVE TYPES OF EACH STYLE MODALITY

As shown in Figure 2, we conducted a correlation analysis between the three style modalities of word motion, font style,

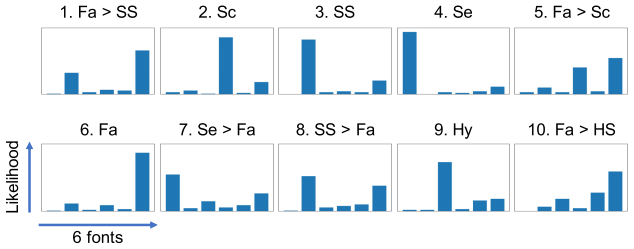


FIGURE 11. Ten font style types by k-means clustering. The horizontal axis corresponds to Serif (Se), Sans-Serif (SS), Hybrid (Hy), Script (Sc), Historical Script (HS), and Fancy (Fa), from left to right. A brief description, e.g., "Se > Fa" (Serif is presented more than Fancy), is attached to each type.

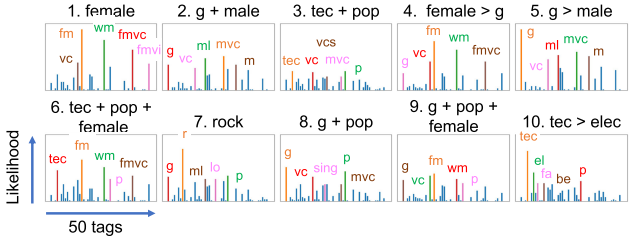


FIGURE 12. Ten music style types by k-means clustering. The five tags with the highest likelihood are printed in orange, green, red, brown, and pink. The abbreviations are as follows: vc: vocal, fm: female, wm: woman, fmvc: female vocal, fmvi: female voice, g: guitar, ml: male, mvc: male vocal, m: man, tec: techno, vcs: vocals, p: pop, r: rock, lo: loud, sing: singing, el: electronic, fa: fast, and be: beat.

and music style. Each feature vector of the three style modalities was extracted from a 30-second time window with a 5-second interval. Consequently, every 5 s, we obtained 70-, 6-, and 50-dimensional vectors for the word motion, font style, and music style, respectively. Note that time windows with no lyric words or only a few words were discarded from the analysis. This resulted in 3,494 feature vectors for each style modality from the 100 lyric videos. We used all of these vectors in the following clustering-based correlation analysis.

In advance of the correlation analysis, standard k-means clustering[5] was performed to quantize the style vectors of each modality. As noted in Section I, cluster analysis is more promising for our task than orthodox multivariate analysis techniques, such as deep regression. Determination of the number of clusters (hyper-parameter k) relies on several criteria, such as the silhouette coefficient [47], the Calinski and Harabasz score [48], and the Davies-Bouldin index [49]. We examined these criteria but found no unanimous suggestion for the value of k. We therefore took the intermediate value of $k = 10$ for all of the style modalities. Consequently, we had $k = 10$ representative types (representative centroid vectors), as shown in Figures 10, 11, and 12 for the word motion, font style, and music style modalities, respectively.

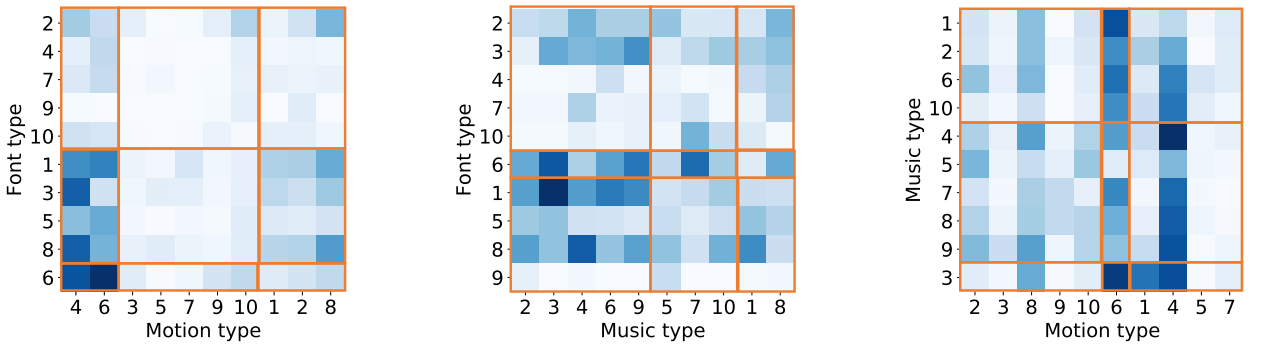
B. CO-OCCURRENCE ANALYSIS BETWEEN THE STYLE MODALITIES

As noted above, for every 5 s, word motion, font style, and music style feature vectors were obtained via an analysis of the 30-second time window. Let those feature vectors be denoted by $w_{t,s} \in \mathbb{R}_+^{70}$, $f_{t,s} \in \mathbb{R}_+^{6}$, and $m_{t,s} \in \mathbb{R}_+^{50}$, respectively, where t is the frame index and $s \in [1, 100]$ is the lyric video ID. We quantized these vectors into the nearest vector of the 10 representative vectors (types) in each modality. Consequently, we obtained $W_{t,s}$, $F_{t,s}$, and $M_{t,s}$, each of which represents the nearest vector index $\in [1, 10]$. Then, we obtained a 10×10 co-occurrence matrix for each pair of two modalities. For example, the co-occurrence matrix $\mathbf{C}_{f,m}$ between the font style and the music style was created by adding 1 to the $(F_{t,s}, M_{t,s})$th element of the matrix for all t and s.

Figure 13 shows the co-occurrence matrices for all three pairs of style modalities. The matrices were pre-processed with biclustering (row-wise and column-wise reordering) such that blocks (sub-matrices) became more visible. Via careful observations of the matrices, the following trends in the lyric video designs were indicated.

- *Word motion and font style* (Figure 13 (a)): There is a block with high co-occurrence in the bottom-left area of the matrix. This block is the intersection of two motion types, #4 (mixed motion) and #6 (mixed short motion), and five font types, #1 (Fa>SS), #3 (SS), #5 (Fa>Sc), #8 (SS>Fa), and #6 (Fa). This indicates that lyric words with active motions (i.e., not "no-motion") are often printed in Fancy and Sans-Serif. See examples in Figure 14.
- *Font style and music style* (Figure 13 (b)): There is another block with high co-occurrence near the bottom-left area of the matrix. This block is the intersection of five music types, #2 (g+male), #3 (tec+pop), #4 (female>g), #6 (tec+pop+female), and #9 (g+pop+female), and four (or five) font types, #6 (Fa), #1 (Fa>SS), #5 (Fa>Sc), and #8 (SS>Fa) (and #9 (Hy), which is weaker). This block suggests that "Fancy" fonts tend to be used for "guitar" and "pop" music.
- *Word motion and music style* (Figure 13 (c)): Motion types #4 (mixed motion) and #6 (mixed short motion) are scattered across all music types except for #5 (g>male). However, observing local correlations, there are strong peaks at #6 (mixed short motion) - #3 (tec+pop), #6 (mixed short motion) - #1 (female), and #4 (mixed motion) - #4 (female>g). These co-occurrences suggest that mixed motions (i.e., active

[5]We compared the results of k-means clustering and the results of agglomerative clustering (so-called hierarchical clustering) by using the adjusted rand index [46], and the results showed that the index score of word motion style types, font style types, and music style types is 0.28, 0.60, and 0.49, respectively. This index becomes 1 when two clustering results are completely the same and zero when there is no correlation. Although the word motion style has a weak correlation, the font style and music style have strong correlations. Therefore, the choice of clustering algorithms is not very sensitive in our task.



(a) $\mathbf{C}_{w,f}$: Word motions and font styles. (b) $\mathbf{C}_{f,m}$: Font styles and music styles. (c) $\mathbf{C}_{w,m}$: Word motions and music styles.

FIGURE 13. Co-occurrence matrices for each modality pair. Biclustering was applied to the matrix for better visibility. Each orange box indicates a bicluster in the matrix.

motions) tend to be used for music with female vocals. Moreover, for music type #3 (tech+pop), various short and active motions are used for the lyric words.

There are also other interesting strong co-occurrences in Figure 13; for example, "Fancy" and "Historical Script" fonts (#6 and #10) are usually used for "rock" music (#7) as shown in Figure 13 (b).

These trends found in our analysis could be useful for assisting in the design of lyric videos. Even though we highlighted strong co-occurrences in the above analysis, no or low co-occurrences might also provide useful information concerning the trends in lyric videos. However, we do not emphasize those low co-occurrences in this paper because they may be caused by insufficient lyric video data and a much larger dataset might prove the importance of such low co-occurrences in future research.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we tackled the novel task of analyzing lyric videos to understand the relationships between three style modalities: word motion, font style, and music style. To conduct this analysis, we developed an original lyric word tracking method, which is detailed in Appendix VIII, and an original font style estimator. Moreover, the clustering-based co-occurrence analysis of the style modalities from 100 lyric videos indicated several trends in the style combinations. That is, we were able to catch such trends in the videos in an objective and reproducible manner without manual annotations.

Because multi-modal analyses of lyric videos have not previously been explored and this paper is the first such attempt, there are tasks left as future work. First, the dataset can be expanded to make the analysis result more reliable. Second, using the discovered trends in the multi-modal style combinations, a recommendation system can be developed to decrease the difficulty of lyric video creation. For example, if a system can automatically suggest word motions and font styles for given music (i.e., audio), even a non-expert could easily create lyric videos. Third, the design of the video background images could be incorporated into the analysis. Even though the color and objects in the background images were not the focus of this paper, they are also an important modality and therefore cannot be ignored in the overall design analysis of lyric videos.

APPENDIX A LYRIC WORD DETECTION AND TRACKING BY USING LYRIC INFORMATION

We introduce the methodology [5] used to detect and track lyric words in a lyric video. The technical highlight of the methodology is the full use of the lyric information (i.e., the lyric word sequence of the song) to obtain accurate tracking results. Note that the conference paper [5] focused only on the word motion style and not on the font style or music style; therefore, no correlation analysis between the style modalities had previously been made.

A. LYRIC WORD CANDIDATE DETECTION

First, lyric word candidates are detected as bounding boxes using two pretrained state-of-the-art scene text detectors, PSENet [50] and CRAFT [51]. The detected bounding boxes are then fed into a state-of-the-art scene text recognizer `TPS-Resnet-BiLSTM-Attn`, which was proposed in [52]. If bounding boxes detected by the above detectors overlap by more than 50%, and the recognition results are the same, these bounding boxes are regarded as duplicates. Accordingly, we remove either box in the later process.

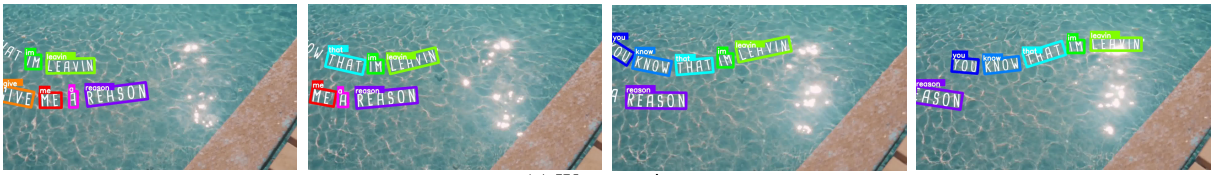
B. LYRIC-FRAME MATCHING

As we described in IV-A, the lyric-frame matching was conducted by associating the word sequence and the frame sequence of the given lyrics after detection and recognition. The red matching path shown in Figure 4 (a) was determined by evaluating the distance $D(k, t)$ between the kth word and the tth frame. A smaller value of $D(k, t)$ means that the probability of the kth lyric word existing in the tth frame is high. More precisely, the distance $D(k, t)$ is defined as $D(k, t) = \min_{b \in B_t} d(k, b)$, where B_t is the set of bounding boxes detected in the tth frame and $d(k, b)$ is the edit function between the kth lyric word detected in the tth frame and the

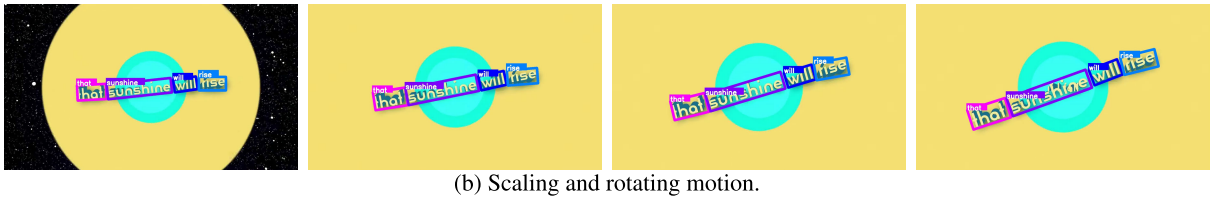
(a) Scaling motion with the "Fancy" font style.

(b) Sliding motion with the "Sans-Serif" font style.

FIGURE 14. Scenes with active motions and the font styles of "Fancy" and "Sans-Serif." These videos have remarkable trends in their relationships. You can see these videos on YouTube. Note that the four still-frame images are arranged in order from left to right.



(a) Wavy motion.



(b) Scaling and rotating motion.

FIGURE 15. Successful results of lyric word detection and tracking under various motion types. The series of video frames is arranged in order from left to right. The bounding boxes of the detected words are shown.

bth word in the same frame. If the kth lyric word is perfectly detected in the tth frame, the distance is $D(k, t) = 0$.

Using the distance $\{D(k, t)|\forall k, \forall t\}$ for dynamic programming (DP), we can efficiently obtain the globally optimal lyric frame matching as shown in the red path in Figure 4 (a). In the dynamic time warping (DTW) algorithm, the DP recursion is calculated for each (k, t) from $(k, t) = (1, 1)$ to (K, T) as follows:

$$g(k, t) = D(k, t) + \min_{t-\Delta \le t' < t} g(k-1, t'),$$

where $g(k, t)$ shows the minimum accumulated distance from $(1, 1)$ to (k, t). The parameter Δ indicates the maximum frame skipped on the path. In the experiment, we set $\Delta = 1,000$. This means that a video with 24 fps is allowed to skip approximately 40 s. The calculation complexity of the algorithm is $O(\Delta TK)$.

Note that this lyric-frame matching process using lyric information is essential for lyric videos. For example, the word "the" appears many times in the lyric text; this means that the spatio-temporal location of a certain "the" is ambiguous. Therefore, the lyric-frame matching process needs to fully utilize the continuity of the lyric words, as well as the video frames to determine the most reliable frame for each lyric word.

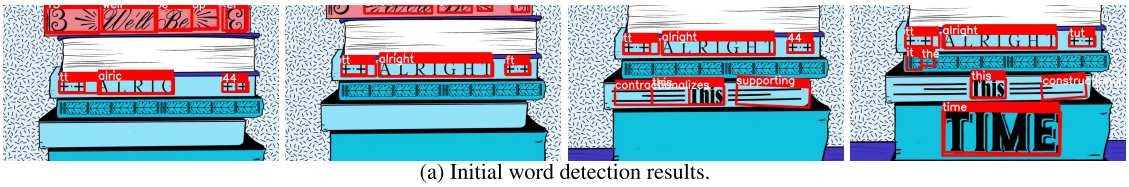
C. TRACKING OF INDIVIDUAL LYRIC WORDS

In the above lyrics-frame matching step, the kth lyric word is only matched to the tth frame; however, this word may also appear around the tth frame. Therefore, we search for such frames around the tth frame, as shown in Figure 4 (b). This search is done not only via simple spatio-temporal similarity but also by evaluating the word similarity with the kth word in the neighboring tth frames. If both similarities are larger than a threshold in the t'th frame, we conclude that the same kth word is also found in the t'th frame.

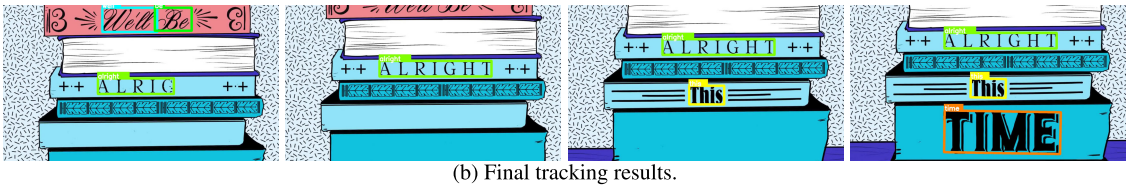
Finally, as shown in Figure 4 (c), we conduct an interpolation process as post-processing. If a lyric word is seriously misrecognized and/or occluded in a certain frame, we cannot track the word around the frame using the above simple searching process. If such a missed frame is found, polynomial interpolation is performed between the neighboring frames. The average running time of lyric word tracking per frame is approximately 440 ms.

D. QUALITATIVE EVALUATION OF LYRIC WORD DETECTION AND TRACKING

We applied the above method to all of the frames of the 100 collected lyric videos (approximately 547,100 frames in total) and obtained tracking results for all of the lyric words



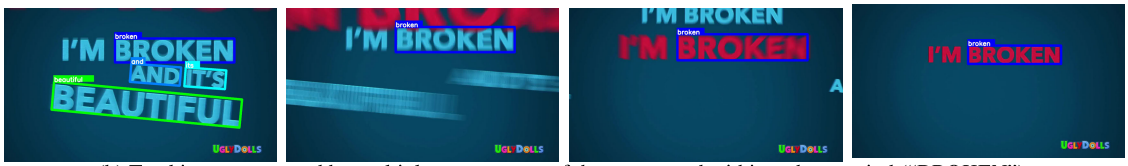
(a) Initial word detection results.



(b) Final tracking results.

FIGURE 16. Effect of lyric information. The lyric words in these frames show "we'll be alright this time."

(a) Heavy distortion by partial occlusion ("PREY").



(b) Tracking error caused by multiple appearances of the same word within a short period ("BROKEN").

FIGURE 17. Failure results of lyric word detection and tracking.

(approximately 33,800 words in total). Figure 15 shows several successful results of lyric word detection and tracking. In Figure 15 (a), we can see that a word with wavy motion can be correctly tracked. As shown in Figure 15 (b), a word under scaling or rotation for each frame can also be correctly tracked.

Figure 16 shows the effect of using lyric information in the lyric-frame matching process and subsequent tracking process to improve accuracy. Because these frames have a complex background (character-like patterns), unnecessary bounding boxes are found in the first word detection step; however, only the correct lyric words remain after matching and tracking the lyric frames.

Figure 17 shows typical failure cases. The failure in Figure 17 (a) is caused by severe distortion resulting from the complicated visual design of the video. The word "PREY" is always partially occluded and therefore never detected, even by the state-of-the-art word detector. The failure in Figure 17 (b) is caused by a refrain of the same phrase "I'M BROKEN" in the lyrics. In lyric videos, an important lyric word or phrase sometimes appears repeatedly (i.e., excessively) while changing its appearance, even though the lyric text contains it only one time.

E. QUANTITATIVE EVALUATION OF LYRIC WORD DETECTION AND TRACKING

Table 1 shows the result of a quantitative evaluation of the lyric word detection and tracking using 1,000 frames described in III as ground-truth data. If the bounding boxes of a lyric word according to the proposed method and the corresponding ground-truth data have IoU > 0.5, the detected box is considered to be a successful result. The evaluation result of the lyric-frame matching step and the later tracking step indicates that the precision is 90.98%. From this, we can see that the false positives are more successfully suppressed than in the case of only lyric-frame matching. The introduction of the interpolation step increased the true positives as expected, even though false positives were also unfortunately increased and the precision value was slightly decreased. The recall is approximately 71%. The main reasons for false positives are too many decorations and distortions in the word appearance, lyric-frame matching errors resulting from ambiguities in matching, and inconsistency between official lyric texts and actual sung lyrics.

APPENDIX B VIDEOS SHOWN IN THE FIGURES

The figures in this paper can be seen in the frame of the following videos. For URLs, the common prefix "`https://www.youtube.com/watch?v=`" is omitted in the list. Note that the URL list of all 100 videos and their annotation data can be found at `https://github.com/uchidalab/Lyric-Video`.

- Figure 1: Dua Lipa, New Rules, `AyWsHs5QdiY`
- Figure 2: Major Lazer & DJ Snake, Lean On, `rn9AQoI7mYU`

TABLE 1. Quantitative evaluation of the lyric word detection and tracking. MA: Lyric-frame matching. TR: tracking. IN: interpolation. TP: #true-positive. FP: #false-positive. FN: #false-negative. P: precision (%). R: recall (%). F: f-measure.

MA	TR	IN	TP	FP	FN	$P = \frac{TP}{TP+FP}$	$R = \frac{TP}{TP+FN}$	$F = \frac{2PR}{P+R}$
✓			72	12	7,698	85.71	0.93	0.0183
✓	✓		5,513	462	2,257	92.27	70.95	0.8022
✓	✓	✓	5,547	550	2,223	90.98	71.39	0.8000

- Figure 3: (a) Kelly Clarkson, Broken & Beautiful, `6l8gyacUq4w`; (b) Green Day, Too Dumb to Die, `qh7QJ_jLam0`; (c) Rita Ora, Your Song, `i95Nlb7kiPo`; (d) Selena Gomez, Only You, `T2urfFpDX1c`
- Figures 4 and 5: Freya Ridings, Castles, `pL32uHAiHgU`
- Figure 6: (left) Anne-Marie, 2002, `1tvLIhEaEKo`; (right) Loud Luxury feat. brando, Body, `IetIg7y5k3A`
- Figure 14: (a) Ed Sheeran, Shape Of You, `_dK2tDK9grQ`; (b) Kelly Clarkson, Broken & Beautiful, `6l8gyacUq4w`
- Figure 15: (a) blackbear, wanderlust, `YCRnw3WELY4`; (b) 311, What The?!, `gUGxyD-NOGo`
- Figure 16: Ed Sheeran, Perfect, `iKzRIweSBLA`
- Figure 17: (a) Imagine Dragons, Natural, `V5M2WZiAy6k`; (b) Kelly Clarkson, Broken & Beautiful, `6l8gyacUq4w`

ACKNOWLEDGMENT

(Daichi Haraguchi and Shota Sakaguchi contributed equally to this work.)

REFERENCES

[1] J. Kato, T. Nakano, and M. Goto, "TextAlive: Integrated design environment for kinetic typography," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 3403–3412.

[2] J. Pons and X. Serra, "Musicnn: Pre-trained convolutional neural networks for music audio tagging," 2019, *arXiv:1909.06654*.

[3] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proc. 19th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2018, pp. 637–644.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[5] S. Sakaguchi, J. Kato, M. Goto, and S. Uchida, "Lyric video analysis using text detection and tracking," in *Proc. 14th Int. Workshop Document Anal. Syst. (DAS)*, 2020, pp. 426–440.

[6] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Image Process.*, vol. 20, no. 3, pp. 790–799, Mar. 2010.

[7] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 385–392, Apr. 2000.

[8] Z. Yang and P. Shi, "Caption detection and text recognition in news video," in *Proc. 5th Int. Congr. Image Signal Process. (CISP)*, Oct. 2012, pp. 188–191.

[9] H. Yang, B. Quehl, and H. Sack, "A framework for improved video text detection and recognition," *Multimedia Tools Appl.*, vol. 69, no. 1, pp. 217–245, Mar. 2014.

[10] D. Zhong, P. Shi, D. Pan, and Y. Sha, "The recognition of Chinese caption text in news video using convolutional neural network," in *Proc. IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Oct. 2016, pp. 658–662.

[11] I. A. Zedan, K. M. Elsayed, and E. Emary, "Caption detection, localization and type recognition in Arabic news video," in *Proc. 10th Int. Conf. Informat. Syst. (INFOS)*, 2016, pp. 114–120.

[12] L.-H. Chen and C.-W. Su, "Video caption extraction using spatio-temporal slices," *Int. J. Image Graph.*, vol. 18, no. 2, Apr. 2018, Art. no. 1850009.

[13] Y. Xu, S. Shan, Z. Qiu, Z. Jia, Z. Shen, Y. Wang, M. Shi, and E. I.-C. Chang, "End-to-end subtitle detection and recognition for videos in east Asian languages via CNN ensemble," *Signal Process. Image Commun.*, vol. 60, pp. 131–143, Feb. 2018.

[14] W. Lu, H. Sun, J. Chu, X. Huang, and J. Yu, "A novel approach for video text detection and recognition based on a corner response feature map and transferred deep convolutional neural network," *IEEE Access*, vol. 6, pp. 40198–40211, 2018.

[15] X. Qian, G. Liu, H. Wang, and R. Su, "Text detection, localization, and tracking in compressed video," *Signal Process. Image Commun.*, vol. 22, no. 9, pp. 752–768, Oct. 2007.

[16] P. Xuan Nguyen, K. Wang, and S. Belongie, "Video text detection and recognition: Dataset and benchmark," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 776–783.

[17] L. Gomez and D. Karatzas, "MSER-based real-time text detection and tracking," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 3110–3115.

[18] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "A new technique for multi-oriented scene text line detection and tracking in video," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1137–1152, Aug. 2015.

[19] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from web videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 542–554, Mar. 2017.

[20] C. Yang, X. C. Yin, W. Y. Pei, S. Tian, Z. Y. Zuo, C. Zhu, and J. Yan, "Tracking based multi-orientation scene text detection: A unified framework with dynamic programming," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3235–3248, Jul. 2017.

[21] W. Y. Pei, C. Yang, L. Y. Meng, J. B. Hou, S. Tian, and X. C. Yin, "Scene video text tracking with graph matching," *IEEE Access*, vol. 6, pp. 19419–19426, 2018.

[22] Y. Wang, L. Wang, and F. Su, "A robust approach for scene text detection and tracking in video," in *Proc. 19th Pacific Rim Conf. Multimedia (PCM)*, 2018, pp. 303–314.

[23] Y. Wang, L. Wang, F. Su, and J. Shi, "Video text detection with fully convolutional network and tracking," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2019, pp. 1738–1743.

[24] X. Yin, Z. Zuo, S. Tian, and C. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[25] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 843–852.

[26] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 877–882, Aug. 1998.

[27] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang, "DeepFont: Identify your font from an image," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 451–459.

[28] N. Goel, M. Sharma, and L. Vig, "Font-ProtoNet: Prototypical network based font identification of document images in low data regime," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 556–557.

[29] O. Bychkov, K. Merkulova, G. Dimitrov, Y. Zhabska, I. Kostadinova, P. Petrova, P. Petrov, I. Getova, and G. Panayotova, "Using neural networks application for the font recognition task solution," in *Proc. 55th Int. Sci. Conf. Inf., Commun. Energy Syst. Technol. (ICEST)*, Sep. 2020, pp. 167–170.

[30] B. Bauermeister, *A Manual of Comparative Typography: The PANOSE system*. New York, NY, USA: Van Nostrand Reinhold Company, 1988.
[31] Y. Shinahara, T. Karamatsu, D. Harada, K. Yamaguchi, and S. Uchida, "Serif or sans: Visual font analytics on book covers and online advertisements," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1041–1046.
[32] *Type Identifier for Beginners*. Seibundo Shinkosha, 2013.
[33] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi, and R. Ingold, "A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution," *Pattern Recognit. Lett.*, vol. 34, no. 2, pp. 209–218, Jan. 2013.
[34] D. Eck, T. Bertin-Mahieux, and P. Lamere, "Autotagging music using supervised machine learning," in *Proc. 8th Int. Conf. Music Inf. Retr. (ISMIR)*, 2007, pp. 367–368.
[35] M. D. Hoffman, D. M. Blei, and P. R. Cook, "Easy as CBA: A simple probabilistic model for tagging music," in *Proc. 10th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2009, pp. 369–374.
[36] E. Coviello, L. Barrington, A. B. Chan, and G. R. G. Lanckriet, "Automatic music tagging with time series models," in *Proc. 11th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2010, pp. 81–86.
[37] M. I. Mandel, D. Eck, and Y. Bengio, "Learning tags that vary within a song," in *Proc. 11th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2010, pp. 399–404.
[38] G. Marques, M. A. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging," in *Proc. 12th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2011, pp. 795–800.
[39] E. Coviello, Y. Vaizman, A. B. Chan, and G. R. G. Lanckriet, "Multivariate autoregressive mixture models for music auto-tagging," in *Proc. 13th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2012, pp. 547–552.
[40] D. Liang, J. W. Paisley, and D. Ellis, "Codebook-based scalable music tagging with Poisson matrix factorization," in *Proc. 15th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2014, pp. 167–172.
[41] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proc. 17th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2016, pp. 805–811.
[42] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning for audio-based music classification and tagging," in *Proc. 20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2019, pp. 67–74.
[43] K. M. Ibrahim, E. V. Epure, G. Peeters, and G. Richard, "Should we consider the users in contextual music auto-tagging models?" in *Proc. 21st Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2020, pp. 295–301.
[44] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proc. 10th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2009, pp. 387–392.
[45] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.
[46] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.
[47] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.
[48] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat., Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[49] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[50] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9336–9345.
[51] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9365–9374.
[52] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4715–4723.

DAICHI HARAGUCHI received the B.S. degree from Saga University, Saga, Japan, in 2019, and the M.Eng. degree from Kyushu University, Fukuoka, Japan, in 2021, where he is currently pursuing the Ph.D. degree with the Graduate School of Information Science and Electronic Engineering. His current research interests include computer vision and pattern recognition.

SHOTA SAKAGUCHI received the B.E. degree from the Sasebo College, National Institute of Technology, Nagasaki, Japan, in 2019, and the M.Eng. degree from Kyushu University, Fukuoka, Japan, in 2021. His current research interests include computer vision and pattern recognition.

JUN KATO received the Ph.D. degree from The University of Tokyo, in 2014. Prior to graduation, he worked at Microsoft and Adobe Research. He is currently a Senior Researcher with the National Institute of Advanced Industrial Science and Technology (AIST) and also works as the Technical Advisor at Arch Inc., Japan. He has focused on human–computer interaction research, designing tools for programming and authoring multimedia content, and regularly gained academic recognition, such as ACM CHI 2013/2015 Best Paper Honorable Mention and 2021 IPSJ/ACM Award for Early Career Contributions to Global Research.

MASATAKA GOTO received the Doctor of Engineering degree from Waseda University, Tokyo, Japan, in 1998. He is currently a Prime Senior Researcher with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. Over the past 30 years, he has published more than 300 papers in refereed journals and international conferences and has received 57 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and the Tenth JSPS PRIZE. He has served as a Committee Member of over 120 scientific societies and conferences, including the General Chair of ISMIR 2009 and 2014.

SEIICHI UCHIDA (Member, IEEE) received the B.E., M.E., and Dr. (Eng.) degrees from Kyushu University, in 1990, 1992, and 1999, respectively. He is currently a Distinguished Professor at Kyushu University. His research interests include image-informatics, especially document analysis and recognition (DAR). He received the 2007 ICDAR Best Paper Award and many other awards. He acted as a Program Chair at DAR-related conferences, such as ICDAR2021. He is an Associate Editor of *Pattern Recognition*.