



Multi-Script Video Caption Localization Based on Visual Rhythms

Marcos Roberto e Souza, Helena de Almeida Maia, Anderson Carlos Souza e Santos, Marcelo Bernardes Vieira & Helio Pedrini

To cite this article: Marcos Roberto e Souza, Helena de Almeida Maia, Anderson Carlos Souza e Santos, Marcelo Bernardes Vieira & Helio Pedrini (2022) Multi-Script Video Caption Localization Based on Visual Rhythms, Applied Artificial Intelligence, 36:1, 2032926, DOI: [10.1080/08839514.2022.2032926](https://doi.org/10.1080/08839514.2022.2032926)

To link to this article: <https://doi.org/10.1080/08839514.2022.2032926>



© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 04 Feb 2022.



Submit your article to this journal [↗](#)



Article views: 726








View related articles [↗](#)



View Crossmark data [↗](#)

Multi-Script Video Caption Localization Based on Visual Rhythms

Marcos Roberto e Souza ^a, Helena de Almeida Maia ^a,
Anderson Carlos Souza e Santos ^a, Marcelo Bernardes Vieira ^b,
and Helio Pedrini ^a

^aInstitute of Computing, University of Campinas, Campinas, Brazil; ^bDepartment of Computer Science, Federal University of Juiz de Fora (UFJF), Juiz de Fora, Brazil

ABSTRACT

Localization of video caption plays an important role in information retrieval in multimedia applications. In this work, we present and evaluate a novel method for localizing video captions using visual rhythms, which enable the representation and analysis of a specific feature throughout the time. We build visual rhythms from the text location maps produced by general text localization methods that are far more common in the literature than caption-oriented ones. Then, we process the maps properly to keep only the captions, generating caption localization masks. To meet the need for a standardized and large dataset, we constructed a new one, where captions with thirteen different scripts are added to the video frames, generating a total of 221 videos with ground truth. Experiments demonstrate that our method achieves competitive results when compared to other literature approaches.

ARTICLE HISTORY

Received 17 July 2021
Revised 20 December 2021
Accepted 18 January 2022

Introduction

Texts present in videos can be categorized into scene texts and captions (Zedan, Elsayed, and Emary 2016). Scene texts occur naturally in video content, whereas captions are artificially embedded into the video frames. In this work, we are particularly interested in localizing video captions, which are usually static, that is, fixed in the same position in some consecutive frames. However, captions can also be scrolled, such as the vertical scrolling in video credits.

Several recent works have investigated different tasks related to video captions, whose development can assist the task of video content analysis (Valio, Pedrini, and Leite 2011). We define detection as the task that aims to verify whether there is a caption in a given frame. The localization problem, in turn, returns a bounding box for the caption of each frame.

CONTACT Helio Pedrini  helio@ic.unicamp.br  Institute of Computing, University of Campinas, Campinas 13083-852, Brazil

© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The script recognition problem aims to recognize the script of the text in order to facilitate the task of text recognition. At last, the text recognition aims to determine the words that are written in the text.

The majority of text localization methods proposed in the literature were not focused on captions. However, they can be used as baseline for methods addressing this specific problem, provided that additional steps are proposed to distinguish the two types of text. Thus, the main objective of our work is to propose a new method for caption localization based on general text methods.

As a first contribution of this work, we propose and evaluate a novel method for localizing video captions based on the visual rhythm. Visual rhythm (Souza et al. 2020) is a spatiotemporal representation that summarizes relevant information present in a video. It is defined as the concatenation of a predefined feature extracted from each frame or small groups of neighboring frames. Initially, we apply a text localization method in the video frames that results in binary maps in which white pixels indicate text regions. Then, we build visual rhythms from the text location maps. The components of these rhythms are properly processed to keep only the captions. Finally, caption localization masks are retrieved from processed rhythms. We can use any text localization method in the first step, including those for natural text. The remaining process aims to transform the initial text localization into caption localization. This is completely deterministic and does not use or require any learning-based technique.

Similar to specific methods, there is also a lack of large datasets containing the necessary annotation to evaluate the methods. Since the methods do not usually employ a standardized and public dataset, it is difficult to compare their results and generalize their findings. For this reason, our second contribution is a new multi-script dataset with ground truth for video caption tasks. In order to build this database, we collected 17 distinct videos from YouTube with subtitles and under the Creative Commons license. Each subtitle was obtained in English and then translated in such a way that we get 11 different scripts. Subtitles are added to videos according to pre-set settings as font color and size. By including 11 types of subtitles in each of the 17 different videos, we built a dataset with 221 unique videos. Concerning the number of frames, our dataset has 87,789 frames from the original videos and more than 1 million frames with subtitles. Since the detection and localization are performed in the frames and not in the video, it can be considered a large dataset. To the best of our knowledge, there is no similar dataset in the literature. Experiments demonstrate that the proposed method achieves superior results when compared to the method used as a baseline. We believe that our results can be even better in future work if we improve the performance of the caption detection step.

This text is organized as follows. This section presents a brief introduction with some relevant concepts. Related work is described in [Section 2](#). [Section 3](#) presents the proposed method for video caption localization. The description of the proposed dataset and its construction is provided in [Section 4](#). [Section 5](#) reports and discusses the obtained results. Final considerations and directions for future work are outlined in [Section 6](#).

Background

This section presents a brief review of the literature related to the topic investigated in this work. Methods for text localization in still images are presented in [Subsection 2.1](#). Methods for text detection and localization in videos are described in [Subsection 2.2](#).

Text Localization in Images

Long, He, and Ya (2018) conducted a detailed literature review, categorizing the works based on traditional image processing and deep learning techniques. Prior to the advent of deep learning techniques, methods mainly used connected component and sliding window strategies. Some of these works are described as follows.

Wu, Hsieh, and Chen (2008) presented an algorithm for localization of text regions in images based on mathematical morphology techniques. First, an average filter is applied to smooth the image. Then, the difference between the opening and closing operations of the smoothed image is calculated. A closing operation is applied to the image of the differences to merge the characters into a single component. The image is binarized and the components are labeled. Since texts can still be divided into several small segments of different orientations, the orientation of the text is estimated based on statistical moments. From the texts and their rotation angles, their properties are used to select the candidate texts. The nearby text segments are joined. Then, an x -projection technique is used to extract features from the candidate texts and verify them.

Epshtein, Ofek, and Wexler (2010) proposed an operator to find the stroke width value for each image pixel, and applied it to the text detection problem. Initially, each pixel is set to an infinite value. Edges are computed using the Canny algorithm (Canny 1987). For each edge pixel, its gradient direction is used to find the next one, which is expected to be in the opposite edge of the same stroke. Each intermediate pixel in this path receives the value of the distance between the two edge pixels, if it is smaller than the current value. Based on these distances, pixels with similar values are clusterized, making them candidates for letters, which are filtered according to their size. The letters are finally grouped into lines of text.

Neumann and Matas (2010) proposed localizing text in images using Maximally Stable Extremal Regions (MSER). Each of the localized regions is classified by the Support Vector Machine (SVM) into characters and non-characters using features such as aspect ratio and compactness. From similarity and spatial proximity features, the characters are joined in lines.

Zhang et al. (2016) used a fully convolutional network (FCN) to detect text blocks. The network has five convolutional stages based on the 16-layer VGG model. A deconvolution layer is added at the end of each stage. The fusion of the result of each feature map produces the saliency map for text detection. Several other neural network architectures have been proposed for text detection, localization, and recognition in images (Arafat and Iqbal 2020; 2017b; He et al. 2017a; Jiang et al. 2020, 2017; Katper et al. 2020; Liao et al. 2018; Liu and Jin 2017; Shi, Bai, and Belongie 2017; Villamizar, Canévet, and Odohez 2020).

Text Localization in Videos

Yin et al. (2016) investigated the most relevant approaches to detection, tracking and recognition of texts in videos. High-pass filters were used for text detection in videos (Agnihotri and Dimitrova 1999). Other works used the coefficients of the Discrete Cosine Transform (DCT) in the Moving Picture Expert Group (MPEG) domain, in order to perform text detection in compressed videos with low computational cost (Zhang and Chua 2000; Zhong, Zhang, and Jain 2000).

Khare, Shivakumara, and Raveendran (2015) proposed a new descriptor for text localization in videos with the invariance of rotation, scaling, font type, and font size. For each frame, the proposed descriptor finds the orientations with the second-order geometric moments. Text candidates are obtained from the analysis of the dominant orientations of the connected components. Candidates of text with constant speed and uniform direction, verified by optical flow analysis, are finally considered as text.

Searching for captions on all video frames can be computationally expensive. In order to reduce the cost, visual rhythm representation (Chun et al. 2002; Concha et al. 2018; Moreira, Menotti, and Pedrini 2017; Pinto et al. 2012, 2015; Souza 2018; Souza et al. 2020; Tacon et al. 2019; Torres and Pedrini 2018; Valio, Pedrini, and Leite 2011) can be used to detect frames in which captions are present. Thus, localization techniques can only be applied to the appropriate frames. Chun et al. (2002) proposed the visual rhythm for the frame detection problem. The visual rhythm is constructed by concatenating the vertical, main and secondary diagonal rhythms. Prewitt filter is applied to the visual rhythm to highlight the horizontal edges. Caption is detected based on the analysis of certain properties such as duration.

Lyu, Song, and Cai (2005) proposed a method for detection, localization and extraction of texts in videos. Text detection is done through edge detection and local thresholding. Text localization is performed through a coarse-to-fine analysis. Text extraction is done by adaptive thresholding, followed by labeling and padding steps.

Lee et al. (2007) detected captions in videos based on the assumption that caption holds in many consecutive frames. Initially, they identify the frames in which new captions start with a strategy based on decreasing the frame rate. Twelve wavelet features are extracted from the region, which are used as input to a classifier that determines whether that region refers to a caption.

Valio, Pedrini, and Leite (2011) addressed the problem of caption detection with rotation invariance through visual rhythms. Initially, the visual rhythm is calculated and segmented. Captions are then determined based on some predefined rules. Visual rhythms are calculated in a zigzag scheme. Different scales were considered, which demonstrated a trade-off between efficiency and efficacy.

Zedan, Elsayed, and Emary (2016) proposed a method for caption detection and localization in videos. The method is based on edge features and the integration of multiple frames. Initially, the edges are computed using the Canny method (Canny 1987), only at the $\frac{1}{3}$ lower part of the frame. Horizontal lines are detected through the number of edge pixels in each row of the frame. The captions localization are determined from an analysis of these horizontal lines. Finally, the frames are clusterized. From the clustering, the authors classified the captions as static and with horizontal or vertical scrolling.

Chen and Su (2018) performed caption localization in videos using visual rhythms. Vertical and horizontal rhythms are extracted from the video. Then, vertical lines in horizontal rhythm and horizontal lines in vertical rhythm are defined using the Sobel filter and the Hough transform. Vertical and horizontal rhythms are extracted in different positions in order to obtain a rhythm that contains a barcode pattern. Localization is also estimated from visual rhythms. Vertical and horizontal projection techniques are finally used to refine the location of captions.

Sravani, Maheswararao, and Murthy (2021) extracted video text using a hybrid method of MSER through morphological filtering. A 2D discrete wavelet transform was employed to remove noise from background and to enhance the text contrast. The method is also combined with traditional text extraction approaches based on edge dependent and connected components to produce better results.

Valery and Jean (2020) developed a method for detecting and localizing embedded subtitles in video streams based on the search for static regions in the video frames. Connected components are extracted from the background via a Gaussian mixture model (GMM), generating binary image masks. Heuristic rules are used to identify the subtitles in the video stream.

Most approaches available in the literature, either in videos or images, localize the scene texts and not caption texts. In this sense, the method proposed in this work can be used to extend such methods to the problem of caption localization in videos. This aspect is interesting due to the scarce development of methods specific to caption texts.

Approaches that address video captioning tasks do not employ a standardized, large-numbered dataset, making it difficult to compare the methods. The dataset proposed in this work can be used to compare the results of the methods to solve these problems. The vast majority of approaches in the literature use their own sets of videos and do not make them publicly available, so it is not possible to compare different methods in the same videos in these cases. For example, Zhang and Chua (2000) employed six television news videos, while Chun et al. (2002) used three long videos (7–59 minutes). Chen and Su (2018) evaluated their method with a set of five videos. Valery and Jean (2020) presented only qualitative results in a small set of videos.

Caption Localization

The method proposed in this work for caption localization applies a scene text localization approach as a baseline, followed by a visual rhythm analysis, which uses the spatio-temporal information of rhythms to maintain only the captions from all scene texts. We use this approach due to the large number of literature methods for the localization of scene texts. The visual rhythm was employed because of its ability to summarize the spatio-temporal information of a video into a single image (or few images), allowing the employment of classic image processing techniques.

Figure 1 presents a diagram with an overview of our method, which has the captioned video as input. Each of the frames has its captions located separately. Then, the horizontal and vertical visual rhythms are constructed from the caption localization masks. Visual rhythms are processed to correct the estimation of caption locations. Finally, we reconstruct the masks across the regions defined by the visual rhythms. The following subsections explain each step of the method.

The core of the Frame Text Localization step can be done by any third-party text localization method (in our case, the method proposed by Epshtein, Ofek, and Wexler (2010)), which can use any technique, such as deep learning. In turn, the other steps use a deterministic algorithm based on classical image processing techniques without any learning-based approach, which tends to have a lower computational cost, in addition to not requiring a training stage.

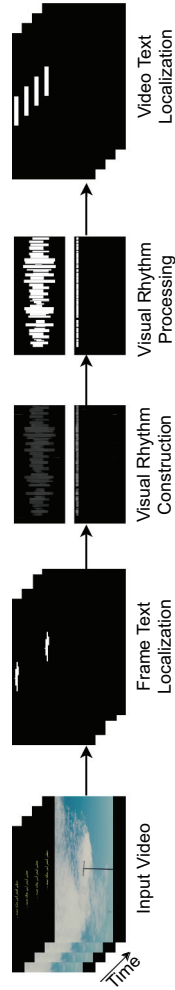


Figure 1. Overview of the proposed video caption localization method based on visual rhythms.

Frame Caption Localization

Initially, we localized the text of each frame independently. Several methods in the literature, such as those discussed in [Section 2](#), could be used in this step. In this work, we apply the method proposed by Epshtein, Ofek, and Wexler (2010), motivated by the quality of the results obtained with this method, and the availability of its code.¹ [Figure 2a](#) presents an example of caption localization obtained through this method.

In our context, we are interested in the regions surrounding the caption. Since the baseline method generates a segmentation result, we perform a post-processing that obtains the caption localization from the segmentation. In addition, since caption texts are often present either at the top or bottom of the frames, we assume that they are not in the central region and we keep only the top and bottom estimates. These estimates are used to generate the mask illustrated in [Figure 2b](#). To do so, a new image of the same size is initialized with zeros. For each row of the localization mask, we identify the first and the last non-black pixels in the corresponding row of the segmentation image. Every pixel between these two (including them) becomes a white pixel in the mask image.



(a) detection through the method by Epshtein et al. (2010)



(b) localization mask

Figure 2. Example of a caption localization mask.

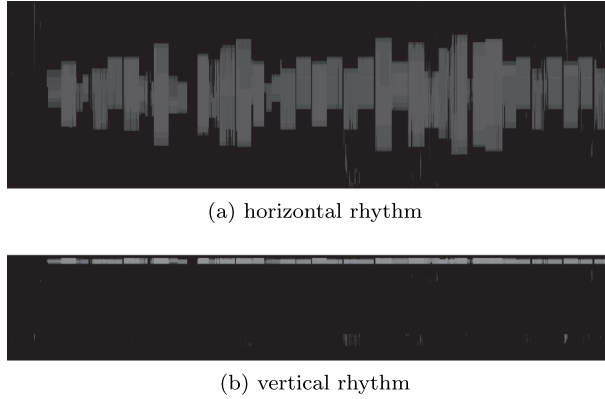


Figure 3. Examples of visual rhythms extracted from mask videos.

Although we use a specific method for this step, any other method that finds text in images could be applied. We conjecture that the final performance may be superior when a better method is used to find text in the frames. Nevertheless, the focus of the method proposed in this work is described in the following subsections, where visual rhythms are employed to refine the localization from temporal information.

Visual Rhythm Construction

After computing the localization mask for all frames, we built the visual rhythms in the vertical and horizontal directions. Visual rhythm Souza et al. (2020) is an image that summarizes information from a video. Inspired by the idea of building visual rhythms from the average of rows and columns (Souza and Pedrini 2020), we use the standard deviation of rows and columns.

Visual rhythms were formally defined by Souza et al. (2020). Let a video be defined as the set $V = \{F_1, F_2, \dots, F_t\}$, where each F_i , $1 \leq i \leq t$ is a $h \times w$ frame represented in matrix form. Let $T(F_i) = S_i$ be an arbitrary operation that maps each F_i into an $n \times 1$ column vector S_i . A visual rhythm (VR) is defined as the $n \times t$ image given by:

$$\text{VR}(V) = [T(F_1)T(F_2) \cdots T(F_t)] = [S_1S_2 \cdots S_t]. \quad (1)$$

In our case, the operation $T(F_i)$ is the standard deviation. To compose the i -th vertical rhythm column, the standard deviation of each row of the mask is calculated for the i -th frame of the video. Similarly, to construct the i -th column of the horizontal rhythm, the standard deviation of each column of the i -th frame mask is calculated. By concatenating the information from each frame into

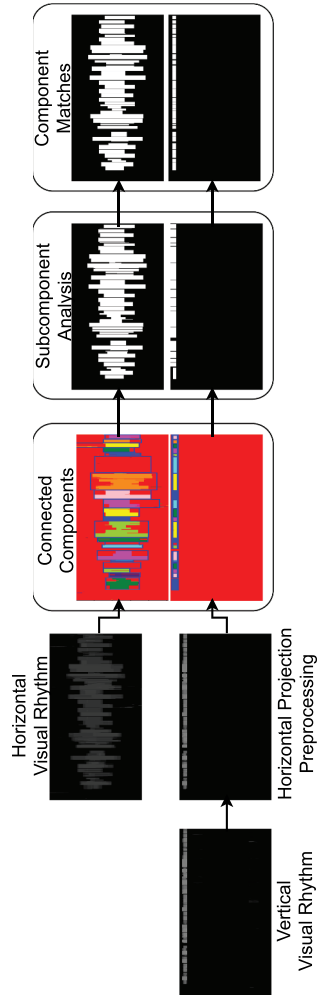


Figure 4. Steps of the visual rhythm processing. Initially, each visual rhythm has its connected components detected. They are then filtered by a subcomponent analysis. Caption positions are retrieved from final visual rhythms.

columns, the vertical rhythm will have dimensions $H \times N$, whereas the horizontal rhythm will have dimensions $W \times N$, where H and W , respectively, correspond to the frame height and width, and N is the number of frames in the video.

The standard deviation was chosen empirically, as we noticed that the average would have problems since the caption usually has a very small size compared to the frame size, which becomes worse in extreme cases. Since the input frame has only values 0 and 1, the standard deviation will give us the information we need: which rows and columns have the most variation. Moreover, horizontal and vertical directions are more suitable for our context (for instance, in comparison with zigzag scheme (Valio, Pedrini, and Leite 2011)) since they match the usual caption direction.

Figure 3 presents examples of visual rhythms, where captions are defined as rectangles. The rectangles are larger at the horizontal visual rhythm because the horizontal dimension of the caption is typically larger than its vertical dimension. We also observed, especially at the horizontal visual rhythm, that the rectangles have discontinuities and are not regular. This is due to text detection failures, which we intend to correct in the next step.

Visual Rhythm Processing

Errors that occur in frame-by-frame text localization can be corrected by incorporating temporal information. From the visual rhythms, we can analyze the temporal information regarding the location of caption from the analysis of adjacent columns. Since we assume that the caption remains fixed in the same position for a certain period of time, it is expected that the corresponding columns should contain uniform rectangles. Thus, in this step we use image processing techniques to make the noisy rectangles from the previous step more uniform.

Figure 4 presents a diagram with the steps of processing visual rhythms. In processing, we consider the binary images of the visual rhythms. To obtain such images, we define as positive any pixel above a constant threshold $T = 10$, empirically chosen.

We preprocess the vertical rhythm based on the horizontal projection as shown in Algorithm 1, in which we determine which rows of vertical visual rhythm will be kept according to an analysis of the projection. In summary, we try to keep consecutive rows with positive pixels (white pixels). This is done in order to calculate the vertical position of the captions, which can be either top or bottom. This preprocessing is done only at the vertical rhythm since the vertical positions of the captions in the frames do not have significant changes throughout the video. In some real-world scenarios, captions may appear at both the top and the bottom, such as in sound effect captions. However, this does not appear in our dataset, and we consider it beyond the scope of our work. Since the caption is predominantly present in a same frame region, false positives can be filtered from the frequency with which estimates appear in a given region. Thus,

we calculate the horizontal projection of the vertical rhythm, which consists of a histogram, where the i -th value is the sum of the pixels of the i -th row. From this histogram, we define a range of rows that represent the location of the captions, so that rhythm rows that fall outside this range are set to null.

Algorithm 1 Vertical rhythm preprocessing.

```

1: procedure HORIZONTAL PROJECTION PREPROCESSING
2: input
3:  $VR_v \leftarrow$  Vertical Rhythm with size  $(H, W)$ 
4:  $p_{max} \leftarrow 0.45$  ▷ Empirical Values
5:  $p_{min} \leftarrow -0.38$ 
6: output
7:  $VR_{P_v} \leftarrow$  Preprocessed Vertical Rhythm with size  $(H, W)$ 
8: begin
9:  $Hist \leftarrow$  an array initialized with  $(H+2)$  zeros
10: for  $i=1$  to  $H+1$  do ▷  $Hist[0]$  and  $Hist[H+1]$  correspond to zero padding
11:  $Hist[i] \leftarrow$  sum of all values of the  $i$ -th rhythm row
12:  $i \leftarrow i+1$ 
13: while  $(Hist[i] - Hist[i-1]) \leq p_{max}$  do ▷  $\Delta[i] \leftarrow Hist[i] - Hist[i-1]$ 
14:  $i \leftarrow i+1$ 
15:  $j \leftarrow H$ 
16: while  $(Hist[j] - Hist[j-1]) \geq p_{min}$  do
17:  $j \leftarrow j-1$ 
18:  $VR_{P_v} \leftarrow$  matrix with size  $(H, W)$  initialized with zeros
19: for  $k$  in  $i+1$  to  $j$  do do
20: for  $l$  in  $0$  to  $W$  do
21:  $VR_{P_v}[k, l] \leftarrow VR_v[k, l]$ 

```

The desired range of the histogram is one that has high values. These values can be represented by one or more nearby peaks. Thus, we calculate the differences in consecutive values, described as

$$\Delta_i = Hist_i - Hist_{i-1} \quad (2)$$

where $Hist_i$ is the i -th value of the histogram. To avoid problems with border values, the zero value is added at the beginning and end of $Hist$, before this calculation.

The histogram rows we are looking for are defined as a range based on an analysis of the Δ values. We assign the first index i as the beginning of the range, where Δ_{i+1} is the first value greater than the p_{max} parameter, and assign the last index j as the end of the range, where Δ_{j-1} the last value that is less than the p_{min} parameter. To define the values of p_{max} and p_{min} , we consider the initial values, chosen empirically, respectively as 0.45 and -0.38 . The final values are chosen separately, checking if there is at least one value in Δ that satisfies these restrictions. If it does not exist for the p_{max} parameter, it will be decremented by 0.01, while for the p_{min} parameter it will be incremented by the same value.

We compute the connected components of the horizontal rhythm and of the preprocessed vertical rhythm. We consider that two pixels belong to the same component if they are connected by an 8-neighborhood and are both positive. Only components that have a minimum width are considered, that is, captions that are in a minimum number of consecutive frames. Since a caption can end in one frame and another start in the next frame, two or more captions can be connected in the same component. [Figure 5a](#) provides an example of this case. Thus, for each of the detected components, we analyze and separate their subcomponents. It is possible to observe that the components are separated by entirely black columns indicating non-captioned frames.

We perform the subcomponent analysis by comparing the component columns. We start with a column set $C = \emptyset$. To consider adding a new column in C , we verify the first and last pixel that there is a positive value in that column. We called these pixels as boundary pixels (superior and inferior). The current column c is added to the set C if at least one of the following conditions are satisfied.

(1) a threshold value is equal or greater than the difference between (a) the position of the boundary pixels in the column c and (b) the averages of the positions of the boundary pixels of all columns currently present in the set C , for at least one of two boundary pixels. In this work, the threshold empirically was chosen as 10 for the horizontal rhythm and 5 for the vertical rhythm;

(2) at least 70% of the next $k = 5$ columns have at least one of two boundary pixels close enough to the average to be part of the set C ;

(3) c is the last column of this component.

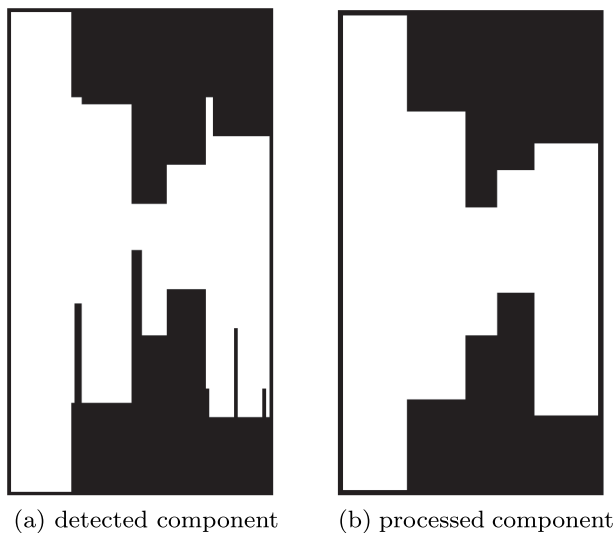


Figure 5. Example of detected and processed components.

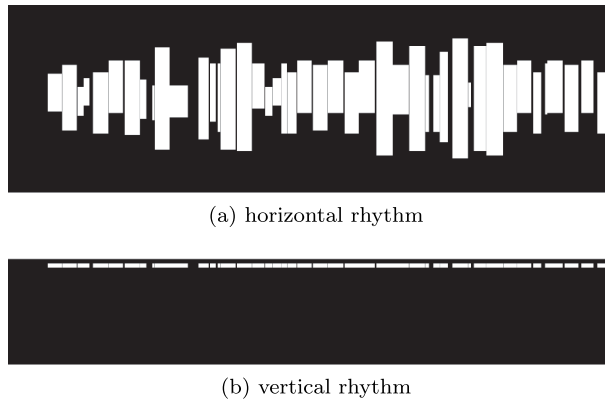


Figure 6. Examples of visual rhythms after processing.

When none of the previous conditions are valid, the end of a subcomponent is determined. This subcomponent starts at the first and ends at the last column added to C . Assuming that the frame text localization method generally gets the right result, the upper and lower bounds given by the first and last rows in which positive pixels exist are computed, respectively, as the mode of the first and last rows of the columns in the set C . Then, if the current column is not the last column of the component, it will be added to the set C and the process continues to the last column. The result of processing a component from subcomponents can be seen in [Figure 5b](#), where we can observe that the irregularities present in the component have been corrected.

Irregularities within a component, as shown in [Figure 5a](#), mean that the location of a specific caption changed over the seconds. However, this contradicts our premise that each caption that appears throughout the video is fixed in the same position and with the same delimitation. If two consecutive subcomponents have close values for both the first and last rows, they are joined so that all columns between them are part of a single component. Finally, to reduce the number of possible failures, we check the correspondence of the localized components of two rhythms, so as to maintain only the intersections of the components of both rhythms. [Figure 6](#) presents the visual rhythms obtained at the end of the process.

Video Caption Localization

As we use the rhythms in the vertical and horizontal directions, we can retrieve the caption localization from the positive pixel coordinates. The positive pixels in the i -th column of the vertical rhythm indicate the rows of the i -th frame

Table 1. Information from videos collected from YouTube to compose the dataset. All videos were tagged with creative commons license.

Video	FPS	Resolution (pixels)	# Frames	Content
1	25	1024 × 576	3309	Music Video
2	25	1024 × 576	3907	Classroom
3	25	1280 × 720	1217	Vlog
4	25	1024 × 576	1848	Animation
5	30	1280 × 738	4897	Animation
6	25	1024 × 576	12287	Homemade Video
7	25	1024 × 576	3423	Sports
8	25	1024 × 576	4005	Sports
9	25	1024 × 576	18840	Lecture
10	25	1024 × 576	1365	Amateur Movie
11	25	1024 × 576	2753	Factory
12	30	1920 × 1080	4810	Open Environment
13	30	1920 × 1080	6015	Vlog
14	24	1920 × 1080	5670	Vlog
15	25	1024 × 576	5866	Vlog
16	25	1024 × 576	4857	Vlog
17	24	1920 × 1080	1720	Vlog

that should be positive. That is, if the j -th pixel of the i -th column is active at the vertical rhythm, the j -th row of the mask of the i -th frame must become active. Similarly, positive horizontal rhythm pixels indicate which columns should be active in the mask. Masks are constructed so that a given pixel is active only if its row and column are active.

Dataset

There is a lack of public data, both in terms of quantity and quality, to evaluate related literature approaches. Thus, we propose a dataset that contains videos with caption and the necessary information for its detection, localization, segmentation, script recognition and text recognition.

To create the dataset, seventeen videos were collected from YouTube. Three criteria were established for a video to be inserted into the dataset: (i) be marked as a Creative Commons license, which gives the right to reuse and edit the video to anyone; (ii) have separate caption in raw text, providing a subtitle file with the format.srt; (iii) has no or little text already embedded in the video.

Preprocessing was done to deal with cases where the video had few frames with embedded text, leaving only frames without embedded text. Thus, it was possible to create a dataset if we had information related to the location of captions in all frames of the video. [Table 1](#) presents information for each of the collected videos. It is possible to observe that the videos were taken at different frame rates per second (FPS), at high resolution and with different durations.

[Figure 7](#) presents a frame of each of the videos. Videos #1, #7, #8, #10 and #12 feature plenty of camera shake and cuts in different scenarios, such as music and sports clips. Videos #2, #3, #13, #14, #15, #16 and #17 have either a fixed camera or slow motion. Videos #4 and #5 are animations. Some other scenarios were also considered, such as indoor environments in videos #6 and #11 and slightly darker surroundings as in video #9.

To add the captions to the videos, different characteristics were considered, namely: position, size, color and script. [Figure 8](#) illustrates the process of inserting captions and creating ground truth. Each video and script pair results in a captioned video. As thirteen scripts were considered, 221 captioned videos were obtained.

For each captioned video, the `ffmpeg` tool was used to insert the caption into the video. Position, color, and size of captions are randomly chosen, all with equal probability, within predefined options. The settings were selected to include the most common conditions found in real cases. Captions were positioned either below or above in the video frames. Font sizes were chosen from small, medium and large, respectively, 12, 24 and 36. White, yellow, and red colors were considered for captions.

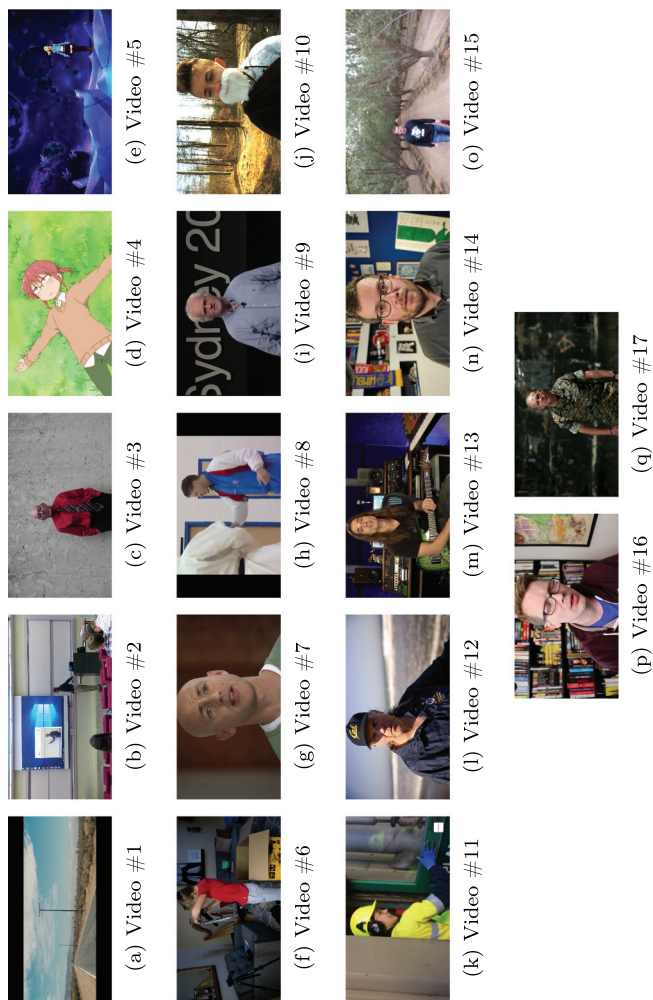


Figure 7. Frames from each of the videos collected from YouTube.

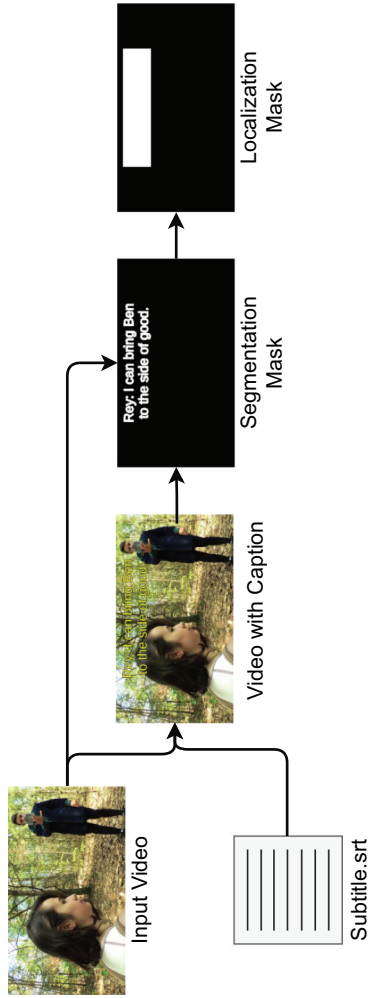


Figure 8. Representation of text insertion in video and creation of ground-truth information.

After inserting captions on videos that did not have embedded text, we can calculate the mask for caption segmentation from the differences between the frames of the original video I and the frames with the added text J , expressed as

$$D(i, j, t) = I(i, j, t) - J(i, j, t) \quad (3)$$

where t indexes video frames over time. In addition, i and j index, respectively, the rows and columns of the frame. Pixel (i, j) is active in segmentation mask of the t -th frame if, and only if, $D(i, j, t)$ is nonzero. If there is at least one nonzero value in $D(t)$, there will be a caption in the t -th frame. By analyzing D rows and columns where there is at least one non-zero value, we can calculate the rectangle surrounding the segmentation mask obtaining the location mask.

As mentioned previously, thirteen different scripts were considered. For this, each of the captions, originally in English, were translated into each of the scripts, presented in Table 2. The scripts adopted are of different types and emerged at different times. For each script, we consider its official language. In the case of Roman, English language was used. Figure 9 illustrates the same sentence in different scripts. The used scripts have substantial variation, which can hamper the development of a system that supports them all. In addition, there is a great similarity between some of the scripts, especially those of the same type, which can make their automatic recognition a hard task.

Figure 10 shows statistics about captions added to the videos. The graphics illustrate the number of videos in which captions were added with a certain color, size or position, relative to the seventeen videos collected (Figure 10(a,b,c)) or the thirteen scripts (Figure 10(d,e,f)). From these statistics, it is possible to observe the dataset profile. For example, for the Urdu script, the added texts were distributed almost equally over position and color. However, few large texts (36) were added. SubFigure 10g shows the number of frames with and without text per video, where it is possible to see which videos have text on most of their frames. Video #6 has the largest number of frames without text, whereas video #4 is the video with the highest percentage of videos with text.

Table 2. Different scripts considered in this work.

Name	Type	Direction	Time
Arabic	Abjad	Right-to-Left	400 CE
Bangla	Abugida	Left-to-Right	1000 CE
Chinese	Logographic	Left-to-Right	3300 BCE
Roman	Alphabet	Left-to-Right	700 BCE
Greek	Alphabet	Left-to-Right	800 BCE
Japanese	Logographic and Syllabic	Varies	300 CE
Kannada	Abugida	Left-to-Right	400 CE
Malayalam	Abugida	Left-to-Right	830 CE
Persian	Abjad	Right-to-Left	800 CE
Russian	Alphabet	Left-to-Right	890 CE
Tamil	Abugida	Left-to-Right	700 CE
Telugu	Abugida	Left-to-Right	900 CE
Urdu	Abjad	Right-to-Left	1200 CE

في مكان ما ، شيء لا يصدق ينتظر أن يكون معروفا.

(a) Arabic

কোথাও, অবিশ্বাস্য কিছু পরিচিত হতে অপেক্ষা করছে।

(b) Bangla

在某個地方，令人難以置信的東西等待著名。

(c) Chinese

Somewhere, something incredible is waiting to be known.

(d) Roman

Κάπου, κάτι απίστευτο περιμένει να γίνει γνωστό.

(e) Greek

どこかで、信じられないほどの何か知られるのを待っています。

(f) Japanese

ಎಲ್ಲೋ, ನಂಬಲಾಗದ ಏನೋ ತಿಳಿದಿರುವುದು ಕಾಯುತ್ತಿದೆ.

(g) Kannada

എവിടെയോ, അവിശ്വസനീയമായ എന്തോ അറിയാൻ കാത്തിരിക്കുകയാണ്.

(h) Malayalam

در جایی، چیزی باورنکردنی منتظر شناخته شدن است.

(i) Persian

Где-то, что-то невероятное ждет, чтобы стать известным.

(j) Russian

எங்காவது, நம்பமுடியாத ஒன்று அறியப்பட காத்திருக்கிறது.

(k) Tamil

ఎక్కడా, అద్భుతమైన ఏదో తెలుసుకోవచ్చు వేచి ఉంది.

(l) Telugu

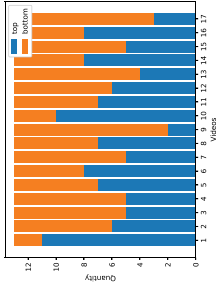
کہیں، کچھ ناقابل یقین معلوم ہونے کا انتظار ہے۔

(m) Urdu

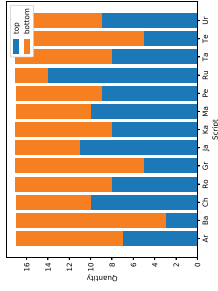
Figure 9. Same sentence for different scripts considered.

Evaluation Metrics

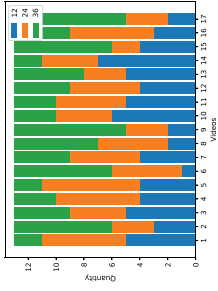
Although we propose a method for caption localization without an earlier step for explicit frame caption detection, we evaluated which video frames a caption was localized and compared it with detection methods available in the literature using the proposed dataset.



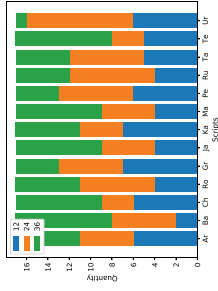
(c) position per video



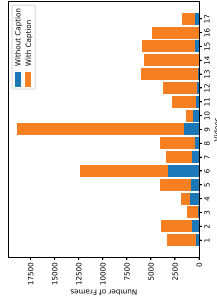
(f) position per script



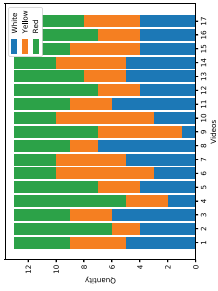
(b) size per video



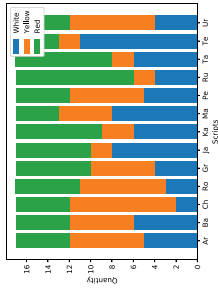
(e) size per script



(g) frames per video



(a) color per video



(d) color per script

Figure 10. Statistics for the built dataset.

Let TP, FP and FN be true positive, false positive, and false negative, respectively. In the detection problem, positive indicates a frame with a caption. Precision and recall metrics are used for caption detection evaluation and can be described respectively as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

Precision evaluates the number of frames that have captions over all the frames that have captions. A lower precision value indicates that fewer frames estimated as captioned frames are incorrect. Recall evaluates the number of frames that are estimated to have captions, within all frames that have captions. A lower recall value indicates that fewer captioned frames were estimated.

The ratio between the intersection area and the union area of the estimation of caption location and the mask can be used for evaluating the caption localization, expressed as

$$\text{IoU} = \frac{\text{area}(A \cap B)}{\text{area}(A \cup B)} \quad (6)$$

where A is the estimate and B the mask of the caption location. Values close to zero are obtained when the estimate differs from the mask, either by size or location, whereas values close to one occur when the estimate and mask are similar. From the intersection over the union, we define the accuracy used in localization as

$$\text{Accuracy} = \frac{\sum_{i=0}^N t_i}{N} \quad (7)$$

where N is the total number of frames in the video, and

$$t_i = \begin{cases} 1 & , \text{if } \text{IoU}(A_i, B_i) > 0.5 \\ 0 & , \text{otherwise} \end{cases} \quad (8)$$

Experiments and Discussions

This section presents the experiments conducted on our dataset with the method proposed in this work and different literature approaches. [Subsection 5.1](#) presents the results obtained for the detection of frames with caption. [Subsection 5.2](#) describes and discusses the results obtained for the caption localization task. We

compared the results of our method to those proposed by Valio, Pedrini, and Leite (2011) and Epshtein, Ofek, and Wexler (2010). The method proposed by Valio, Pedrini, and Leite (2011) performs caption detection, while the one proposed by Epshtein, Ofek, and Wexler (2010) performs text localization. The experiments presented in [Subsection 5.1](#) compare our method with them. These experiments are preliminary in relation to those presented in [Subsection 5.2](#) and show an aspect in which our method can be improved in future work. These methods were chosen due to their good results and code availability. It is important to highlight the difficulty in making a comparison with other methods in the literature, because, in addition to the lack of available codes, the datasets were not, until this work, standardized for these tasks, in such a way that each literature work used its own set of a few videos.

Caption Frame Detection

The following experiments aim to present the results of frame detection methods in the proposed dataset, in addition to verifying and comparing the results obtained by determining which frames had a caption found by the localization methods. In order to do so, we analyzed the information of the generated masks, in such a way that it was possible to make a comparison with the method proposed by Valio, Pedrini, and Leite (2011). The results obtained with our method and with the approach developed by Epshtein, Ofek, and Wexler (2010) are calculated by assigning true when the sum of the caption mask is greater than zero, and false otherwise. The experiments of this subsection refer only to detection of frames with captions, which is important to eventually determine the methods drawbacks. By improving the quality of the methods at this step, the final results (localization) also tend to be improved.

[Table 3](#) presents the average results obtained for each source video. For the method proposed by Valio, Pedrini, and Leite (2011), the recall values obtained are above 90% for almost all videos, which shows that this method rarely erroneously disregards frames with a caption. In terms of precision, most videos have values above 90%, however, videos #4, #6 and #10 have low values, with 49.4% in the worst case.

The results of the method proposed by Epshtein, Ofek, and Wexler (2010) show a better balance between recall and precision, with recall values above 90% for almost all videos, and precision below 80% only for video #10. For our method, we can observe that the precision values were above 90% for almost all videos, outperforming the low results of the method proposed by Valio, Pedrini, and Leite (2011). The precision value for video #10 was also higher than the value of the method proposed by Epshtein, Ofek, and Wexler (2010). On the other hand, recall values are lower, especially for videos #5, #6, #7 and #11. This shows that our method may have trouble finding captions in some frames, but almost does not erroneously find captions in frames that do not



Table 3. Results obtained for detecting frames with captions for the different videos in the dataset.

Video	Valio, Pedrini, and Leite (2011)		Epshtein, Ofek, and Wexler (2010)		Our Method	
	Precision	Recall	Precision	Recall	Precision	Recall
1	0.957 ± 0.01	0.936 ± 0.16	0.996 ± 0.00	0.990 ± 0.01	0.997 ± 0.00	0.947 ± 0.02
2	0.836 ± 0.00	0.996 ± 0.00	0.915 ± 0.00	0.931 ± 0.06	0.958 ± 0.00	0.813 ± 0.08
3	0.961 ± 0.05	0.970 ± 0.10	0.960 ± 0.00	0.999 ± 0.00	0.976 ± 0.00	0.972 ± 0.01
4	0.494 ± 0.00	0.997 ± 0.00	0.808 ± 0.00	0.975 ± 0.03	0.903 ± 0.01	0.846 ± 0.07
5	0.872 ± 0.02	0.896 ± 0.16	0.947 ± 0.00	0.941 ± 0.05	0.975 ± 0.00	0.719 ± 0.12
6	0.740 ± 0.00	0.994 ± 0.01	0.834 ± 0.01	0.919 ± 0.07	0.916 ± 0.04	0.600 ± 0.19
7	0.889 ± 0.00	0.979 ± 0.04	0.935 ± 0.00	0.939 ± 0.05	0.972 ± 0.00	0.735 ± 0.10
8	0.902 ± 0.00	0.976 ± 0.08	0.984 ± 0.00	0.969 ± 0.02	0.998 ± 0.00	0.815 ± 0.08
9	0.995 ± 0.00	0.885 ± 0.27	0.973 ± 0.00	0.957 ± 0.04	0.976 ± 0.00	0.848 ± 0.10
10	0.599 ± 0.02	0.947 ± 0.09	0.647 ± 0.00	0.998 ± 0.00	0.707 ± 0.01	0.888 ± 0.04
11	0.874 ± 0.00	1.000 ± 0.00	0.902 ± 0.00	0.996 ± 0.00	0.917 ± 0.03	0.786 ± 0.18
12	0.943 ± 0.00	0.950 ± 0.09	0.982 ± 0.00	0.969 ± 0.03	0.993 ± 0.00	0.814 ± 0.10
13	0.983 ± 0.00	1.000 ± 0.00	0.984 ± 0.00	0.999 ± 0.00	1.000 ± 0.00	0.931 ± 0.10
14	0.999 ± 0.00	1.000 ± 0.00	1.000 ± 0.00	0.999 ± 0.00	1.000 ± 0.00	0.931 ± 0.10
15	0.924 ± 0.00	1.000 ± 0.00	0.987 ± 0.00	0.980 ± 0.02	0.995 ± 0.00	0.870 ± 0.07
16	1.000 ± 0.00	0.988 ± 0.01	1.000 ± 0.00	0.988 ± 0.02	1.000 ± 0.00	0.837 ± 0.14
17	0.801 ± 0.01	0.981 ± 0.06	0.953 ± 0.00	0.972 ± 0.04	0.992 ± 0.00	0.870 ± 0.12
Average	0.868 ± 0.14	0.970 ± 0.03	0.929 ± 0.09	0.971 ± 0.02	0.957 ± 0.07	0.836 ± 0.09

Table 4. Results obtained for detecting frames with captions for different scripts.

Script	Valio, Pedrini, and Leite (2011)		Epshtein, Ofek, and Wexler (2010)		Our Method	
	Precision	Recall	Precision	Recall	Precision	Recall
Arabic	0.870 ± 0.13	0.979 ± 0.05	0.926 ± 0.09	0.930 ± 0.07	0.947 ± 0.08	0.742 ± 0.19
Bangla	0.866 ± 0.14	0.963 ± 0.07	0.930 ± 0.08	0.969 ± 0.03	0.958 ± 0.06	0.772 ± 0.15
Chinese	0.863 ± 0.14	0.950 ± 0.11	0.930 ± 0.09	0.980 ± 0.03	0.958 ± 0.06	0.899 ± 0.09
Roman	0.871 ± 0.13	0.981 ± 0.05	0.931 ± 0.08	0.994 ± 0.01	0.960 ± 0.06	0.877 ± 0.12
Greek	0.872 ± 0.13	0.997 ± 0.00	0.930 ± 0.09	0.985 ± 0.03	0.962 ± 0.06	0.909 ± 0.08
Japanese	0.871 ± 0.13	0.983 ± 0.06	0.930 ± 0.08	0.981 ± 0.03	0.955 ± 0.07	0.900 ± 0.09
Kannada	0.866 ± 0.13	0.885 ± 0.26	0.930 ± 0.08	0.969 ± 0.03	0.957 ± 0.07	0.800 ± 0.13
Malayalam	0.868 ± 0.13	0.942 ± 0.16	0.930 ± 0.08	0.976 ± 0.04	0.957 ± 0.06	0.823 ± 0.15
Persian	0.872 ± 0.13	0.999 ± 0.00	0.930 ± 0.08	0.965 ± 0.03	0.960 ± 0.06	0.814 ± 0.11
Russian	0.866 ± 0.14	0.968 ± 0.08	0.931 ± 0.08	0.993 ± 0.00	0.956 ± 0.07	0.829 ± 0.13
Tamil	0.872 ± 0.13	0.997 ± 0.00	0.932 ± 0.08	0.995 ± 0.00	0.961 ± 0.06	0.881 ± 0.07
Telugu	0.868 ± 0.13	0.981 ± 0.05	0.928 ± 0.09	0.957 ± 0.05	0.950 ± 0.07	0.799 ± 0.15
Urdu	0.871 ± 0.13	0.987 ± 0.03	0.928 ± 0.09	0.942 ± 0.07	0.956 ± 0.07	0.815 ± 0.12
Average	0.868 ± 0.00	0.970 ± 0.03	0.929 ± 0.00	0.972 ± 0.01	0.956 ± 0.00	0.835 ± 0.05

Table 5. Results obtained for detecting frames with captions for different caption characteristics.

		Valio, Pedrini, and Leite (2011)		Epshtein, Ofek, and Wexler (2010)		Our Method	
		Precision	Recall	Precision	Recall	Precision	Recall
Size	12	0.866 ± 0.14	0.908 ± 0.17	0.925 ± 0.10	0.965 ± 0.05	0.949 ± 0.08	0.848 ± 0.14
	24	0.864 ± 0.14	0.994 ± 0.03	0.928 ± 0.08	0.980 ± 0.03	0.957 ± 0.06	0.842 ± 0.12
	36	0.875 ± 0.12	0.999 ± 0.00	0.935 ± 0.07	0.970 ± 0.04	0.963 ± 0.06	0.818 ± 0.15
Position	Top	0.863 ± 0.14	0.976 ± 0.08	0.920 ± 0.10	0.973 ± 0.04	0.949 ± 0.08	0.848 ± 0.14
	Bottom	0.874 ± 0.13	0.965 ± 0.12	0.939 ± 0.07	0.971 ± 0.04	0.964 ± 0.05	0.824 ± 0.13
Color	White	0.891 ± 0.11	0.999 ± 0.00	0.942 ± 0.07	0.980 ± 0.02	0.966 ± 0.06	0.852 ± 0.11
	Yellow	0.859 ± 0.14	0.982 ± 0.07	0.914 ± 0.10	0.966 ± 0.05	0.945 ± 0.08	0.833 ± 0.15
	Red	0.857 ± 0.15	0.935 ± 0.15	0.932 ± 0.08	0.970 ± 0.05	0.958 ± 0.06	0.822 ± 0.14

have them. This result also indicates that our method removes true-positive frames from the estimation, which affects its final results. Possibly, the poor performance of all methods in the video #10 may be explained by their high rate of smaller and yellow captions located mainly at the top, where the colors of objects in the scene, such as yellowed dry leaves, can cause methods to confuse captions with the background.

Table 4 presents the average results calculated for each script. We can observe that the values obtained for the different scripts are very close for all the metrics to the results obtained by Valio, Pedrini, and Leite (2011). This was expected since this method considers information independent of the characters used, which shows script invariance. Similar results can be seen for different caption font characteristics, presented in Table 5.

The method proposed by Epshtein, Ofek, and Wexler (2010) shows superior results for precision rate, with recall above 90%. For our method, high precision was obtained, but in some scripts, such as Arabic, Bangla, and Telugu, we achieved lower recall values. Similarly to the results obtained by Valio, Pedrini,

Table 6. Average accuracy achieved for video caption localization.

Video	Our Method	Epshtein, Ofek, and Wexler (2010)	Our Method Correct Frames	Epshtein, Ofek, and Wexler (2010) Correct Frames
1	0.774 ± 0.15	0.797 ± 0.24	0.791 ± 0.16	0.783 ± 0.26
2	0.481 ± 0.15	0.337 ± 0.18	0.458 ± 0.20	0.266 ± 0.21
3	0.780 ± 0.17	0.671 ± 0.29	0.791 ± 0.18	0.662 ± 0.30
4	0.584 ± 0.07	0.501 ± 0.06	0.191 ± 0.16	0.121 ± 0.12
5	0.522 ± 0.16	0.530 ± 0.15	0.536 ± 0.21	0.449 ± 0.18
6	0.454 ± 0.16	0.300 ± 0.16	0.409 ± 0.25	0.204 ± 0.18
7	0.636 ± 0.13	0.538 ± 0.14	0.722 ± 0.15	0.463 ± 0.17
8	0.582 ± 0.19	0.535 ± 0.25	0.637 ± 0.21	0.496 ± 0.28
9	0.634 ± 0.15	0.495 ± 0.18	0.710 ± 0.14	0.481 ± 0.18
10	0.427 ± 0.10	0.343 ± 0.07	0.222 ± 0.14	0.196 ± 0.08
11	0.465 ± 0.25	0.441 ± 0.21	0.485 ± 0.30	0.424 ± 0.22
12	0.554 ± 0.13	0.569 ± 0.19	0.647 ± 0.13	0.563 ± 0.19
13	0.466 ± 0.29	0.476 ± 0.19	0.504 ± 0.30	0.476 ± 0.19
14	0.651 ± 0.31	0.758 ± 0.38	0.702 ± 0.33	0.758 ± 0.38
15	0.515 ± 0.27	0.451 ± 0.31	0.538 ± 0.31	0.421 ± 0.33
16	0.503 ± 0.23	0.441 ± 0.32	0.609 ± 0.24	0.444 ± 0.31
17	0.679 ± 0.17	0.659 ± 0.20	0.645 ± 0.21	0.575 ± 0.25
Average	0.571 ± 0.10	0.520 ± 0.13	0.564 ± 0.17	0.457 ± 0.18

and Leite (2011) and Epshtein, Ofek, and Wexler (2010), the values obtained with our method showed invariance for different characteristics of the caption font. These results can be seen in Table 5.

Caption Localization in Frames

In the following experiments, we present the final results for the caption localization. Table 6 presents the average accuracy for each video for our method and the method based on the stroke width operator proposed by Epshtein, Ofek, and Wexler (2010). Additionally, the accuracy rate was also computed only in frames where there was a caption according to the ground truth (correct frames). This was done so that we could also evaluate the results with a lower impact of the errors on non-caption frames by isolating the

Table 7. Average accuracy obtained for caption location for each script.

Script	Our Method	Epshtein, Ofek, and Wexler (2010)	Our Method w/ Correct Frames	Epshtein, Ofek, and Wexler (2010) Correct Frames
Arabic	0.543 ± 0.20	0.426 ± 0.22	0.560 ± 0.27	0.360 ± 0.26
Bangla	0.440 ± 0.18	0.328 ± 0.19	0.419 ± 0.27	0.249 ± 0.21
Chinese	0.728 ± 0.21	0.699 ± 0.22	0.718 ± 0.29	0.654 ± 0.27
Roman	0.640 ± 0.22	0.642 ± 0.25	0.620 ± 0.27	0.587 ± 0.29
Greek	0.693 ± 0.16	0.690 ± 0.22	0.665 ± 0.25	0.637 ± 0.28
Japanese	0.711 ± 0.21	0.711 ± 0.23	0.703 ± 0.28	0.667 ± 0.29
Kannada	0.474 ± 0.20	0.465 ± 0.24	0.485 ± 0.24	0.401 ± 0.26
Malayalam	0.479 ± 0.19	0.376 ± 0.23	0.469 ± 0.23	0.303 ± 0.25
Persian	0.565 ± 0.12	0.476 ± 0.21	0.566 ± 0.23	0.406 ± 0.26
Russian	0.600 ± 0.25	0.593 ± 0.28	0.611 ± 0.34	0.536 ± 0.31
Tamil	0.569 ± 0.20	0.459 ± 0.26	0.550 ± 0.26	0.386 ± 0.28
Telugu	0.466 ± 0.20	0.430 ± 0.21	0.470 ± 0.26	0.370 ± 0.23
Urdu	0.523 ± 0.21	0.471 ± 0.20	0.523 ± 0.25	0.407 ± 0.23
Average	0.571 ± 0.09	0.520 ± 0.13	0.566 ± 0.09	0.458 ± 0.14

Table 8. Average accuracy obtained for caption localization with different characteristics.

		Our Method	Epshtein, Ofek, and Wexler (2010)	Our Method Correct Frames	Epshtein, Ofek, and Wexler (2010) Correct Frames
Size	12	0.602 ± 0.23	0.594 ± 0.25	0.583 ± 0.31	0.541 ± 0.30
	24	0.605 ± 0.18	0.601 ± 0.21	0.614 ± 0.24	0.538 ± 0.26
	36	0.514 ± 0.23	0.381 ± 0.25	0.505 ± 0.28	0.313 ± 0.28
Position	Top	0.572 ± 0.23	0.530 ± 0.28	0.554 ± 0.30	0.470 ± 0.31
	Bottom	0.571 ± 0.21	0.513 ± 0.24	0.578 ± 0.26	0.449 ± 0.28
Color	White	0.568 ± 0.22	0.527 ± 0.26	0.579 ± 0.26	0.479 ± 0.29
	Yellow	0.587 ± 0.23	0.522 ± 0.28	0.578 ± 0.29	0.466 ± 0.31
	Red	0.562 ± 0.21	0.515 ± 0.24	0.544 ± 0.28	0.436 ± 0.29

assessment of error location in false positive detection. In this case, we do not modify our method to take into account only the correct frames, but only calculate the evaluation metrics in those frames.



Figure 11. Results for frames with scene and caption text.

For twelve of the seventeen videos, our method had better accuracy, with the most noticeable improvement for videos #2, #6 and #9. On the other hand, the results of the other five videos were slightly lower, with negative highlighting for video #14.

For results considering only correct frames, there was an improvement for all videos, except video #14, with the most evident improvement on videos #2, #6, #7 and #9. These results show that our method achieved better results than the method proposed by Epshtein, Ofek, and Wexler (2010) mainly due to the improved localization of captions in frames that had already been detected. However, the final result is not as good as it could be, because our method erroneously disregarded some true positives, such as video #7, which has a low recall on detection, as shown in Table 3.

Table 7 presents the average accuracy in caption localization for each script. Our method obtained better results than those achieved by Epshtein, Ofek, and Wexler (2010) for all scripts, especially Arabic, Bangla, Malayalam, Persian, and Tamil, where their method obtained low accuracy, probably due to the character formats of these scripts. Since our method considers temporal information, without analyzing the character format, we possibly solved problems that occurred in the caption localization due to the nature of the script.

Table 8 shows the average accuracy in caption localization for different font characteristics. Regarding the font size, our method was superior for larger fonts, maintaining good results for videos with smaller fonts, while both methods were invariant to position and color of the captions. From the results obtained, we

believe that the method by Epshtein, Ofek, and Wexler (2010) had problems in dealing with very large fonts, and this reflected the better performance of our method.

Figure 11 presents examples of results for frames where scene texts and captions are shown together. In an overview, our method performed a good caption localization (despite some misalignments) and ignored scene texts. In the first case, the scene text is present in the center of the frame, a region that is disregarded by our method. In the second example, the scene text appears on the opposite side of the captions. The success of our method, in this case, is associated with considering only captions in the same region of the frame, and the scene text is eliminated through temporal analysis. The third example also uses temporal analysis to succeed. In this case, the caption and scene texts are presented in the same region and the method separated them. Finally, the last example shows that our method is also capable of disregarding large scene texts.

Conclusions

The contributions of this paper are twofold. We presented a novel method for localizing video captions using visual rhythms for temporal analysis of frame-by-frame location estimates. A new dataset was created with seventeen initial videos. For each video, we added captions with thirteen distinct scripts to obtain a total of two hundred and twenty one videos with ground truth.

Experiments were performed on the proposed dataset to compare our results against two other methods. It is worth mentioning that a more extensive comparison would be a very difficult task due to the lack of available codes and annotated public datasets. In this sense, the dataset built in this work may provide an opportunity for other authors to evaluate their methods. Experimental results demonstrated that our method can considerably improve results for most videos, with improvement being evident in some specific videos. However, our method could still be improved in order to reduce the great number of false positives, which would mitigate the obtained results.

As directions for future work, we intend to evaluate the results of our method considering frame-by-frame localization made by a deep machine learning approach, such as convolutional neural networks. We will also improve the comparative analysis of the literature on the dataset proposed in this paper.

Note

1. <https://github.com/aperrau/DetectText/>

Acknowledgments

The authors are thankful to São Paulo Research Foundation (FAPESP grant #2017/12646-3), National Council for Scientific and Technological Development (CNPq grant #309330/2018-1), Coordination for the Improvement of Higher Education Personnel (CAPES) and Minas Gerais Research Foundation (FAPEMIG) for their financial support.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by São Paulo Research Foundation (FAPESP grant #2017/12646-3).

ORCID

Marcos Roberto e Souza  <http://orcid.org/0000-0003-4342-5220>

Helena de Almeida Maia  <http://orcid.org/0000-0002-8253-9004>

Anderson Carlos Souza e Santos  <http://orcid.org/0000-0002-7806-3410>

Marcelo Bernardes Vieira  <http://orcid.org/0000-0003-3356-6679>

Helio Pedrini  <http://orcid.org/0000-0003-0125-630X>

References

- Agnihotri, L., and N. Dimitrova (1999). Text detection for video analysis. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, Fort Collins, CO, USA, 109–13. IEEE.
- Arafat, S. Y., and M. J. Iqbal. 2020. Urdu-text detection and recognition in natural scene images using deep learning. *IEEE Access* 8:96787–803. doi:10.1109/ACCESS.2020.2994214.
- Canny, J. 1987. A computational approach to edge detection. In *Readings in Computer Vision*, 184–203. San Francisco, CA, USA: Elsevier.
- Chen, L.-H., and C.-W. Su. 2018. Video caption extraction using spatio-temporal slices. *International Journal of Image and Graphics* 18 (2):1850009. doi:10.1142/S0219467818500092.
- Chun, S. S., H. Kim, K. Jung-Rim, S. Oh, and S. Sull (2002). Fast text caption localization on video using visual rhythm. In *International Conference on Advances in Visual Information Systems*, Hsin Chu, Taiwan, 259–68. Springer.
- Concha, D. T., H. A. Maia, H. Pedrini, H. Tacon, A. S. Brito, H. L. Chaves, and M. B. Vieira (2018). Multi-stream convolutional neural networks for action recognition in video sequences based on adaptive visual rhythms. In *17th IEEE International Conference on Machine Learning and Applications*, Orlando, FL, USA, 473–80. IEEE.
- Epshtein, B., E. Ofek, and Y. Wexler (2010). Detecting text in natural scenes with stroke width transform. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2963–70. IEEE.
- He, D., X. Yang, C. Liang, Z. Zhou, A. G. Ororbi, D. Kifer, and C. Lee Giles (2017a). Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 3519–28.

- He, P., W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li (2017b). Single shot text detector with regional attention. In *IEEE International Conference on Computer Vision*, Venice, Italy, 3047–55.
- Jiang, D., S. Zhang, Y. Huang, Q. Zou, X. Zhang, M. Pu, and J. Liu. 2020. Detecting dense text in natural images. *IET Computer Vision* 14 (8):597–604. doi:10.1049/iet-cvi.2019.0916.
- Jiang, Y., X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo (2017). R2CNN: Rotational region CNN for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*.
- Katper, S. H., A. R. Gilal, A. Waqas, A. Alshantqiti, A. Alsughayyir, and J. Jaafar. 2020. Deep neural networks combined with STN for multi-oriented text detection and recognition. *International Journal of Advanced Computer Science and Applications* 11 (4):178–85. doi:10.14569/IJACSA.2020.0110424.
- Khare, V., P. Shivakumara, and P. Raveendran. 2015. A new histogram oriented moments descriptor for multi-oriented moving text detection in video. *Expert Systems with Applications* 42 (21):7627–40. doi:10.1016/j.eswa.2015.06.002.
- Lee, -C.-C., Y.-C. Chiang, H.-M. Huang, and C.-L. Tsai (2007). A fast caption localization and detection for news videos. In *Second International Conference on Innovative Computing, Information and Control*, Kumamoto, Japan, 226–226. IEEE.
- Liao, M., Z. Zhu, B. Shi, G.-S. Xia, and X. Bai (2018). Rotation-sensitive regression for oriented scene text detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 5909–18.
- Liu, Y., and L. Jin (2017). Deep matching prior network: Toward tighter multi-oriented text detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 1962–69.
- Long, S., X. He, and C. Ya (2018). Scene text detection and recognition: The deep learning era. *arXiv preprint arXiv:1811.04256*.
- Lyu, M. R., J. Song, and M. Cai. 2005. A comprehensive method for multilingual video text detection, localization, and extraction. *IEEE Transactions on Circuits and Systems for Video Technology* 15 (2):243–55. doi:10.1109/TCSVT.2004.841653.
- Moreira, T. P., D. Menotti, and H. Pedrini (2017). First-person action recognition through visual rhythm texture description. In *International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, USA, 2627–31. IEEE.
- Neumann, L., and J. Matas (2010). A method for text localization and recognition in real-world images. In *Asian Conference on Computer Vision*, Queenstown, New Zealand, 770–83. Springer.
- Pinto, A., H. Pedrini, W. Schwartz, and A. Rocha (2012). Video-based face spoofing detection through visual rhythm analysis. In *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, Ouro Preto, MG, Brazil, 221–28. IEEE.
- Pinto, A., W. R. Schwartz, H. Pedrini, and A. Rocha. 2015. Using visual rhythms for detecting video-based facial spoof attacks. *IEEE Transactions on Information Forensics and Security* 10 (5):1025–38. doi:10.1109/TIFS.2015.2395139.
- Shi, B., X. Bai, and S. Belongie (2017). Detecting oriented text in natural images by linking segments. In *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2550–58.
- Souza, M. R. (2018). Digital video stabilization: Algorithms and evaluation. Master's thesis, Institute of Computing, University of Campinas, Campinas-SP, Brazil.
- Souza, M. R., and H. Pedrini. 2020. Visual rhythms for qualitative evaluation of video stabilization. *EURASIP Journal on Image and Video Processing* 2020:1–19. doi:10.1186/s13640-020-00508-4.
- Souza, M., H. Maia, M. Vieira, and H. Pedrini. 2020. Survey on visual rhythms: A spatio-temporal representation for video sequences. *Neurocomputing* 402:409–22. doi:10.1016/j.neucom.2020.04.035.

- Sravani, M., A. Maheswararao, and M. K. Murthy. 2021. Robust detection of video text using an efficient hybrid method via key frame extraction and text localization. *Multimedia Tools and Applications* 80 (6):9671–86. doi:10.1007/s11042-020-10113-2.
- Tacon, H., A. S. Brito, H. L. Chaves, M. B. Vieira, S. M. Villela, H. de Almeida Maia, D. T. Concha, and H. Pedrini (2019). Human action recognition using convolutional neural networks with symmetric time extension of visual rhythms. In *International Conference on Computational Science and Its Applications*, Saint Petersburg, Russia, 351–66. Springer.
- Torres, B. S., and H. Pedrini. 2018. Detection of complex video events through visual rhythm. *The Visual Computer* 34 (2):145–65. doi:10.1007/s00371-016-1321-1.
- Valery, G., and S. Jean (2020). Detection and localization of embedded subtitles in a video stream. In *International Conference on Computational Science and Its Applications*, Cagliari, Italy, 119–28. Springer.
- Valio, F. B., H. Pedrini, and N. J. Leite. 2011. Fast rotation-invariant video caption detection based on visual rhythm. In *Iberoamerican Congress on Pattern Recognition*, ed. César San Martín and Sang-Woon Kim, 157–64. Springer.
- Villamizar, M., O. Canévet, and J.-M. Odobez. 2020. Multi-scale sequential network for semantic text segmentation and localization. *Pattern Recognition Letters* 129:63–69. doi:10.1016/j.patrec.2019.11.001.
- Wu, J.-C., J.-W. Hsieh, and Y.-S. Chen. 2008. Morphology-based text line extraction. *Machine Vision and Applications* 19 (3):195–207. doi:10.1007/s00138-007-0092-0.
- Yin, X.-C., Z.-Y. Zuo, S. Tian, and C.-L. Liu. 2016. Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Transactions on Image Processing* 25 (6):2752–73. doi:10.1109/TIP.2016.2554321.
- Zedan, I. A., K. M. Elsayed, and E. Emary (2016). Caption detection, localization and type recognition in Arabic news video. In *10th International Conference on Informatics and Systems*, Giza, Egypt, 114–20. ACM.
- Zhang, Y., and T.-S. Chua. 2000. Detection of text captions in compressed domain video. In *ACM Workshops on Multimedia*, 201–04. New York, NY, USA: ACM.
- Zhang, Z., C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai (2016). Multi-oriented text detection with fully convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 4159–67.
- Zhong, Y., H. Zhang, and A. K. Jain. 2000. Automatic caption localization in compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (4):385–92. doi:10.1109/34.845381.