JOURNAL OF DATA SCIENCE | Vol.2025, Issue 1, No.15
eISSN:2805-5160

Image Segmentation and Text Extraction Techniques for Efficient Information Retrieval

Wydyanto[1*], Ade Putra[2], Maria Ulfa[3]

[1,2,3]Faculty Science Technology Universitas Bina Darma Palembang, Sumatera Selatan Indonesia

Email: wydyanto@binadarma.ac.id[1*], ade.putra@binadarma.ac.id[2], maria.ulfa@binadarma.ac.id[3]

Abstract

Image segmentation and text extraction are important tasks in the field of computer vision and image processing. Image segmentation and text extraction involve identifying and separating objects or regions within an image, and helping visually impaired individuals access visual content. The ability to accurately extract text from scene images can have various applications and benefits. This can help automatically generate captions or descriptions for images, making them more accessible to individuals with visual impairments. This can help in tasks such as indexing and image search, where the extracted text can be used to improve the accuracy and relevance of search results. In addition, the extracted text can be used for sentiment analysis or other forms of text-based analysis, providing valuable insights into the content and context of the images. As the techniques discussed in the paper have the potential to help enhance the utility and usability of scene images in various applications and domains.

Keywords

Image Segmentation, Text Extraction, OCR, Visual Accessibility, Scene Images, Information Retrieval

Introduction

One of the main advantages of text extraction from images is its potential to assist individuals with visual impairments by conveying text information through audio to detect objects and convey their contours through tactile graphics, allowing BVI users to hear visual information from the natural scene around them (Mukhiddinov & Cho, 2021) (Mukhiddinov & Kim, 2021)(Mlyahilu, Mlyahilu, Lee, Kim, & Kim, 2022). This can also help in tasks such as indexing and image search, where the extracted text can be used to improve the accuracy and relevance of search results. "Automated sentiment analysis can be used for sentiment analysis on session patient records in an eating disorder treatment setting, providing valuable insights into the content and context (Hoda, Salleh, Grundy, & Tee, 2017).

Submission: 11 September 2025; **Acceptance:** 30 November 2025; **Available Online:** December 2025

Overall, the techniques discussed in this paper have the potential to significantly enhance the utility and usability of scene images in various applications and domains. The presence of text data in images and videos containing useful information for automatic deletion, indexing, and organizing images. "Text information extraction involves text detection, localization, tracking, extraction, enhancement, and recognition of text from images. (Kim, 2015)(Jung, Kim, & Jain, 2004). "Automatic text extraction is challenging due to variations in text size, style, orientation, alignment, low image contrast, and complex backgrounds (Jung et al., 2004). Techniques like text detection, localization, tracking, extraction, enhancement, and recognition are used to address these challenges.

Text extraction from natural scene images involves detection and localization, text extraction and enhancement, and optical character recognition (OCR). The challenges include variations in font size, color, alignment, illumination change, and reflections (Kim, 2015). (Jung et al., 2004)(Zhang, Zhao, Song, & Guo, 2013). Although comprehensive surveys on related issues such as face detection, analysis, documents, and image & video indexing can be found, the issue of text information extraction has not been well explored. In this paper, we present a generic framework and methodology for automatically extracting text content from images obtained from slide video recordings. "Automatic text extraction is challenging due to variations in text size, style, orientation, alignment, low image contrast, and complex backgrounds. Techniques like text detection, localization, tracking, extraction, enhancement, and recognition are used to address these challenges.

Text extraction from natural scene images involves detection and localization, text extraction and enhancement, and optical character recognition (OCR)(Seema Barate1, Chaitrali Kamthe2, Shweta Phadtare3, 2016). The challenges include variations in font size, color, alignment, illumination change, and reflections. (Jung et al., 2004) (Thesis et al., 2014) Specifically, we are using video lectures as our initial target because they can serve as a foundation for other scenarios such as meetings and conferences. In addition to OCR, we also discuss how unique information is available in text layout. This information can be used for indexing and retrieval.

Various approaches for text information extraction from images and videos have been proposed for specific applications, including license plate extraction, cargo container verification, text-based video structuring, genre recognition, and part identification in industrial automation. These approaches use methods such as learning-based, texture-based, gradient information, connected component-based, and OCR technology (Kwang & Anil, 2004). However, extensive research, it is not easy to design a series of general objectives. system.

This is because there are many potential sources of variation when extracting text. "Text information extraction from images and video is a challenging task due to variations in font size, style, color, orientation, and alignment. Existing methods for text extraction include edge-based, connected component-based, and texture-based detection, each with its limitations. A new method has been proposed for positioning text regions automatically in low-contrast images, which involves image color space transformation, edge detection, image enhancement, morphological operations, connected component analysis, and SVM. This method has shown good results in

accurately locating text areas in low-contrast images with different sizes and languages (Liu, Jiang, Cun, Shi, & Hao, 2017) (Ye, Huang, Gao, & Zhao, 2005).

The variation in drawing pencil lines can be made difficult to automate by introducing a deviational parameter called squiggle, which adds random displacements to create irregular paths. (Qijie Zhao1, 2014)(Lu, Chen, Tian, Lim, & Tan, 2015). In general, text detection methods can be classified into three categories. (AlMeraj, Wyvill, Isenberg, Gooch, & Guy, 2009)The first consists of methods based on connected components, which assume that text regions have a uniform color and meet certain size, shape, and spatial alignment constraints.

The method described in the source (Juang, Tsai, & Fan, 2015) for extracting text in color images is not effective when the text has a color similar to the background. (Mancas-Thillou & Gosselin, 2007) (Jung et al., 2004) . Our proposed approach is based on morphology, which consists of a dilation and erosion process to extract text and recognize black and white text areas that contain document text or images.(Šarić, 2017) The second consists of texture-based methods, which assume that the region text has a specific texture. Although this method is comparatively less sensitive to background color, it may not distinguish text from a background that is similar to the text. The third yang consists of edge-based methods (Kulkarni, Harale, & Thakur, 2018).

Text regions are detected under the assumption that the edges of the background and object areas are less frequent than the text regions. However, such an approach is not very effective for detecting large-sized texts. font size. (Lu et al., 2015) compared Support Vector Machines (SVM)-based methods with multilayer perceptrons (MLP)-based methods for text verification based on four independent features, namely, distance map features, spatial gray-level co-occurrence matrix features, constant gradient variance features, and DCT coefficient features. (Hsin-Te Lue, Ming-Gang Wen, Hsu-Yung Cheng, Kuo-Chin Fan, Chih-Wei Lin, n.d.).

They found that better detection results were obtained by SVM rather than by MLP. Multi-resolution-based text detection methods are often adopted to detect text at different scales (Soumya & Hegde, 2018). Additionally, the redundant temporal information available from the video is often used in verifying candidate text regions and eliminating falsely detected text regions. "Text regions in videos often have distinct characteristics such as high contrast with the background or specific font-color and languages. These characteristics can be utilized for text detection algorithms (Ye et al., 2005).

By analyzing the temporal information across frames, these methods can improve the accuracy of text verification and reduce false positives. Furthermore, the use of multi-resolution techniques allows for the detection of text in videos with varying levels of resolution, ensuring that text is accurately identified regardless of its size or scale within the video frame.

The combination of multi-resolution analysis and temporal information processing has proven to be effective in enhancing text verification in video content (Jung et al., 2004)(Jung et al., 2004). Multi-resolution text regions can be detected from edge maps sampled from the original image. The paper proposes a unified framework for video text detection, localization, and tracking in compressed videos based on block DCT coefficients. Various methods such as connected component-based and texture-based are used for text detection.

The DCT coefficients of an 8x8 block of an I-frame are selected to represent the texture intensity of the block. (Qian, Liu, Wang, & Su, 2007) Candidate text block regions are verified by texture constraints. The experimental results show the effectiveness of the proposed methods (Justo, 2015). The main contribution of this paper lies in proposing a shape decomposition-based strategy to segment compound characters into basic shapes, reducing classification complexity and improving recognition accuracy. The method uses chain code histogram feature set with MLP based classifier (Pakrasi, Laviers, & Chakraborty, 2018).

The purpose of Optical Character Recognition (OCR) is to classify optical patterns (often contained in digital images) that correspond to alphanumeric characters or other characters. "The OCR process involves segmentation, feature extraction, and classification. Any standard OCR software can be used for text recognition in segmented frames (Coates, Chin, & Chung, 2011). However, a deep examination of the character candidate region properties in the frame or image segment shows that most OCR software packages will face significant difficulties in recognizing the text. "Document images differ from natural images primarily because they contain text with some graphics and images (Hamad & Kaya, 2016).

Text detection, localization, extraction, geometrical normalization, enhancement /binarization, and recognition are essential procedures for processing document images. Document image analysis involves classifying document images using text-based methods, visual features, and hybrid methods for the highest accuracy(Saric, 2017). Document comparison methods include text recognition, visual similarity measures, and duplicate detection techniques (Libouga et al., 2018). Because the images have very low resolution and were captured using handheld devices, it is difficult to extract the complete layout structure (logical or physical) from these documents, and even more challenging to apply standard OCR systems. (Thesis et al., 2014) The proposed method combines color and layout features to identify documents captured from low-resolution handheld devices. It estimates the document image color density surface and represents it with an equivalent ellipse, while also computing the document's shallow layout structure hierarchically.

These combined features are arranged in a symbolic file unique to each document, called the document's visual signature. The identification method uses color information in the signatures to narrow down the search space based on similar color distribution and then selects the document with the most similar layout structure. This approach is particularly useful for slide documents captured using handheld devices (Behera, Lalanne, & Ingold, 2005). In the case of original electronic documents in the repository, the extraction of the same signature is straightforward; the PDF or PowerPoint format of the original electronic document is converted into images with relatively high resolution (TIFF, JPEG, etc.) where the signature is calculated. Finally, the captured document signatures were compared with all the original electronic documents. signing documents to find a match

Methodology

The proposed system performs automatic text extraction from MPEG video using a structured pipeline that includes frame extraction, segmentation, classification, and OCR. The methodology is divided into six main stages, as depicted in Figure 1.

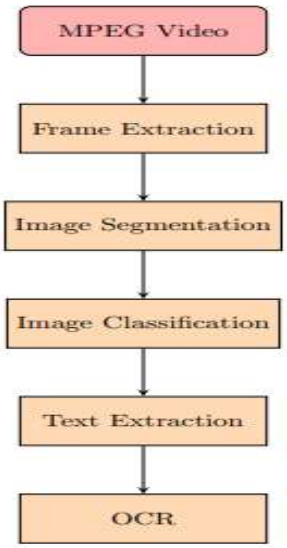

Fig 1. Workflow for Text Extraction from MPEG Video using Image Segmentation and OCR.

A. MPEG Video Input

The process begins with input in the form of an MPEG video file, typically consisting of lecture recordings, conference sessions, or slideshow-based presentations. These videos contain temporal visual content from which text needs to be extracted.

B. Frame Extraction

The MPEG video is segmented into individual frames at a predefined frame rate (e.g., 1 frame per second). These static frames are used as the basis for subsequent image processing. Frame selection can be uniform or adaptive depending on slide change detection.

C. Image Segmentation

Each extracted frame undergoes image segmentation to isolate regions of interest, particularly text-containing regions. Otsu thresholding is applied to convert grayscale images into binary images, enhancing contrast between foreground text and background.

Step 1: Convert image to grayscale.
Step 2: Apply Otsu's method to obtain a binary image.
Step 3: Perform noise reduction and morphological operations to refine regions.

D. Image Classification

Segmented image regions are classified into two categories:

- Textual Regions: Contain potential text components based on edge density, aspect ratio, and region shape. Non-textual Regions: Contain diagrams, figures, or background clutter.
- This classification step reduces false positives before applying text extraction.

E. Text Extraction

Using Run-Length Smoothing Algorithm (RLSA), textual regions are connected horizontally and vertically to form complete words or lines. Projection profile analysis further segments lines into words and characters. Horizontal RLSA: Connects characters along the line.

Vertical RLSA: Refines vertical alignment. Combination: Logical AND of horizontal and vertical RLSA improves text block isolation.

F. Optical Character Recognition (OCR)

The localized and binarized text blocks are processed using OCR (e.g., ABBYY FineReader or Tesseract). The OCR engine converts image-based characters into digital text. OCR output is filtered to remove noise or misrecognized symbols. Final text output is stored for indexing or further analysis. To clarify the sequence of operations in the proposed system, the following table summarizes each stage of the methodology along with its function and the techniques employed.

Table 1.Summary of Methodology Steps

Stage	Description	Techniques/Tools Used
1. MPEG Video Input	The system receives input in the form of an MPEG video containing slide presentations or lecture content.	Video decoding, FFmpeg
2. Frame Extraction	Key frames are extracted from the video at regular intervals or based on content changes.	Frame sampling, keyframe detection
3. Image Segmentation	Each frame is converted to grayscale and segmented to isolate foreground text from background.	Otsu thresholding, morphological filtering
4. Image Classification	Segmented regions are analyzed and classified as text or non-text (e.g., diagrams, images).	Heuristics: shape, density, aspect ratio
5. Text Extraction	Connected components and Run-Length Smoothing Algorithm (RLSA) are applied to extract and refine text regions.	RLSA (horizontal & vertical), projection profile
6. OCR Processing	Recognized text blocks are passed through an OCR engine to convert image text to machine-readable text.	ABBYY FineReader / Tesseract OCR

Each stage in the methodology plays a crucial role in ensuring accurate text detection and extraction. The integration of segmentation, classification, and OCR enables the system to handle complex image conditions such as low contrast, varying fonts, and noisy backgrounds.

Results and Discussion

Image segmentation

If the document image appears different in terms of overall resolution, then resampling is necessary to bring the image to the overall resolution. This is necessary because in this scenario, the geometric properties of various visual features are considered during the matching for document identification captured in images. Due to poor resolution, it is not possible to advance to the character level when adjacent characters overlap in the captured document. "The captured document image is pre-processed for perspective correction and noise removal. (K. Bhargavi & S. Jyothi, 2014)(K. Bhargavi & S. Jyothi, 2014).

Additionally, the average horizontal run length of both black and white pixels in the precise segmentation of the foreground object was examined. For example, for an image document with a dark background and a bright foreground, the inverted binarization result is a black background (represented as 0) and a white foreground (represented as 1) (Figure 2 (a), (b).

Fig 2. Otsu segmentation: (a) original slide document, (b) output of Otsu segmentation

The white pixels in the Otsu segmentation output are counted. Usually, the average horizontal run length of background pixels is much higher than that of foreground pixels. If the average length of horizontal runs of black pixels is relatively higher compared to white pixels in the binary image from Otsu's segmentation output, then the black and white pixels are only replaced with the necessary image. Image 1 depicts one such image that has a dark background and a brighter foreground. For this image (Fig.41(b)), the average horizontal length of black pixels is 32.3 and for white pixels is 6.1. The image in Figure 4.2 (b) is corrected using this black and white path information for perfect segmentation, with black as the foreground and white as the background.

Localization and text extraction ?

Path Length Smoothing Algorithm

The RLSA method is applied row by row and column by column on the binary mentioned above. an image of a document representing white pixels with 1 and black pixels with 0. RLSA transforms a binary sequence x into an output sequence y according to the rules described by Behera as follows [10]: i) 1 in x is changed to 0 in y if the number of adjacent 1s is less than the specified threshold, T. ii) 0 in x remains unchanged in y.

For example, with T = 5, the sequence x is mapped to y, as illustrated in Figure 4.2. It sounds like you're describing an application of the Run Length Smoothing Algorithm (RLSA), which is often used in document image processing to connect neighboring foreground pixels (such as text characters) that are separated by small gaps. The idea is to modify the binary sequence based on the adjacency of 1s in the sequence, with a threshold.

Here's a breakdown of how RLSA works in this context:
1. Input: A binary sequence x (with 0s and 1s), and a threshold T.
2. Transformation Rule:
   - If a sequence of 1s in x is shorter than T, those 1s are converted into 0s in y.
   - If a sequence of 1s in x is equal to or longer than T, those 1s remain as 1s in y.
   - Any 0 in x stays as 0 in y (unchanged).

RLSA Example with T = 5:
Let's assume the following binary sequence:
x = [1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1]

For each group of adjacent 1s, we count their length:
- Group 1: 1, 1 → length = 2 (less than 5), so these are converted to 0s in y.
- Group 2: 1, 1, 1 → length = 3 (less than 5), so these are converted to 0s in y.
- Group 3: 1, 1, 1, 1, 1 → length = 5 (equal to T), so these stay as 1s in y.
- Group 4: 1 → length = 1 (less than 5), so this is converted to 0 in y.

Thus, the transformation looks like this:
x = [1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1]
y = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0]

Visualization:

This transformation can also be visualized with a diagram (flowchart or image) if you'd like to see a graphical representation of the process. Let me know if you'd like a visual or further explanation of the algorithm. The basic idea of RLSA is to connect adjacent black areas when they are separated. less than T pixels. The level of connectivity depends on T, the distribution of white and black pixels in the document, and the document's resolution in 'dpi' (dots per inch). Both different bitmaps are generated using RLSA in both horizontal and vertical directions. Often, the distance between components in a document image tends to differ horizontally and vertically.

Therefore, two different thresholds Th and Tv are used for RLSA in each horizontal and vertical direction. For the slide document, this threshold is set as Th = 80 and Tv = 100. Two bitmaps from the RLSA output, each in the horizontal and vertical directions, are combined using the AND logical operator to detect various components in the image document. Additional horizontal smoothing using RLSA (Ts = 15) results in the final segmentation output. Figure 4.3 illustrates the RLSA algorithm horizontally, vertically, and in combination. output f(c). (c). (c). This threshold value has been evaluated and adjusted using around one hundred slide images.

Produce the images of the RLSA (Run Length Smoothing Algorithm) results in both horizontal and vertical directions, followed by the combined result, I will first perform Otsu's

segmentation on the image, then apply the RLSA in the respective directions. Here's the approach I will follow:

1. Horizontal RLSA: Apply smoothing by extending the length of runs of 1's along the horizontal axis.
2. Vertical RLSA: Apply smoothing by extending the length of runs of 1's along the vertical axis.
3. Combined RLSA: Combine the results of horizontal and vertical RLSA to generate a final processed image.

I'll start with Otsu segmentation and then perform RLSA. Let me proceed.

Fig 3. Output of the RLSA: (a) horizontal direction, (b) vertical direction and (c) combining both the directions, of the binary image in Fig. 4.1 (b)

Here are the output images from the RLSA (Run Length Smoothing Algorithm):

1. Horizontal RLSA: Download Horizontal RLSA Output
2. Vertical RLSA: Download Vertical RLSA Output
3. Combined RLSA (Horizontal + Vertical): Download Combined RLSA Output

These results show how the RLSA operates in both horizontal and vertical directions, followed by the combined effect. Let me know if you need further modifications or explanations!

Introduction to Optical Character Recognition

Binarization is achieved with the gray threshold value obtained from the Otsu method [38]. Additional steps were taken to obtain a better binary image. First of all, a horizontal projection profile analysis is conducted to detect white text on a black background. If the text color is considered white, the OT area is inverted. This is necessary because most OCR software works best with black. text on a white background. An example of Otsu segmentation is illustrated in Figure 4.3.

Fig 4. Otsu segmentation: (a) original slide document, (b) output of Otsu segmentation

Here are the results of the image processing:

1. **Original Slide Document:**
2. **Otsu's Segmentation**

Finally, vertical projection profile analysis was performed to discard unwanted pixel areas that fall below the calculated threshold. The final binarization image is shown in Figure 4.5 (a) and the connected components are illustrated in Figure 4.5. (b). (b). Connected components will be explained later. We use ABBYY FineReader 8.0 (http://finereader.abbyy.com) for OCR. We found that it was sufficient for our purposes.

To produce the final binarized image from the image you provided, you can follow these steps using the Otsu's binarization method:

1. Grayscale Conversion: Convert the image to grayscale to have a single color channel, which is brightness intensity.
2. Otsu's Thresholding: Apply the Otsu method to determine the optimal threshold value. Otsu's method automatically calculates the best threshold value to separate objects (text) from the background.
3. Binarization: Apply thresholding to convert a grayscale image into a binary image, where pixels with intensities above the threshold will become white (1), and those below the threshold will become black. (0)

With this method, you will get the final result in the form of an image consisting of only two colors, white and black, which separates the text from the background. Here is the result of the binarization process applied to the given image. On the left is the original grayscale image, and on the right is the binary version, where the thresholding method has been used to convert it to a binary format for text extraction or OCR processing. This binary image highlights the text and objects that stand out while filtering out background noise.

Fig 5. The extracted text; (a) result for OCR of a binarized frame. (b) connected components

Analysis of Extracted Characters

ABBYY FineReader is good OCR software but not for low-resolution documents. After extracting individual characters in a document and determining their properties, we compared them to the original text. Letters like "g", "j" & "y", which extend below the baseline, are in one group; tall letters, like "l" and "T", are in another group; short letters, like "a" and "c", are in another group; and floating characters, like "'" and "^", are at the end. After being classified, the extracted characters are compared with the studied characters that belong to the same group. If no good match is found, the extracted character is then compared with other original characters, regardless of the group. Because we found that some characters successfully passed through the original

character recognition algorithm, we deemed it necessary to perform additional operations on the less recognized characters.

The main cause that can be observed from the recognition errors in our original program is characters that are connected as wide characters. An "r" will only touch an "i" very slightly, and that character will be recognized as an "n". To address this issue, we divided the character at its narrowest point. This algorithm might cause problems with something like "mi" -- with the poorly scanned "m", the combined character could get corrupted in the middle of the "m", find an "n", and do something unexpected with the remnants of the letters "m" and "i". Another common cause of misidentification is character fragmentation. An "r" might be split in the middle, leaving a figure resembling "l" on the left, and something incomprehensible on the right. However, with practice and training, these recognition errors can be reduced.

Improving Text Information

In addition to OCR obtained from binary frames, we propose several other unique features that can be used to represent low-resolution images. Layout shape signatures can be used as feature vectors to measure the distance between two images in a query with an example search system. Here we brie(MST). (MST). (MST).

Sentence Length

Using text as a feature vector will result in a discrepancy between the original text and the generated one. Therefore, an algorithm is proposed to classify horizontal words into three categories: short, medium, and long sentences. The sentences are determined from a number of connected components as shown in Figure 5.1. A connected component consists of a set of characters.

Here is a simple algorithm to classify horizontal words into three categories: short sentences, medium sentences, and long sentences based on the number of characters or the length of the words in pixels (for example, the result of the bounding box on the text after OCR or connected components):

1. Input:
   - The result of OCR or connected components from an image containing text.
   - The size or length of the sentence in pixels or number of characters.
2. Define the category limit for sentence length:
   - Short sentence: A sentence with a character count ≤ 30 or a sentence length in pixels ≤ 200.
   - Medium sentence: A sentence with a character count > 30 and ≤ 60 or a sentence length in pixels > 200 and ≤ 400.
   - Long sentence: A sentence with more than 60 characters or a sentence length in pixels greater than 400.
3. Process:

   Step 1: **Take the text from OCR results or bounding box detection.**

   Step 2: **For each generated sentence or word:**

   Calculate its length, either based on the number of characters or the length of the bounding box.

   Step 3: **Classify the sentence length:**

- If the number of characters ≤ 30 or the length of the sentence ≤ 200 pixels → Short sentence.
- If the number of characters > 30 and ≤ 60 or the length of the sentence > 200 and ≤ 400 pixels → Medium sentence.
- If the number of characters > 60 or the length of the sentence > 400 pixels → Long sentence.
  Contexto: 4. Salida: Texto a traducir: 4. Salida:
  o The category of each sentence: "Short", "Medium", or "Long".

Explanation of the Algorithm:

Input Step: The algorithm takes the OCR results or connected components that have been extracted from the image.

Measurement: The length of the sentence is measured both in the number of characters and based on the length of the bounding box in pixels. Classification: Based on the length of the sentence, it will be categorized into short, medium, or long sentences according to the predetermined limits. With this algorithm, you can automatically classify text from images into three categories after text extraction using OCR or connected components method.

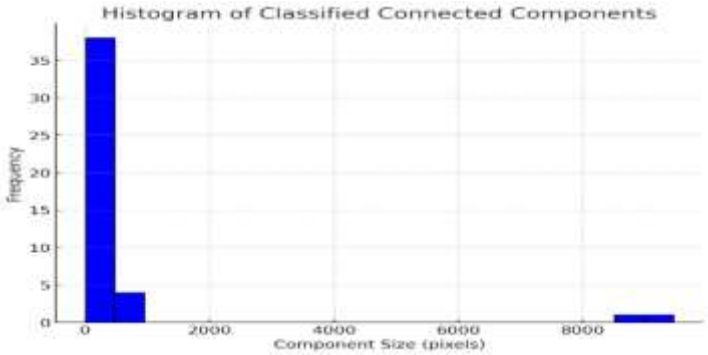

Fig 6. The histogram of classified connected components.

Minimum spanning tree

Basically, a minimum spanning tree is a subset of the edges of a graph, such that there is a path from any vertex to any other vertex and the sum of the weights of these edges is minimized. We consider each sentence as a node. Here is the minimum spanning tree from the example (Figure 5.2) of Prim's Algorithm by Robert C. Prim.The algorithm (greedily) builds a minimum spanning tree. by adding nodes iteratively to the work tree:

1. Start with a tree that has only one node.
2. Identify a node (outside the tree) that is closest to the tree and add the minimum edge weight from that node to several nodes in the tree, and merge the additional node as part of the tree.
3. If there are fewer than n – 1 edges in the tree, return to 2 For the example graph, here's how it works. The generated graph will have 8 MSTs as shown in the Figure. 5.2

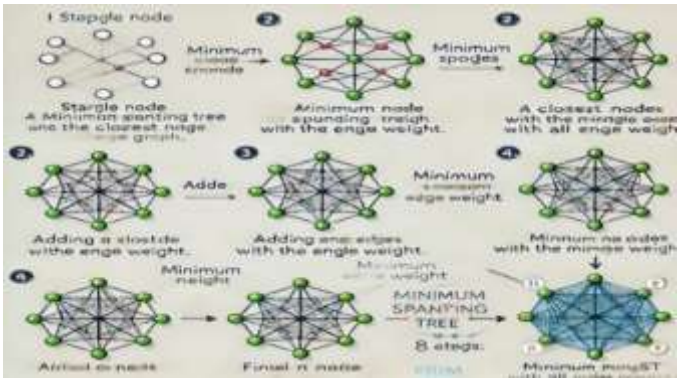

Fig 7. Prim's Algorith m; (a) Nodes, (b) Edges,(c) MST

Conclusion

Image segmentation is an important task and requires careful attention. We have presented an innovative and new framework for information separation in an image. This research, which aims to obtain text information including OCR and layout signatures, has focused on images taken from slideshow videos. While OCR allows us to search for words in documents, layout signatures will be used in indexing.

A good indexing scheme for an image or frame will be able to produce fast and accurate retrieval. However, the results are highly dependent on the quality of the image or OCR document. In general, the proposed algorithm can provide reliable OCR results. This allows keyword searching for retrieval applications. However, for low-resolution images, we propose a new signature derived from textual information. The new feature of the textual information we obtain as categorized nodes and MST is unique to an image. In the future, we intend to develop a Content-Based Indexing and Retrieval (CBIR) system using features obtained from vectors. We believe that through proper indexing, the proposed CBIR system is capable of addressing the issue of low-resolution documents. In this paper, our focus is on images of scenes from meetings and conferences. After we have an efficient CBIR, it is not difficult to create a similar one. system for video and images from another scenario.

Acknowledgements

The authors would like to express their sincere gratitude to Universitas Bina Darma Palembang for the support and resources provided throughout the course of this research. Special thanks are extended to the Faculty of Science and Technology for facilitating access to laboratory equipment and technical assistance. We also appreciate the constructive feedback from peer reviewers, which greatly contributed to improving the quality of this paper. Lastly, we acknowledge the contributions of our colleagues and research assistants who provided valuable input during the experimentation and analysis phases of this study.

References

AlMeraj, Z., Wyvill, B., Isenberg, T., Gooch, A. A., & Guy, R. (2009). Automatically mimicking unique hand-drawn pencil lines. Computers and Graphics (Pergamon), 33(4), 496–508. https://doi.org/10.1016/j.cag.2009.04.004

Behera, A., Lalanne, D., & Ingold, R. (2005). Combining Color and Layout Features for the Identification of Low-resolution Documents. International Journal of Signal Processing, 2(1), ISSN: 1304-4478: 7-14.

Coates, D. R., Chin, J. M., & Chung, S. T. L. (2011). Discovery of an in vivo Chemical Probe of the Lysine Methyltransferases G9a and GLP. Bone, 23(1), 1 – 7.

Hamad, K., & Kaya, M. (2016). A Detailed Analysis of Optical Character Recognition Technology. International Journal of Applied Mathematics, Electronics and Computers, 4(Special Issue-1), 244–244. https://doi.org/10.18100/ijamec.270374

Hoda, R., Salleh, N., Grundy, J., & Tee, H. M. (2017). Systematic literature reviews in agile software development: A tertiary study. Information and Software Technology, 85, 60–70. https://doi.org/10.1016/j.infsof.2017.01.007

Juang, J., Tsai, Y., & Fan, Y. (2015). Visual Recognition and Its Application to Robot Arm Control. 851–880. https://doi.org/10.3390/app5040851

Jung, K., Kim, K. I., & Jain, A. K. (2004). Text information extraction in images and video: A survey. Pattern Recognition, 37(5), 977–997. https://doi.org/10.1016/j.patcog.2003.10.012

Justo, P. (2015). Raspberry Pi vs Arduino. December 4, 2015, 1. Retrieved from https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/

K. Bhargavi, & S. Jyothi. (2014). A Survey on Threshold Based Segmentation Technique in Image Processing. International Journal of Innovative Research & Development, 3(12), 234–239. https://www.researchgate.net/profile/Singaraju-Jyothi/publication/309209325

Kim, K. B. (2015). Image binarization using intensity range of grayscale images. International Journal of Multimedia and Ubiquitous Engineering, 10(7), 139–144. https://doi.org/10.14257/ijmue.2015.10.7.15

Kulkarni, S. S., Harale, A. D., & Thakur, A. V. (2018). Image processing for driver's safety and vehicle control using raspberry Pi and webcam. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017, 1, 1288–1291. https://doi.org/10.1109/ICPCSI.2017.8391917

Libouga, D., Marius Otesteanu, Ideal Oscar Libouga, Laurent Bitjoka, & Popa, G. D. (2018). A Review on Image Segmentation Techniques and Performance Measures. *Zenodo (CERN European Organization for Nuclear Research)*, *12*(12), 1107–1117. https://doi.org/10.5281/zenodo.2571650

Liu, G., Jiang, M., Cun, H., Shi, Z., & Hao, J. (2017). An Automatic Text Region Positioning Method for the Low-Contrast Image. Journal of Computer and Communications, 05(10), 36–49. https://doi.org/10.4236/jcc.2017.510005

Lu, S., Chen, T., Tian, S., Lim, J. H., & Tan, C. L. (2015). Scene text extraction based on edges and support vector regression. International Journal on Document Analysis and Recognition, 18(2), 125–135. https://doi.org/10.1007/s10032-015-0237-z

Lue, H.-T. (2010). A Novel Character Segmentation Method for Text Images Captured by Cameras. *ETRI Journal*, *32*(5), 729–739. https://doi.org/10.4218/etrij.10.1510.0086

Mancas-Thillou, C., & Gosselin, B. (2007). Color text extraction with selective metric-based clustering. Computer Vision and Image Understanding, 107(1–2), 97–107. https://doi.org/10.1016/j.cviu.2006.11.010

Mlyahilu, J. N., Mlyahilu, J. N., Lee, J. E., Kim, Y. B., & Kim, J. N. (2022). Morphological geodesic active contour algorithm for the segmentation of the histogram-equalized welding bead image edges. IET Image Processing, 16(10), 2680–2696. https://doi.org/10.1049/ipr2.12517

Mukhiddinov, M., & Cho, J. (2021). Smart glass system using deep learning for the blind and visually impaired. Electronics (Switzerland), 10(22). https://doi.org/10.3390/electronics10222756

Mukhiddinov, M., & Kim, S. Y. (2021). A systematic literature review on the automatic creation of tactile graphics for the blind and visually impaired. Processes, 9(10), 1–31. https://doi.org/10.3390/pr9101726

Pakrasi, I., Laviers, A., & Chakraborty, N. (2018). A Design Methodology for Abstracting Character Archetypes onto Robotic Systems. Proceedings of International Conference on Movement Computing (MOCO'18). https://doi.org/10.1145/3212721.3212809

Qian, X., Liu, G., Wang, H., & Su, R. (2007). Text detection, localization, and tracking in compressed video. Signal Processing: Image Communication, 22(9), 752–768. https://doi.org/10.1016/j.image.2007.06.005

Rashid, S.F. (2014). Optical Character Recognition - A Combined ANN/HMM Approach.

Šarić, M. (2017). Scene text segmentation using low variation extremal regions and sorting based character grouping. *Neurocomputing*, *266*, 56–65. https://doi.org/10.1016/j.neucom.2017.05.021

Seema Barate1, Chaitrali Kamthe2, Shweta Phadtare3, R. J. (2016). Implementasi Ekstraksi Karakter Teks dari Gambar Tulisan Tangan yang Dipotret ke Teknik Pencocokan Template Konversi Teks. MATEC Web of Conferences.

Soumya, B., & Hegde, S. S. (2018). Text Extraction in Images. 6(2), 119–121. https://www.ijcstjournal.org/volume-6/issue-2/IJCST-V6I2P24.pdf

Ye, Q., Huang, Q., Gao, W., & Zhao, D. (2005). Fast and robust text detection in images and video frames. Image and Vision Computing, 23(6), 565–576. https://doi.org/10.1016/j.imavis.2005.01.004

Zhang, H., Zhao, K., Song, Y. Z., & Guo, J. (2013). Text extraction from natural scene image: A survey. Neurocomputing, 122, 310–323. https://doi.org/10.1016/j.neucom.2013.05.037

Zhao, Q. J., Cao, P., & Meng, Q. X. (2014). Image Capturing and Segmentation Method for Characters Marked on Hot Billets. *Advanced Materials Research*, *945-949*, 1830–1836. https://doi.org/10.4028/www.scientific.net/amr.945-949.1830