Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL APPROACH FOR CONTENT EQUIVALENCE ANALYSIS IN COMPRESSED DOCUMENT IMAGES A SYSTEMATIC STUDY

[1] KAVITA V. HORADI, [2] DR. JAGADEESH PUJARI, [3] NARASIMHA PRASAD BHAT

[1]Assistant Professor, Department of ISE, Global College of Engineering and Technology, Bangalore, India

[2] Professor, Department of ISE, SDM College of Engineering and Technology, Dharwad, India

[3] Principal Consultant, Infosys, Brookfield WI, USA

Email: [1]kavitaresearch8@gmail.com, [2]jaggudp@gmail.com, [3]bhatnp@gmail.com

ABSTRACT

Rapid growth of digital data with complex content has led to various challenges in processing. Exponential increase in the size of 'Big Data' due to videos, audios, images and textual content has created several problems which need to be addressed by the research community. Currently, huge amount of digital data is generated by various sources. The high quality data require more space and consume excessive bandwidth during transmission. To overcome these issues, digital data are stored in compressed form using different compression algorithms stated in literature. In order to analyze these data traditional schemes use decompression techniques which are a time consuming process and increases the computation overhead. To overcome these issues, currently compressed domain image processing techniques have been adopted where complete decompression may not be required. In this work, we adopt document image processing in compressed domain which contains printed text in the document images. Our main aim is to identify the similarity and find the equivalence between two or more compressed document images. In order to achieve this, first of all, we apply JPEG encoding which generates encoded data. This data further processed through the proposed line, word and character segmentation scheme. Further, we apply SIFT (Scale-Invariant Feature Transform) to extract the feature from compressed domain segmented data. Finally, feature matching scheme is applied which uses Brute force feature matcher and k-nearest neighbor. We have tested this approach on publically available PubLayNet, IIIT-AR-13K, and Tobacco-3482 datasets which contains large scale document images. The experimental analysis shows the robustness of proposed approach to identify the similarity between compressed documents images.

Keywords: *JPEG; Compressed Document Images (CDI); Document processing; Content Equivalence; Compressed Domain; SIFT; Brute force; KNN.*

1. INTRODUCTION

Recently, we have noticed a tremendous growth in multimedia data such as audios, images, videos, fax documents, receipts and invoices over the internet. These data are widely used by digital libraries and e-governance applications. Due to high storage requirement, compressing these data files is considered as an essential task which minimizes the storage requirement and improves the data transfer efficiency. However, we discuss three primary concerns of compression schemes. First, an efficient compression scheme is required to maintain the document images into compressed form for reduced space consumption and reduced bandwidth consumption during storage and transmission. Second important concern is about accessing the compressed data. The traditional algorithms are significantly reducing the storage requirement but don't provide direct access to the data. So, efficient algorithm is required to extract the data from compressed document images. Third notable concern is about readability of data because the lossy compression and progressive transmission approaches reduce the resolution and use texture-preserving process that might result in rendering the document image in an unreadable format. However, to analyze these data for further computations, decompression is performed which requires additional computations resources and time. Thus, processing these compressed data without performing decompression is motivating research topic. Currently, compressed domain image processing techniques are developed which is significant breakthrough to process compressed document images. Here, we use the term CDI and CDP to refer compressed document images and compressed domain processing respectively.

Several types of techniques are present for document image compression such as PDF, BMP, TIFF, and JPEG which are commonly used formats for digital images. TIFF is widely adopted in handwritten documents and digital libraries also it is used in network applications such as printers, fax and Xerox machines. The TIFF comprises of several compression algorithms such as CCITT Group 3 (T.4), CCITT Group 4 (T.6) and many more. The performance of T.6 is better than other algorithms [1]. Similarly, JPEG compression is a promising technique and gained popularity in compressed domain image processing. This scheme uses JPEG compressed domain coefficients for improving the performance. However, directly processing Discrete Cosine Transform (DCT) coefficients is a new concept in this field. Several techniques have adopted JPEG compression for compressed domain image processing such as use of deep learning techniques for image retrieval in JPEG compressed domain images [2] [4], compressed domain video watermarking in HEVC videos [3], video summarization [5], real-time motion detection [6] and many more.

The document images plays an important role which can be used in various cognitive processes such as graphic understanding, knowledge database creation, content categorization, text summarization, document retrieval, text classification, text recognition and document layout matching and many more. Generally, the document layout analysis techniques are classified into three categories [7] such as (a) region based methods, (b) pixel based methods, (c) connected component based methods [8]. In region based methods, the document is divided into multiple zones and then these zones are classified into semantic classes. In second approach, pixel based methods consider each pixel individually and generates a labelled image with the help of classifier and connected components based methods require local information to generate the hypothesis for objects which are inspected, combined, refined and classified. Currently, deep learning based schemes are adopted in performing layout analysis in printed text document images such as Dynamic Residual Feature Fusion [9], DoT-Net technique which uses CNN [10], MSNet [11], PCA based deep neural network [12] and many more [13]. However, these techniques are not applied on document images in compressed domain. In this work, our main aim is to process the document images in compressed domain to match their content similarity.

Rest of the paper illustrates the following sections: Section II describes the brief overview of existing techniques on document layout analysis and content matching in compressed domain. Section III presents proposed solution for printed text content equivalence detection in compressed document images. Section IV presents the experimental analysis and comparative study to show the robustness of proposed model, and finally, section V presents the concluding remarks and future works.

2. LITERATURE SURVEY

In this section, we present a brief discussion about existing techniques of image processing in compressed domain which includes several types of image compression techniques. Recently, Beratoğlu et al. [14] used this concept for vehicle license plate detection. In this work, the High Efficiency Video Coding (HEVC) based compressed video sequences are considered. These sequences are partially decoded which is used to generate the block partition and prediction unit images. Further, YOLO V3 Tiny Object Detector is applied on these partially decoded images to detect the license plates.

Liu et al. [15] focused on object tracking and reported that existing techniques are slow thus authors adopted compressed domain processing to improve the speed. In this work, video frames are divided into key and non-key frames where key frames are restored in RGB form where CNN is applied for training and detection whereas non-key frames are directly processed through the CNN based on motion information provided in compressed domain.

Jamil et al. [16] presented a content based image retrieval model in JPEG compressed domain. This scheme extracts the features from JPEG images and generates an optimal codebook by using partial decoding of images. In order to generate the codebook, optimal number of images and feature vector length using optimization cost based on precision and recall are required. The generated codebook shows a significant impact on retrieval performance.

Rajesh et al. [17] focused on document layout analysis where holistic word recognition is considered as a tedious task. To tackle this problem, authors developed a novel approach where DCT coefficient of compressed domain images are extracted and CNN model is applied on these coefficients for recognition. Similar technique is adopted in [18] for plant leaf disease detection using transfer learning.

Temburwar et al. [2] focused on CBIR systems and reported that conventional pixel domain based techniques use low level attributes such as shape, colour and texture for image retrieval. However, matching of these features with huge databases is a very time consuming process. Thus, authors adopted JPEG compressed domain processing where DCT coefficients are considered for extracting the global and local attributes. Further, it uses CNN based model called as ResNet-50 to learn the attributes.

Phadikar et al. [19] developed a CBIR system in DCT compressed domain by using hybrid schemes with the help of genetic algorithm. In this approach, a combination of color histogram, color moments and edge histogram are extracted directly from the compressed domain. Further, Euclidian distance is computed to measure the similarity for image retrieval.

Delac et al. [20] developed face recognition technique in JPEG and JPEG2000 compressed domain image processing. This approach avoids the full decompression of image and considers only transform coefficients as input for face recognition.

Byju et al. [21] developed remote sensing image classification system in JPEG-2000 compressed domain using deep learning. The conventional deep learning techniques require fully decompressed image which consumes more time. To overcome this issue, authors proposed a technique to process JPEG 2000 compressed remote sensing image. This approach is divided into two phases where first of all, finer resolution subbands of reversible biorthogonal wavelet filters are approximated in the JPEG 2000. In next phase, high level semantic content of approximated wavelet subbands and scene classification are characterized based on the descriptors. In order to achieve this, wavelet subabnds approximated to finer resolution subbands with the help of transposed convolution layers. Later, convolution layers are used to model the high-level semantic content of approximated wavelet subbands.

These studies shows that the compressed domain image analysis improves the computation speed and achieves better accuracy in various applications. Currently, document layout analysis and content matching in document images are the hot research areas and performing these tasks in compressed domain is challenging due to loss of original content post compression. The automated content matching and layout analysis (DLA) is beneficial in various applications. Several techniques have been presented for this purpose such as Wu et al. [9] introduced Dynamic Residual Fusion Network (DRFN) for DLA. This DRFN model uses low-dimensional information and generates high-dimensional category information. Moreover, to deal with overfitting problems dynamic selection mechanism is used which helps to fine-tune the limited training data.

Kosaraju et al. [10] developed Document Layout Classification Using Texture-based CNN (DoT-Net) to classify the various classes of document blocks. Mainly it adopts the dilated convolution layer and replaces other convolution layers for texture analysis. Moreover, it uses deep learning scheme for feature extraction rather than using predefined features.

3. PROPOSED MODEL

In this section, we present the proposed approach for document content matching in JPEG compressed domain images. First of all, we present the JPEG encoding scheme which generates the compressed image data. In this work, we use DCT coefficients rather than considering the entire image and these coefficients are processed through the SIFT feature extraction. Further, we apply SIFT matching scheme to find the content similarity in document images. The overall proposed architecture is depicted in below given figure 1:

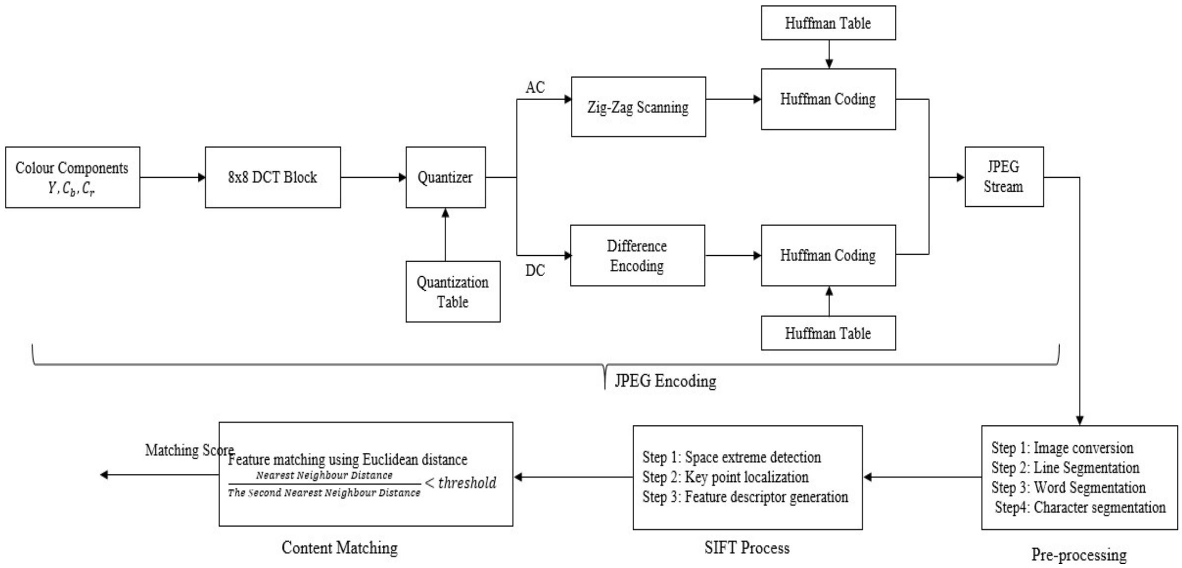

Huffman Table
AC
Zig-Zag Scanning
Huffman Coding
Colour Components
Y, C_b, C_r
8x8 DCT Block
Quantizer
JPEG
Stream
Quantization
Table
DC
Difference
Encoding
Huffman Coding
Huffman Table
JPEG Encoding
Matching Score
Feature matching using Euclidean distance
$\frac{Nearest\ Neighbour\ Distance}{The\ Second\ Nearest\ Neighbour\ Distance} < threshold$
Step 1: Space extreme detection
Step 2: Key point localization
Step 3: Feature descriptor generation
Step 1: Image conversion
Step 2: Line Segmentation
Step 3: Word Segmentation
Step4: Character segmentation
Content Matching
SIFT Process
Pre-processing

Figure 1: Overall Architecture of the Proposed Approach

3.1. JPEG Compression

JPEG is known as compressed domain transcoder. This process contains multiple phases such as decomposition, DCT, scaling, rounding, zig-zag mapping, Run Length Encoding, DPCM and Huffman coding.

- Decomposition: in this step, the input image is divided into 8x8 blocks
- DCT: the 8x8 decomposed blocks are transformed by applying Discrete Cosine Transform (DCT). These transformed blocks contain signal energy in the form of coefficients. The initial element (0,0) of transformed block is known as DC component and remaining 63 elements are known as AC components. The 2D-DCT of 8x8 block $f_{x,y}\ x, y = 0,1,2 \dots 7$ is given as:

$$F_{u,v} = \frac{C_u C_v}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f_{x,y} \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y}{}\right.$$

where $C_u, C_v = \frac{1}{\sqrt{2}}$ for $u, v = 0$ otherwise the value of $C_u, C_v = 1$. Here, u denotes the horizontal spatial frequency and v denotes the vertical spatial frequency, $f_{x,y}$ denotes the pixel value at (x, y) and $F_{u,v}$ is the obtained DCT coefficient at (u, v)

- Scaling: in this step, the transformed blocks are divided by the corresponding element in 8x8 quantization table.
- Rounding: now the scaled coefficients are rounded to the nearest integer. These two steps i.e. scaling and rounding are known as quantization.
- Zig-zag scanning: now the 8x8 blocks are processed through the zigzag scanning which generates a 64 element vector by using one-to-one mapping.
- Run Length Encoding (RLE): in RLE, runs of data i.e. a sequence of data where same data values are occurring in consecutive elements are stored as single data value and count. Thus, the quantized coefficients are processed through this and encoded data is obtained. The data blocks obtained from this phase are known as semi-compressed (SC) block. This SC block contains DC and AC values, and run length.
- DPCM: this step considers DC coefficient of SC blocks for encoding
- Huffman coding: this is the final stage of JPEG compression where SC blocks are converted to bit-stream.

This complete process of data encoding or compression using JPEG encoding technique is depicted in below given figure 2.

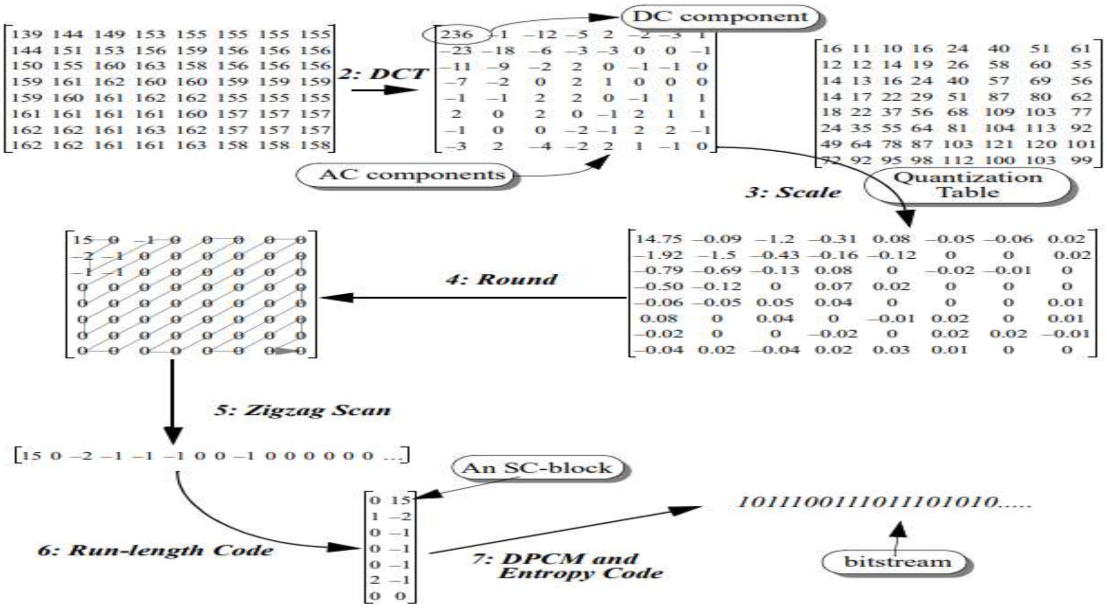


Figure 2: JPEG Encoding

3.2. Line, word and character segmentation

In this section, we describe the line, word and character segmentation methods from the compressed document images which has printed English text content. To achieve this, we have four different phase of image processing which are: (a) reading, converting and finding the threshold of image, (b) line segmentation, (c) word segmentation, and (d) character segmentation.

In first step, we consider the compressed domain color image which is converted into grayscale and processed through the thresholding operation where we define minimum and maximum pixel values. This grayscale image is processed through the Otsu's thresholding operation. Otsu thresholding considers linear discriminant criteria and it assumes that the image contains background and foreground. It helps to mitigate the overlapping of class distribution. This mechanism segments the image into light and dark regions as $T_0 = \{0,1,..,t\}$ and $T_1 = \{t, t+1,..,l-1,l\}$ where t denotes the threshold. This scheme scans all possible threshold values and computes the minimum value for the pixel for each side of the threshold. Here, the main aim of this approach is to find with minimum entropy for sum of background and foreground. Let us consider that $P(i)$ denotes the histogram probabilities of considered gray values given as:

$$P(i) = \frac{number\ \{(r,c)|image(r,c) = i\}}{(R,C)}$$

where r, c denotes the current index of row and column of the image whereas R, C denotes total number of rows and column in image. The best threshold value is known as where minimum class variance is obtained. This within class variance can be defined as:

$$\sigma_w^2 = w_b(t) * \sigma_b^2(t) + w_f(t) * \sigma_f^2(t)$$

$$w_b(t) = \sum_{i=1}^{t} P(i), w_f(t) = \sum_{i=t+1}^{l} P(i),$$

$$\mu_b(t) = \frac{\sum_{i=1}^{t} i * P(i)}{w_b(t)}, \ \mu_f(t) = \frac{\sum_{i=t+1}^{l} i * P(i)}{w_f(t)} \quad ,$$

$$\sigma_b^2(t) = \frac{\sum_{i=1}^{t}(i-\mu_b(t))^2 * P(i)}{w_b(t)} \quad \text{and}$$

$$\sigma_f^2(t) = \frac{\sum_{i=t+1}^{l}\left(i-\mu_f(t)\right)^2 * P(i)}{w_f(t)}$$

where $w_b(t)$, $\mu_b(t)$ and $\sigma_b^2(t)$ denote the weight, mean and variance of T_0 with intensity value from 0 to t, similarly, $w_f(t), \mu_f(t)$ and $\sigma_f^2(t)$ denote the weight, mean and variance of class T_1 with intensity value from $t+1$ to 1, and σ_w^2 represents the weighted sum of group variance.

In next phase, we focus on segmenting the line with the help of "close" morphological operation in iterative process. In two iterations, it generates a morphology image which is processed through the dilation phase by using 2x4 kernels. Further, we find the contour by using dilated image. Now these contours are used to generate the bounding boxes on the segmented lines. Further, this segmented line is used to obtain the word segmentation and follows the same process as used for line segmentation. Once, we obtain the segmented word, we use these word images for character segmentation where dilation step is replaced with eroding and other steps remain same as mentioned for line and word segmentation. For morphology, 2x4 kernel is used, dilation and eroding process uses 1x5 kernel.

3.3. SIFT Feature extraction

SIFT descriptor technology is based on the multiple scale spaces which was introduced by Lowe in 2004. Mainly SIFT features deals with various issues of image matching such as image rotation, affine transform, view point variation, and intensity variations. The complete process is comprised into four phases. In first phase, we compute the Difference of Gaussian (DoG) which is used for estimating the scale space extrema. In next phase, key point localization scheme is applied where key point candidates are localized which are further refined by eliminating the low contrast points. In next step, key point orientations are assigned based on the local image gradient. Finally, a descriptor generator mechanism is used to generate the local image descriptor based on the orientation, gradient and magnitude of image. These steps are described below:

3.3.1. Space extreme detection

In this step, images and their octaves are obtained. These images are further sampled by factor 2. Later, a Gaussian smoothing function is applied to smooth each image corresponding to octaves. Then, Difference of Gaussian (DoG) pyramid is obtained by taking the difference of Gaussian pyramid between two adjacent scales belonging to same octave. Figure 2 depicts the Gaussian difference and Gaussian pyramid. This can be expressed as:

$$\begin{aligned} L(x,y,\sigma) &= G(x,y,\sigma) * I(x,y) \\ DoG(x,y,\sigma) &= L(x,y,k\sigma) \\ &\quad - L(x,y,\sigma) \end{aligned} \tag{2}$$

where σ denotes the scale factor , $I(x,y)$ denotes the input JPEG image, $*$ is a convolution operation between x and y, and $G(x,y,\sigma)$ denotes the Gaussian function with varied scale space kernels.

3.3.2. Keypoint Localization

This is the second phase of SIFT approach where it analyses the data to handle the orientation, ratio, scale of principle curve and edges. Some of the points contains low contrast and carry unbalanced response to edges. These types of points lead toward the inappropriate localization. Thus, these points are discarded to obtain the robust points to decrease the false positives. For this purpose, a Taylor expansion is used in each point of interest and the low contrast points can be discarded by using following formula:

$$D(X) = D + \frac{\partial D^T}{\partial X}X + \frac{1}{2}X^T\frac{\partial^2 D}{\partial X^2}X \tag{3}$$

where $X = (x,y,\sigma)^T$ denotes the displacement from current point, the accurate position of point of interest $\hat{X}$ is obtained by computing the derivatives of function D with respect to point X. Here, derivation of the function D is set to zero as follows:

$$\hat{X} = -\left(\frac{\partial^2 D}{\partial X^2}\right)^{-1}\frac{\partial D}{\partial X} \tag{4}$$

By substituting Eq. (4) into Eq. (3), we can rewrite the Eq. (3) as:

$$D(\hat{X}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X}\hat{X} \tag{5}$$

This formula gives an absolute threshold value and we define a threshold value to discard the low contrast position. Later, these points will be processed through the Hessian matrix to improve the response to the edges.

This can be expressed as:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{6}$$

At this stage, if the value of $\frac{Tr^2(H)}{Det(H)}$ is obtained below threshold then it is considered at interest point.

3.3.3. Feature Descriptor Generation

This section describes the process to generate the feature descriptors. According to this process, a primary orientation is assigned to each point of interest which helps to compute the neighborhood orientation histogram corresponding to same point of interest. Here, orientation points are allowed for relative representation which helps to maintain the same point of interest for varied rotations of the image. Further, a maximum value of orientation histogram is obtained. In order to obtain the better and precision orientation histograms, the peaks of histogram are interpolated by adjacent points.

At this point, the actual coordinates are also rotated according to the orientation of an image which helps to generate the rotation invariant features. Later, gradient magnitude and orientation of a sample of each of its neighbouring points are computed to obtain the vector of each point of interest. This results in generating 128 element descriptor with respect to each of extracted features.

3.3.4. Document Image Content Matching

The aforementioned stages generate the local feature points from the image. In order to obtain the similarity between images, we compute Euclidean distance between two different corresponding local features in different images. During matching process, the Euclidean distance of all features in different images are computed and compared individually. Here, a feature point of first image is said to be matched if the ratio of nearest neighbour and second nearest is less than the given threshold. In this work, we use Brute-Force matcher with k-nearest neighborhood.

4. RESULTS AND DISCUSSION

This section desccribes the outcome of the proposed approach along with its intermediate stages such as line, word, character segmentation and matching of the features. The proposed approach is implemented using python 3.8 installed on windows operating system. The windows operating system has 8GB RAM, 6GB NVIDIA RTX 2060 graphic card, intel i7 processor and 1TB of storage.

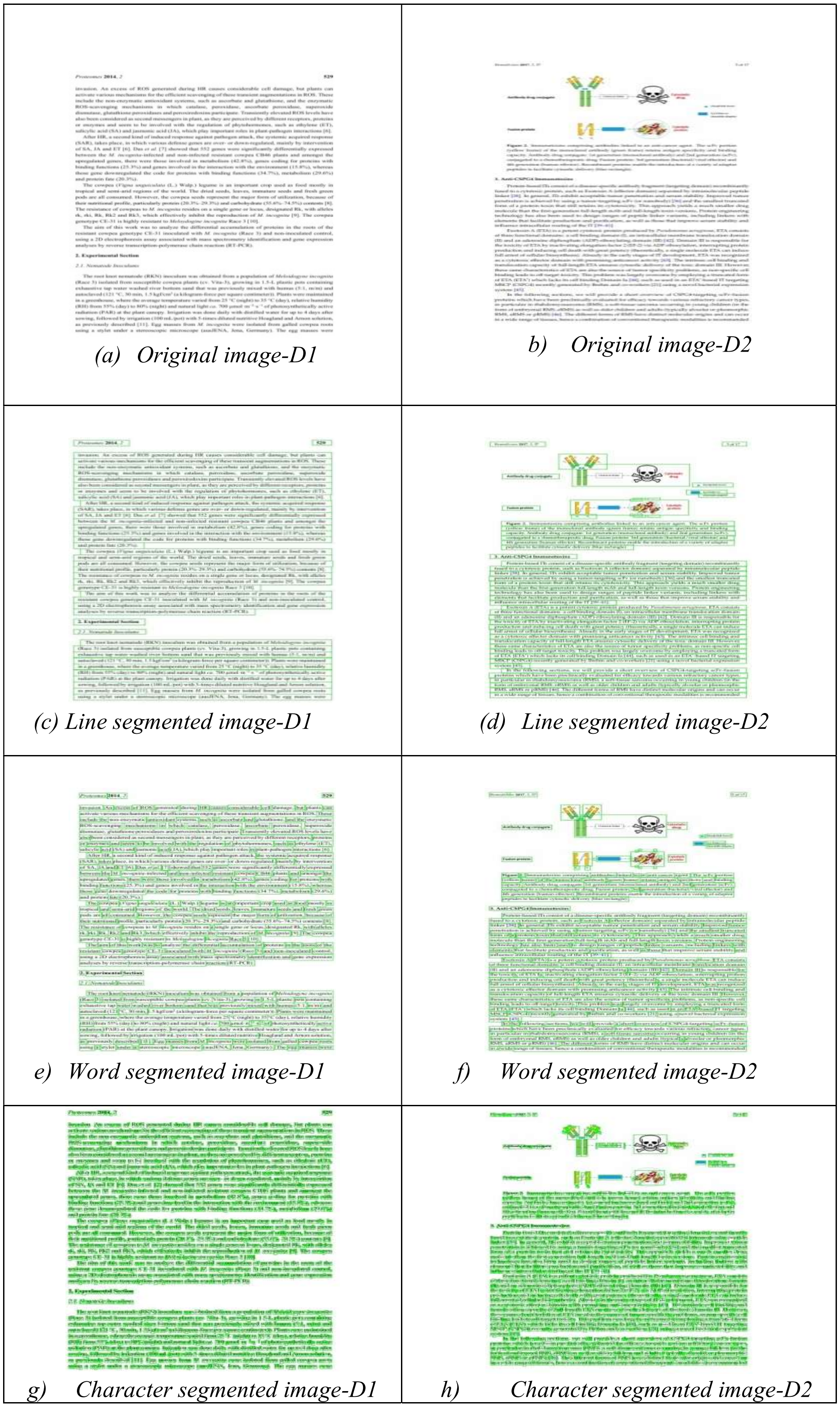
The proposed approach is tested on publically available dataset called as PubLayNet [22] IIIT-AR-13K dataset and Tobacco-3482 dataset. Publaynet is a huge dataset relased by IBM. This repository contains over 1 million PDF documents which are collected from PubMed Central. In this dataset, over 360k page samples which cover typical document layout such as figure, text, title, list and table are considered. Below given table 1 shows the different types of categories included in this dataset.

This proposed approach is tested for printed textual content and the results may vary or behave differently for the hand-written textual content or the text present as a watermark or embedded within a graphics which is a limitation of this work.

Table.1. Categories of document layout included in PubLayNet

Document Layout category	XML category
Text	Author, affilitation, article informaiton, copyright description, abstract, paragraphin main text, foot note, figure and table captions
Title	Manuscript title, standalone, subsection title.
List	List
Table	Main body of table
Figure	Main body of figure

These images are stored in JPEG form. In first step, we apply line, word and character segmentation on these compressed document images. Below given figure 3 depicts the sample outcome of the proposed approach.



(a) Original image-D1

b) Original image-D2

(c) Line segmented image-D1

(d) Line segmented image-D2

e) Word segmented image-D1

f) Word segmented image-D2

g) Character segmented image-D1

h) Character segmented image-D2

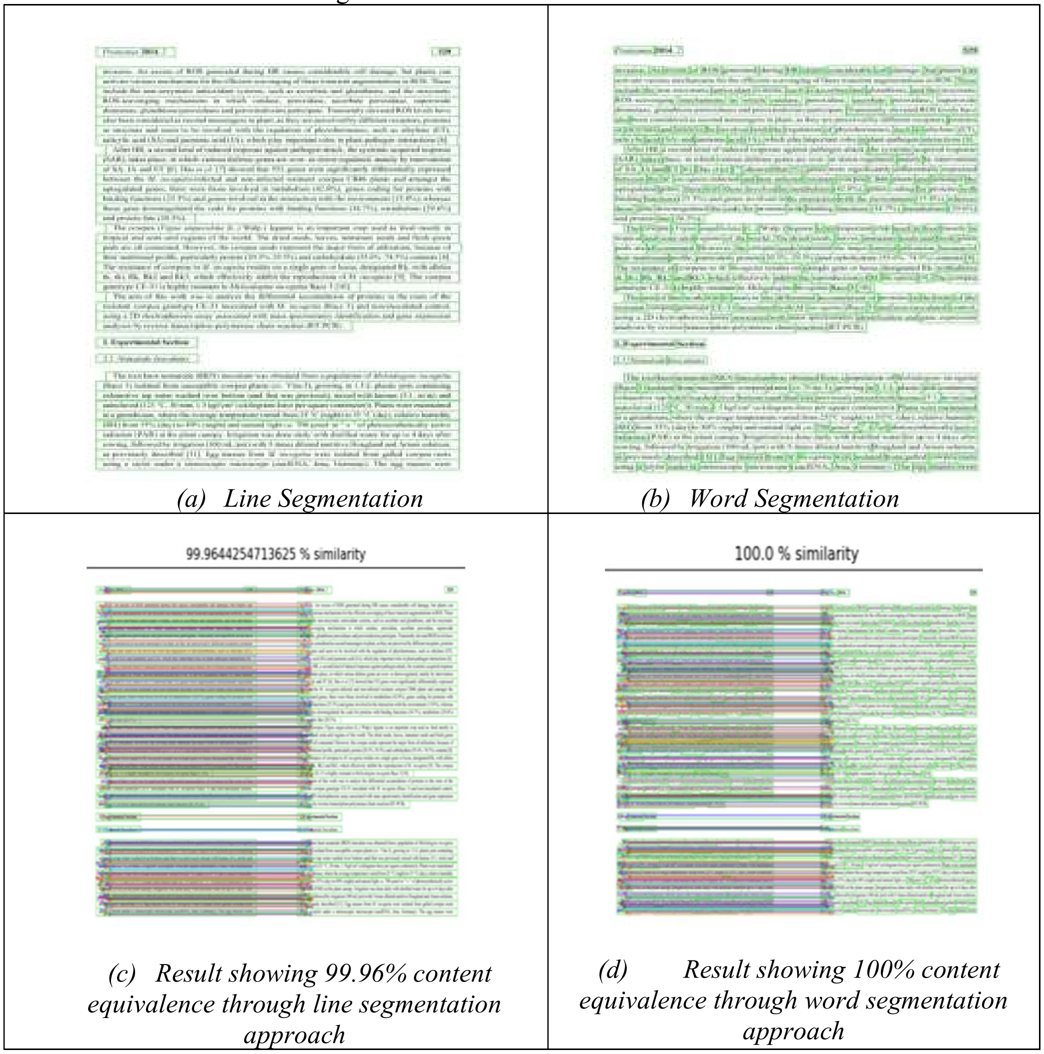
Figure 3: (a) & (b) Original image-D1 & D2 (c) & (d) Line segmented image-D1 & image-D2 (e) & (f) Word segmented image-D1 & image-D2 (g) & (h) Character segmented image-D1 & image-D2

In this experiment, figure 3. (a) and (b) are the input compressed document images obtained from PubLayNet dataset. Here, we use the terms D1 and D2 to refer document one and document two respectively. First of all, we perform line segmentation. The obtained segmentation results is depicted in figure 3 (c) and (d) for both images. In next stage, we apply word segmentation from the line segmented image. Figure 3 (e) and (f) show the word segmentation output. Finally, we apply character segmentation and obtained segmentation is presented in figure 3 (g) and (h). Later, we apply SIFT processing to match the content of different images.

In order to show the performance of the content equivalence detection, we consider different scenarios as follows:

(a) Both images are exactly same
(b) Both images are entirely different
(c) Approximately 50% content is same, and

In first experiment we have considered two similar images where we match the line and word segmented image data.



(a) Line Segmentation

(b) Word Segmentation

(c) Result showing 99.96% content equivalence through line segmentation approach

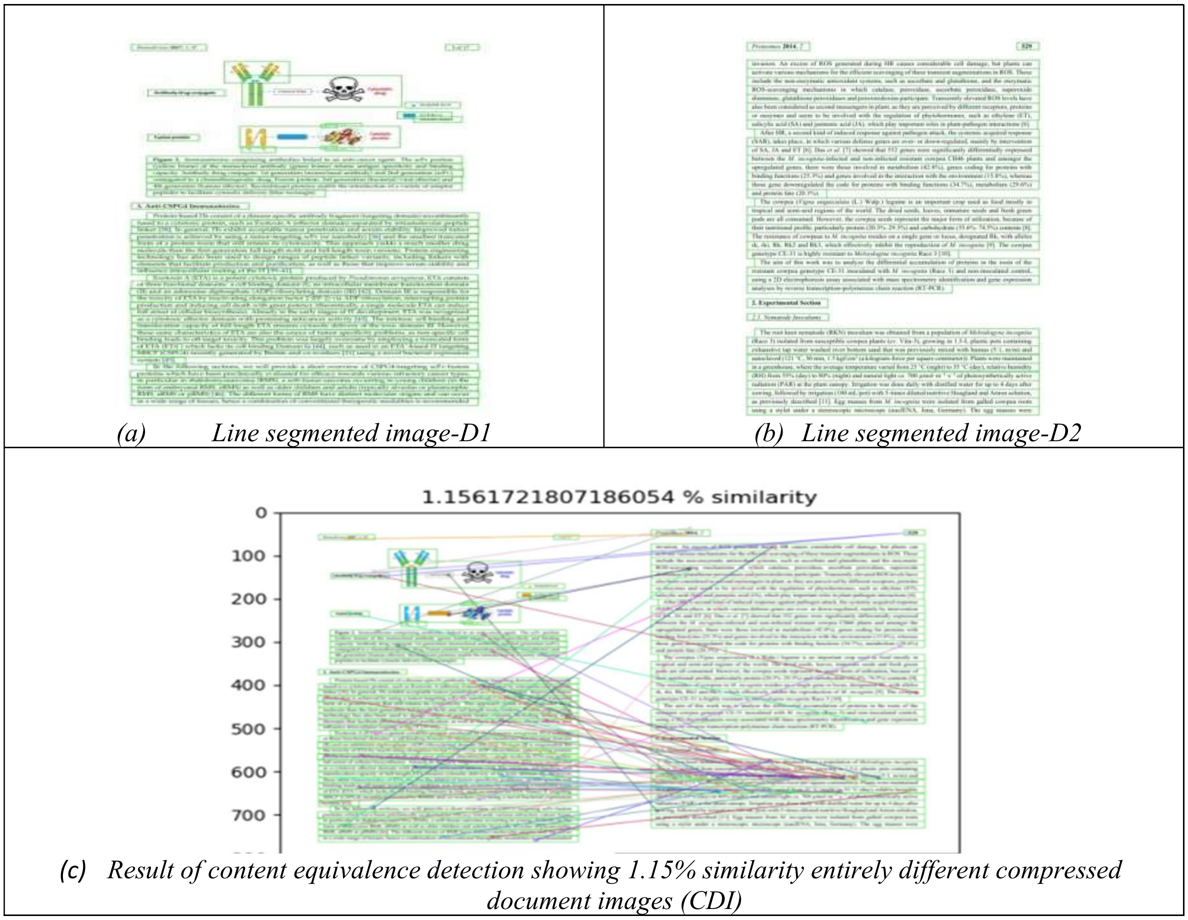
(d) Result showing 100% content equivalence through word segmentation approach

Figure 4: Content Equivalence performance analysis in Compressed Document Images (CDI)

The above figure 4 depicts content equivalence performance analysis results for exactly same images. (a) and (b) showing the results of line and word segmentation for CDI and (c) and (d) showing the results for content equivalence detection through line segmentation and word segmentation approaches respectively.

In this experiment, the accuracy of content equivalence through line segmentation is 99.96% and through word segmentation is 100%.

In next experiment, we consider two entirely different images as depicted in figure 5 given below.



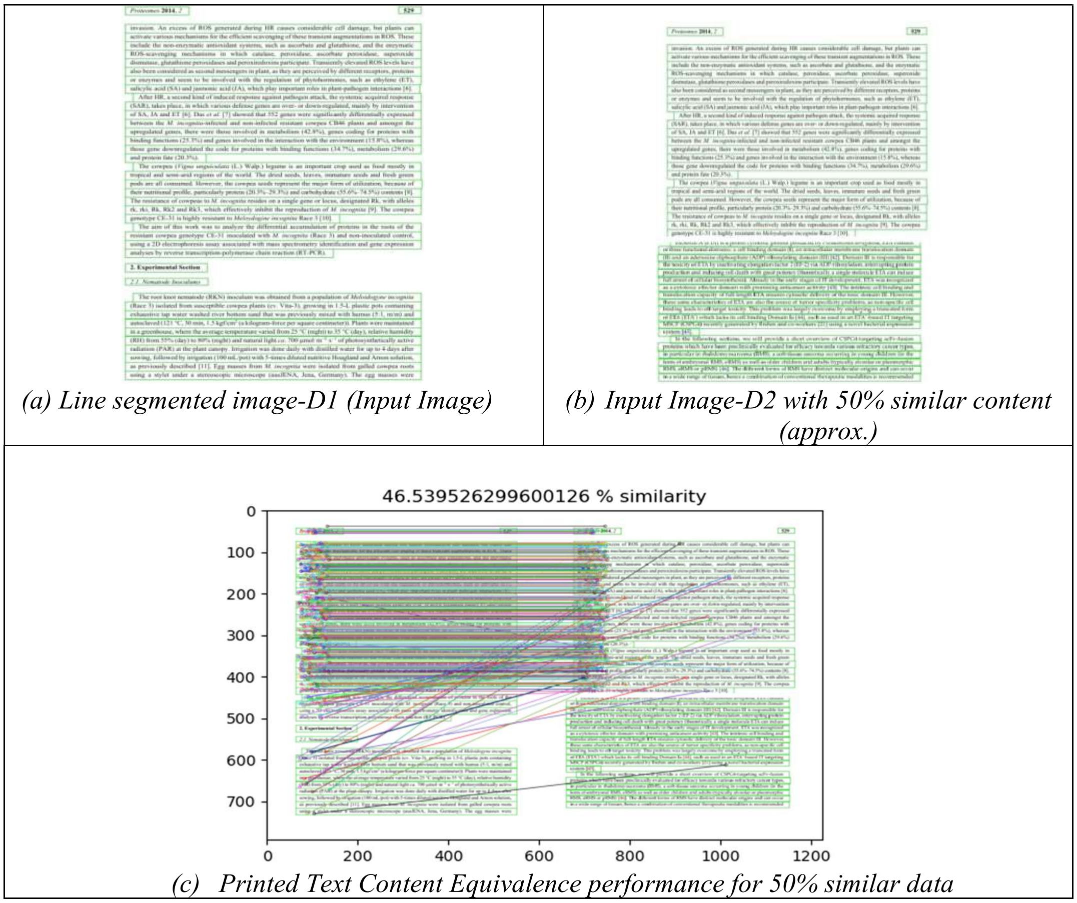
(a) Line segmented image-D1

(b) Line segmented image-D2

(c) Result of content equivalence detection showing 1.15% similarity entirely different compressed document images (CDI)

Figure.5: Outcome for entirely different compressed document images (CDI)
(a) & (b) line segmented image-D1 and D2 respectively
(c) Result of content equivalence detection showing 1.15% similarity for entirely different compressed document images (CDI)

In this experiment, figure 5 (a) and (b) are entirely different compressed document images. For these images, the accuracy of the content equivalence detection is obtained 1.15%. In the next experiment, shown in figure 6, we consider approximately 50% similar content where rest of the content is replaced by some other content in the document image. Below given figure 6 shows the obtained performance.



(a) Line segmented image-D1 (Input Image)

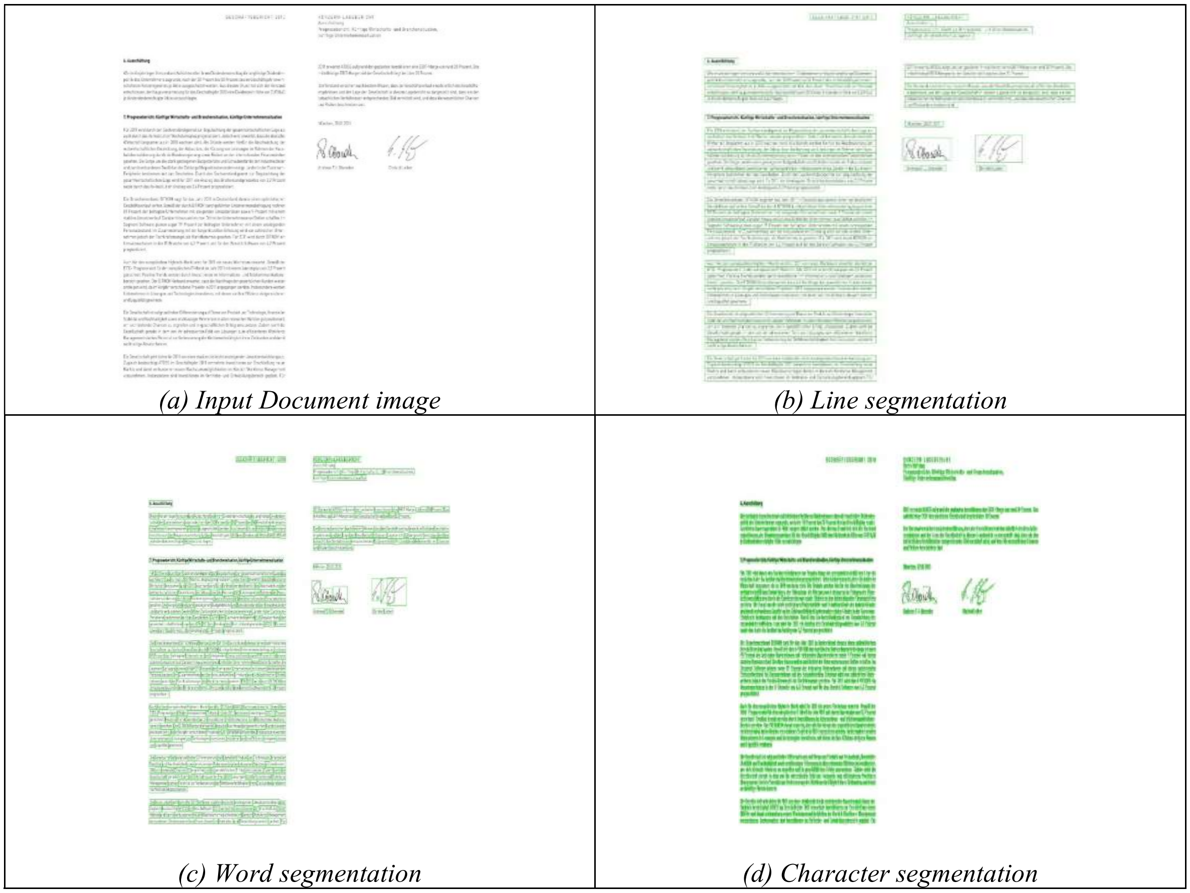
(b) Input Image-D2 with 50% similar content (approx.)

(c) Printed Text Content Equivalence performance for 50% similar data

Figure.6. (a) Line segmented image-D1 (b) Input Image - D2 with 50% similar content (approx.) (c) Printed Text Content Equivalence results for 50% similar data

Figure 6 (a) and (b) are the considered as input document images where we have considered 50% similar data between these two document images to measure the matching performance. The proposed approach shows that these two document images are 46.53% similar as depicted in figure 6 (c). These experiments show the matching ratio of two images. Based on this we can identify the similar or different document contents. In this work, we have considered 75% as the threshold to identify the similar documents.

Similarly, we measure the performance for IIIT-AR-13K dataset which is widely adopted for detecting graphical objects in various business documents. This dataset contains manually annotated bounding boxes on the objects in publically available annual reports. This dataset comprises of totally 13000 annotated page image containing five diverse categories such as natural image, logo, table, figure and signature. In this experiment also, we consider the three test cases where in first case, we compare the similar image, in second stage we compare the entirely different image and in third stage we compare the images which have 50% similar content as shown in the experiments below.



(a) Input Document image

(b) Line segmentation

(c) Word segmentation

(d) Character segmentation

Figure 7: (a) Input document image-D1 from IIIT-AR-13k dataset
(b) Line segmentation (c) Word segmentation and (d) Character segmentation

Figure 7 (b) depicts the result of line segmentation, figure (c) depicts the word segmentation result and figure (d) denotes the result of character segmentation. We extend our experimental analysis where two same compressed document images are fed to the content equivalence detection module. These two compressed document images are having same content thus; we obtain the content similarity outcome as 100%.

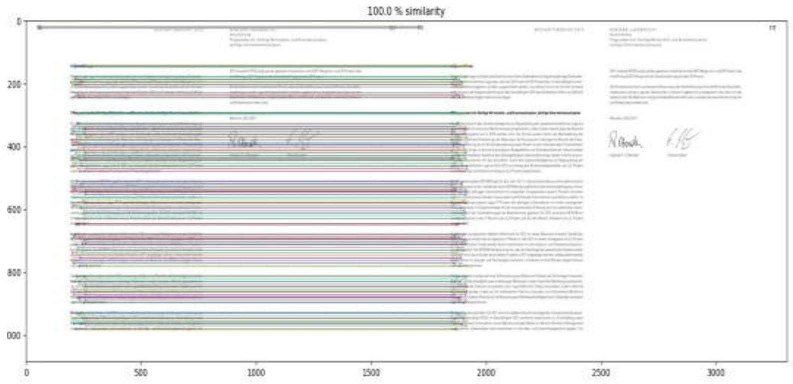
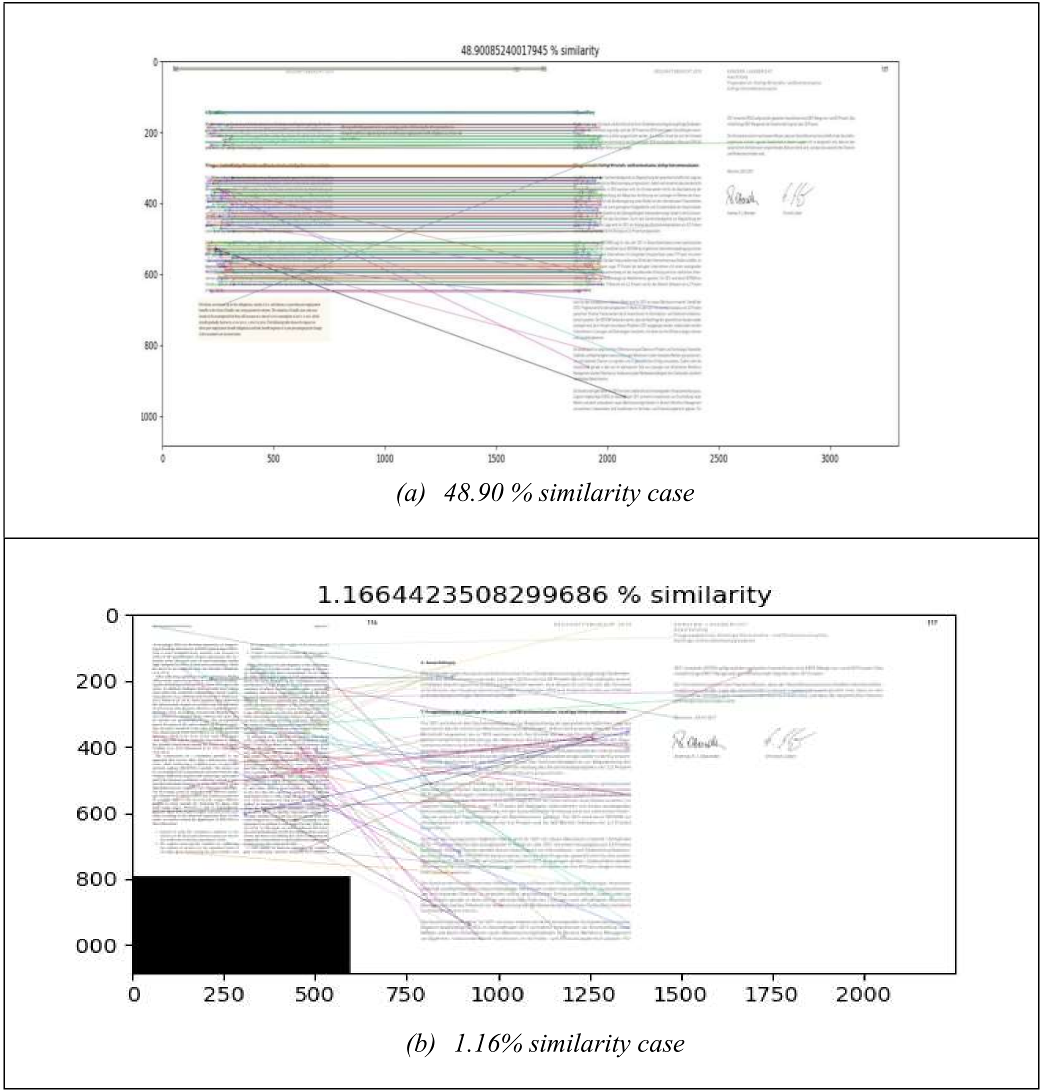


Figure 8: 100% Content Equivalence Detection Test Case for IIIT-AR-13K dataset

In next case, we process these images with almost 50% of similar data. The obtained outcome is presented in below given figure 9(a) and 9(b) for compressed document images having approximately 50% entirely different content showing 48% similarity and completely different content showing 1.166% similarity.

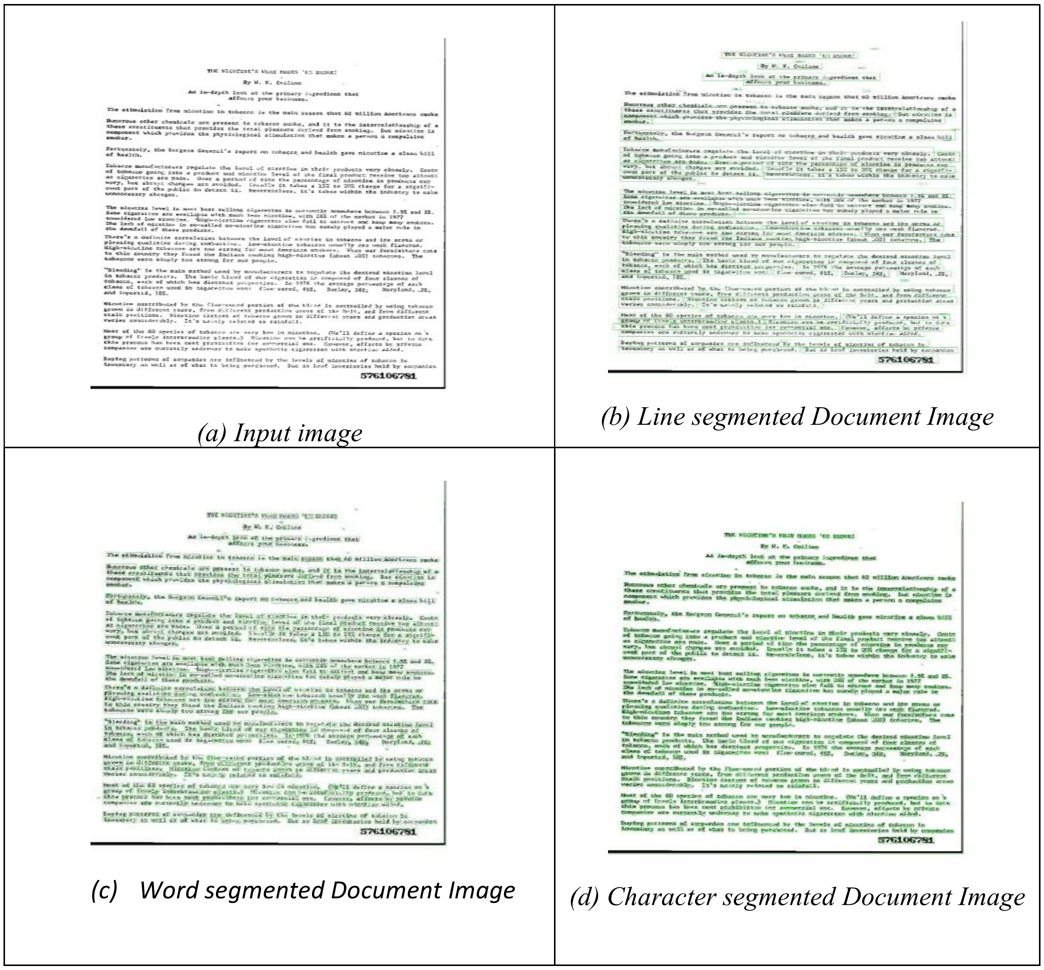


(a) 48.90 % similarity case

(b) 1.16% similarity case

Figure 9: (a) 48.90 % and (b) 1.16 % similarity cases.

Furthermore, we measure the performance for Tobacco-3482 dataset. This dataset contains total 3482 images which contain 10 categories such as Advertisements, Emails, Memos, and Scientific Report [23]. For this experiment also, first of all, we apply line, word and character segmentation process and later, content matching process is performed. Below given figure 10 depicts the outcome of line, word and character segmentation.



(a) Input image

(b) Line segmented Document Image

(c) Word segmented Document Image

(d) Character segmented Document Image

Figure.10 (a) Input Document Image-D1 from Tobacco-3482- dataset (b) Line segmented Document Image (c) Word segmented Document Image(d) Character segmented Document Image

For this dataset, we evaluate the content similarity performance for completely similar images, 50% similar content and entirely different image. Below given figure 11 depicts the outcome for the test case where we have considered completely similar images.

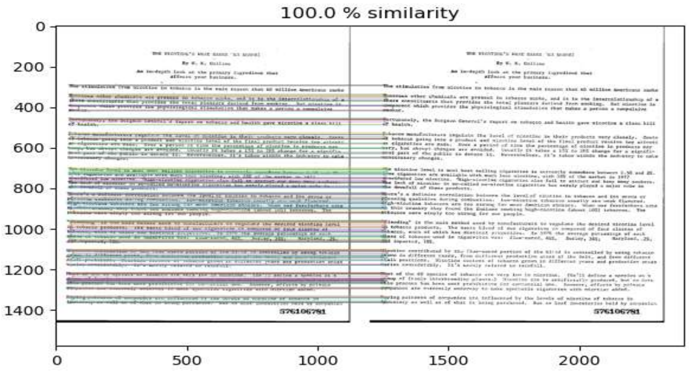
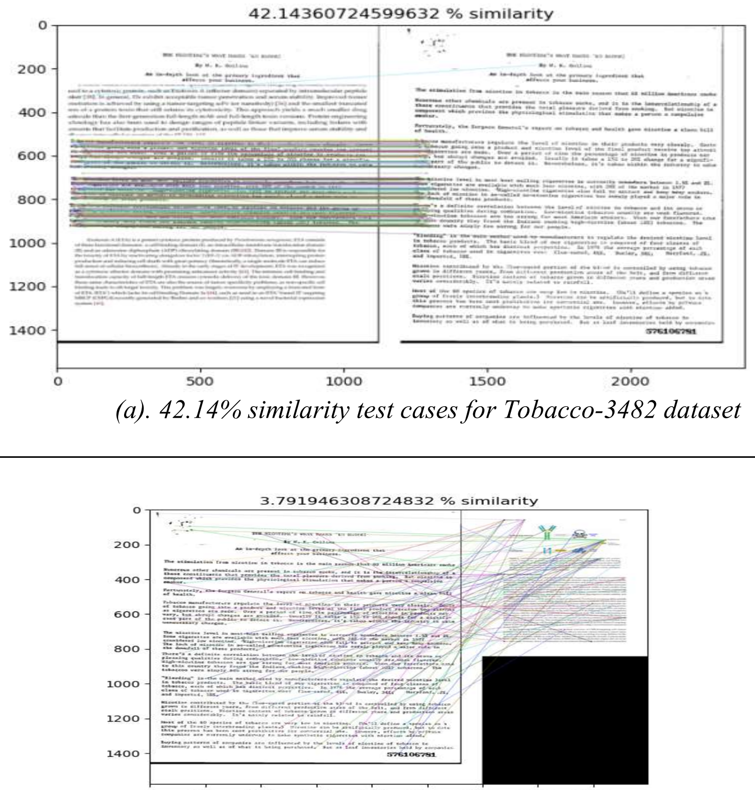


Figure.11. 100% content equivalence detection for Tobacco-3482- dataset

Further, we present the experiment for 50% similar image and entirely different images. Below given figure 12 (a) shows the 42% (almost 50%) similarity and (b) depicts the outcome for completely different images thus the similarity is obtained 3% similarity (almost 0%).



(a). 42.14% similarity test cases for Tobacco-3482 dataset

(b) 3.79% similarity test cases for Tobacco-3482 dataset

Figure.12. (a) 42.14% similarity and (b) 3.79% similarity test cases for Tobacco-3482 dataset

5. Conclusion and Future Directions

In this work, we have focused on compressed domain document image processing in content matching field. The compressed domain data processing has faster processing when compared with other schemes. Nowadays, demand of digital data has increased which require lot of space for storage, and consumes bandwidth to process the data. To overcome these issues, we adopted a novel method of performing compressed domain document image processing where rather than decompressing the complete image, some coefficients need to be extracted which can be processed for further tasks. In this work, our aim is to find the content equivalence between the given document images in compressed domain. This can serve as one of the application for plagiarism detection, answer scripts evaluation process to find matching content or copy detection, keyword matching etc. To the best of our knowledge, this work is not being reported in the literature so far. Furthermore, to obtain this, we considered JPEG encoded image data and processed it through proposed line, word and character segmentation schemes. These segmented compressed document images are considered for further processing through SIFT technique where we use Brute force matcher and K nearest neighbor technique to find the content similarity. We have considered 75% matching score as threshold to decide the content similarity between the document images in the compressed domain. The results are tested on different publically available datasets such as PubLayNet dataset, IIIT-AR-13k dataset and Tobacco-3482 dataset and it shows that the results are encouraging which justifies the proposed approach. In future, this work can be tested on document images having different font styles and sizes, degraded document images, historical document images, compressed domain handwritten document images, comparison between printed and handwritten document images and so on which imposes further challenges due to the variations in the handwriting.

REFERENCES

[1] Javed, M., Nagabhushan, P., & Chaudhuri, B. B. (2014). Direct processing of document images in compressed domain. arXiv preprint arXiv:1410.2959.

[2] Temburwar, S., Rajesh, B., & Javed, M. (2021). Deep Learning Based Image Retrieval in the JPEG Compressed Domain. arXiv preprint arXiv:2107.03648.

[3] Dutta, T., & Gupta, H. P. (2017). An efficient framework for compressed domain watermarking in p frames of high-efficiency video coding (HEVC)--encoded video. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 13(1), 1-24.

[4] Jamil, A., Majid, M., & Anwar, S. M. (2019). An optimal codebook for content-based image retrieval in JPEG compressed domain. Arabian Journal for Science and Engineering, 44(11), 9755-9767.

[5] Wang, F., Liu, F., Zhu, S., Fu, L., Liu, Z., & Wang, Q. (2019, December). HEVC intra frame based compressed domain video summarization. In Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing (pp. 1-7).

[6] Kim, Y. K., Jeon, Y. G., & Shin, S. H. (2021, February). Real-time motion detection in H. 264 compressed domain for surveillance application. In Journal of Physics: Conference Series (Vol. 1780, No. 1, p. 012032). IOP Publishing.

[7] Le, V. P., Nayef, N., Visani, M., Ogier, J. M., & De Tran, C. (2015, August). Text and non-text segmentation based on connected component features. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 1096-1100). IEEE.

[8] Augusto Borges Oliveira, D., & Palhares Viana, M. (2017). Fast CNN-based document layout analysis. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 1173-1180).

[9] Wu, X., Hu, Z., Du, X., Yang, J., & He, L. (2021, July). Document Layout Analysis via Dynamic Residual Feature Fusion. In 2021 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE.

[10] Kosaraju, S. C., Masum, M., Tsaku, N. Z., Patel, P., Bayramoglu, T., Modgil, G., & Kang, M. (2019, September). DoT-Net: Document layout classification using texture-based CNN. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1029-1034). IEEE.

[11] Wang, B., Zhou, J., & Zhang, B. (2020). MSNet: A Multi-scale Segmentation Network for Documents Layout Analysis. In Learning Technologies and Systems (pp. 225-235). Springer, Cham.

[12] Seuret, M., Alberti, M., Liwicki, M., & Ingold, R. (2017, November). PCA-initialized deep neural networks applied to document image analysis. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 877-882). IEEE.

[13] Javed, M., Nagabhushan, P., & Chaudhuri, B. B. (2018). A review on document image analysis techniques directly in the compressed domain. Artificial Intelligence Review, 50(4), 539-568.

[14] Beratoğlu, M. S. (2021). Vehicle License Plate Detector in Compressed Domain. *IEEE Access*.

[15] Liu, Q., Liu, B., Wu, Y., Li, W., & Yu, N. (2019). Real-time online multi-object tracking in compressed domain. IEEE Access, 7, 76489-76499.

[16] Jamil, A., Majid, M., & Anwar, S. M. (2019). An optimal codebook for content-based image retrieval in JPEG compressed domain. Arabian Journal for Science and Engineering, 44(11), 9755-9767.

[17] Rajesh, B., Jain, P., Javed, M., & Doermann, D. (2021, March). HH-CompWordNet: Holistic Handwritten Word Recognition in the Compressed

Domain. In 2021 Data Compression Conference (DCC) (pp. 362-362). IEEE.

[18] Sharma, A., Rajesh, B., & Javed, M. (2021). Detection of Plant Leaf Disease Directly in the JPEG Compressed Domain using Transfer Learning Technique. arXiv preprint arXiv:2107.04813.

[19] Phadikar, B. S., Phadikar, A., & Maity, G. K. (2018). Content-based image retrieval in DCT compressed domain with MPEG-7 edge descriptor and genetic algorithm. Pattern Analysis and Applications, 21(2), 469-489.

[20] Delac, K., Grgic, M., & Grgic, S. (2009). Face recognition in JPEG and JPEG2000 compressed domain. *Image and Vision Computing*, *27*(8), 1108-1120.

[21] Byju, A. P., Sumbul, G., Demir, B., & Bruzzone, L. (2020). Remote-Sensing Image Scene Classification With Deep Neural Networks in JPEG 2000 Compressed Domain. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(4), 3458-3472.

[22] Zhong, X., Tang, J., & Yepes, A. J. (2019, September). Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1015-1022). IEEE.

[23] Mandivarapu, J. K., Bunch, E., You, Q., & Fung, G. (2021). Efficient Document Image Classification Using Region-Based Graph Neural Network. arXiv preprint arXiv:2106.13802.