Improving Performance of DOM in Semi-structured Data Extraction Using WEIDJ Model

Ily Amalina Ahmad Sabri, Mustafa Man
School of Informatics and Applied Mathematics Universiti Malaysia Terengganu, Terengganu, Malaysia

Article Info

Article history:

Received Jun 9, 2017
Revised Nov 20, 2017
Accepted Dec 11, 2017

Keywords:

DOM (Document Object Model)
JSON (JavaScript Object Notation)
Information Extraction
Semi-structured Data.

ABSTRACT

Web data extraction is the process of extracting user required information from web page. The information consists of semi-structured data not in structured format. The extraction data involves the web documents in html format. Nowadays, most people uses web data extractors because the extraction involve large information which makes the process of manual information extraction takes time and complicated. We present in this paper WEIDJ approach to extract images from the web, whose goal is to harvest images as object from template-based html pages. The WEIDJ (Web Extraction Image using DOM (Document Object Model) and JSON (JavaScript Object Notation)) applies DOM theory in order to build the structure and JSON as environment of programming. The extraction process leverages both the input of web address and the structure of extraction. Then, WEIDJ splits DOM tree into small subtrees and applies searching algorithm by visual blocks for each web page to find images. Our approach focus on three level of extraction; single web page, multiple web page and the whole web page. Extensive experiments on several biodiversity web pages has been done to show the comparison time performance between image extraction using DOM, JSON and WEIDJ for single web page. The experimental results advocate via our model, WEIDJ image extraction can be done fast and effectively.

Corresponding Author:

Ily Amalina Ahmad Sabri,
School of Informatics and Applied Mathematics
Universiti Malaysia Terengganu, Terengganu, Malaysia
Email: ilylina@yahoo.com

1. INTRODUCTION

Data integration is considered as one of the hot issues to be solved especially in integrating unstructured data with multiple types and formats and stored in different location [1]. The integration between the structured and unstructured data is concerned by certain organizations in terms of their benefits to retrieve valuable information and knowledge [2]. In data integration, imposing a single global schema for all users can seriously interfere with their individual work, as the autonomy of information receivers is violated. The autonomy of information receivers implies that integrated information use must be non-intrusive [3]. This means that users should not be forced to adapt to any standard concerning the structure and meaning of the data they request. The desired kind of data integration can thus be characterized by the optimal fitness of the supplied information for a certain purpose, concerning the organization, presentation, and semantics. To state this differently, the integrated information that is provided has all the relative qualities required by a particular user in a specific task. The data resides in different forms, ranging from unstructured data (USD) in file systems to highly structure in relational database systems. We need to consider three types of data such as structured data (SD), unstructured data (USD) and semi-structured data (SSD) as shown in Figure 1 [4].

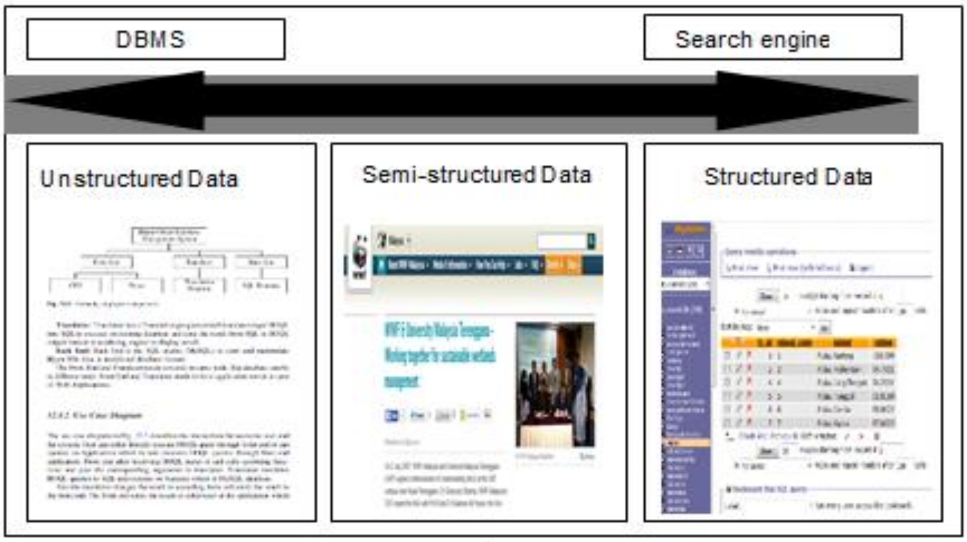


Figure 1. Types of Data

Data extraction is a part of data integration process. It is the process to extract useful information based on user required information. Extraction of the information from web is called as “Web Data Extraction”. It allows user to analyse semi-structured data from different views and arranged in structured form such as tabular format. Extraction and analysis information from web pages is an excited research area in the data extraction field. Internet has made the *world wide web (www)* as an ocean of information that provide users to collect and analyse information.

Web is the largest platform that large pool of information source for people. The various information that are usually available on the web are advertisement, navigation, contact information and decoration. The miscellaneous information found on any web pages are seldom related to valuable information. It is called as noisy. Furthermore, each web page has multiple topics that are not related to each other. Most web applications developed with recent progress in computing technologies also contain multimedia data such as images, graphics and audio have increased over the past several years [5]. Recently, detecting or extracting certain information or web page content becomes user priority [6]. Multimedia contents should be just easy to retrieved and accessed as alphanumeric data by user.

Human perception is the most important thing that we need to consider to make user feel comfortable and easy to use a software. Many software developed using various techniques. Document Object Model or known as DOM is a technique that can structure web page contents into details with each html tags in tree structure. The majority web pages are written in html compare to xml format. The main objective of this research is to provide user the ease of understanding to structure of web page contents compare to complicated technique. Naturally, mankind understand better from the visual presentation than text presentation. When a web page is presented to user, visual view can help the user to divide web page into several part.

In previous works [7, 8], we discussed the experimental process of image extraction based on DOM and JSON. The performance of information extraction based on previous study shows DOM is faster than other approaches. However, DOM consumes large amount of memory when the html structure becomes large. The large usage of memory will affect the operational speed. Thus, we present Wrapper for Image Extraction using DOM and JSON (WEIDJ) model in details to extract images from a web page. Every web page is made of their own structure includes main topic, related topics, additional information, advertisement, contact information, image, audio and video file. Web pages provides large pool of information for people. These information can be used for beneficial purpose. This paper proposes a new tool to extract images that will demonstrate better performance than existing methods such as DOM and other conventional method.

In this research, we will highlight the extraction problem from the user's perspective with regards to search of multimedia information. The initial process of extraction starts with description of targeted web page which is provided via query interface. Our proposed model, WEIDJ aims to improve the processing time for loading images, the accuracy of the extraction process and it is more efficient and lightweight than manual process. This is because manual process of extraction is time consuming.

The remainder of this paper is organized as follows: The related works and applications are discussed in next section. Special interest integrating DOM and JSON has been focused in extracting information from web page in WEIDJ Model. Section Results demonstrates the experimental result of this paper in extracting images using three approach. Finally, we offer our conclusions and plans for future works in Conclusion.

2. RELATED WORKS

Web data extractions are applications that used to extract semi-structured data from web page. Usually the extraction process involved web data extraction system and web source. The system will interact with the web page and extract semi-structured data. The extracted contents may contain various elements of multimedia data such as audio, video, text and image. After extraction process, the data will be stored in temporary folder or directly store into database. These system have been developed to assist human in a wide range applications. The advantages of web data extraction systems are it can collect meaningful data efficiently and decrease human effort in structured way. There are many discussions from different perspectives about scientific methods and techniques. The design and implementation comes from various disciplined such as machine learning, natural language processing and logic.

2.1 Document Object Model (DOM)

The Document Object Model (DOM) is a programming API for html and xml documents. People can create and build documents using DOM. Besides that, this model can be used to manipulate elements and contents of html and xml documents such as add, modify or delete. Narawade, et al. [9] developed page level data extraction system using DOM tree. There are two types of technique for data extraction; online and offline mode. Online mode is applied in real time extraction but offline mode is vice versa. There are three stages web page renderer, section selector, and pattern generator. The system will extract the content dynamically from the different structured web pages such as blogs, forums, articles and etc. DOM tree structure has been applied for content extraction in order to obtain better representation of the data format.

Sangeetha [10] proposed a tool that can process the Resource Description Framework (RDF) Based Search and DOM Based Search to extract the relevancy data. RDF is used along with DOM to give user query answer precisely. The DOM segment fusing algorithm is used to analyze and fuse the extracted information from web.

Mehta and Narvekar [11] proposed and redesigned basic DOM approach for content extraction to make it applicable for different web page structures such as blogs, forums and articles. The tool can extract information based on two different searching methods; Runtime generated list and Stored URL list.

2.2 JavaScript Object Notation (JSON)

The development of web applications has become attractive disciplined in the web environment. The use and composition of different of API technology is very important and influent applications. This issue need to be deal to discover JSON approach based on the web. In recent years, a new technology, JSON based on web applications has been spreading the web environment. JSON is a lightweight data-interchange format. It is self-describing and easy to understand. It is easy for humans to read and write. It is also easy for machines to parse and generate data and very efficiency for data extraction and query retrieval [12]. JSON, DOM and XML are different technologies that have been developed to solve different problems. They are designed for different purpose. Table 1 discussed different technologies of JSON, DOM and XML.

Table 1. Comparisons of Different Technologies

JavaScript Object Notation (JSON)	Document Object Model (DOM)	eXtensible Markup Language (XML)
- JSON is a lightweight, text based format can be used for data interchange format. - it is human readable	- DOM is used for manipulating and representing html and xml documents.	- XML is designed to store and transport data. - It is readable for human and machine. - XML is more complex compare to JSON.

The work by [13-15] proposed Ducky focused on the data extraction. Ducky, as a semi-automatic system using JSON approach which can extract data from web sources and represent all the information in structured format. In Ducky, the configuration file will be defined as a well-formed JSON document and contains several parameters used for data extraction process. In this file, data management and rules are been specified as it is to be readable and to be expressed very simply by using JSON format.

Wang [16] proposed recursive algorithm to translate XML and JSON objects in serializing forms based on the multi tree data structure. This research is motivated due to XML and JSON are widely used in application development. In this experimental work, JSON is analyzed as string arrays. This is the reason why JSON is faster than DOM-style XML objects.

JSON has been widely used in the actual development of web applications maybe in similar functions but in different domain and applications. Besides that, web wrapper is a program that can extract information from web sources and translates them into relational form. Wrappers can apply JSON and DOM in their functionality.

2.3 Wrappers

Nowadays, many systems for data extraction from web pages have been developed [17]. A traditional approach is to write specific programs called as “extractor” or “wrappers” is developed to extract the contents of the web pages based on certain criteria. A survey that offers a rigorous taxonomy to classify web data extraction systems has been presented by Laender, et al. [18].

Alarte, et al. [19] proposed a method that can remove irrelevant information from web template. DOM tree is used to analyse the similarity between a collections of a webpage that are detected using a hyperlink analysis. Abidin, et al. [20] introduced an automated unstructured data capturing for structured storing that deals with multimedia data. This research stated that the unstructured data such as multimedia files, documents, spreadsheets, news, emails, memorandums, reports and web pages are difficult to capture and store in the common database storage. Even there are many tools and techniques that proved to be successful in transforming unstructured data to valuable information but it simply do not work when it comes to unstructured or semi-structured data.

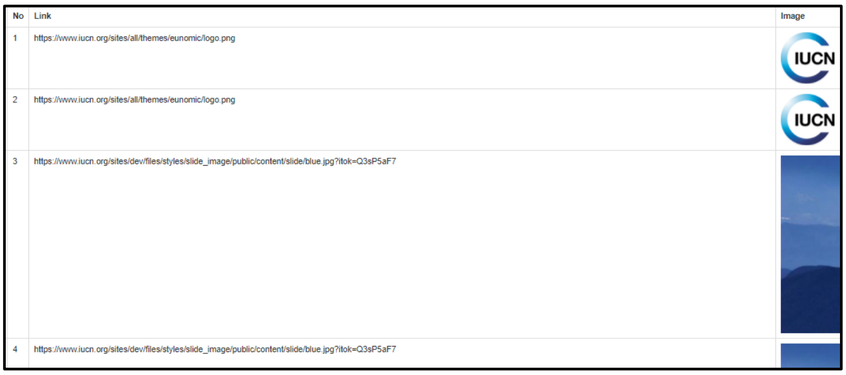
Web wrapper is a procedure that might implement several of techniques in their algorithms. The goal is to seek and find data required by human users which is extracting unstructured or semi-structured data from web sources. Finally the data will be transformed into structured data for multi-purpose. Lately, the problem of extracting information from unknown sites is getting much attention, but the only conclusive results are regarding unstructured or semi-structured documents. The theme of Web Data Extraction is covered by a number of reviews. Ferrara et al. [17] presented a survey tools and techniques overview for Web Data Extraction. The goal of this survey is to provide a structured and comprehensive overview of the research of Web Data Extraction. In 2008, a relevant survey on information extraction has been discussed by [21]. This paper believed that the automatic extraction of information from unstructured sources has opened up new avenues for querying, organizing, and analyzing data by drawing upon the clean semantics of structured databases and the abundance of unstructured data. Flesca, et al. [22] surveyed approaches, techniques and tools for extracting information available on the Web.

In Table 2, we summarize applications that have been developed for semi-structured data extraction. However, there is not much research done pertaining to data extraction for multimedia data such as image, audio and video.

Table 2. Comparative Analysis of Data Extraction

Author	Mode	Semi-structured Data			
		Text	Audio	Video	Image
Raza & Gulwani [23]	Online	/			
Waghmare & Maral [24]	Offline	/			
Narawade et al. [9]	Online Offline	/			
Song, Sun, & Liao [25]	Online	/			
Bhardwaj & Mangat [26]	Online	/	/	/	/
Kadam & Pakle [27]	Online	/			
López et al. [28]	Online	/			
Abidin, Idris, & Husain [20]	Online	/	/	/	/

Document Object Model (DOM) can be applied directly to find the required information from html documents. Abidin, et al. [20] constructed DOM tree structure on the first step. Then, unnecessary nodes such as script, style need to be filtered. Classification process is important to search classes of multimedia data. Data for media will be recognized when the parser found word “src=” in the data structure. Finally multimedia data can be extracted. However, it has been found that it requires large amount of processing time during the extraction process of web pages that consist of large html structure. Besides that, the extraction process will extract all images without consider repetitive files as show in Figure 2. WEIDJ Model is proposed to overcome the limitations of DOM model in extracting images.



No	Link	Image
1	https://www.iucn.org/sites/all/themes/eunomic/logo.png	IUCN
2	https://www.iucn.org/sites/all/themes/eunomic/logo.png	IUCN
3	https://www.iucn.org/sites/dev/files/styles/slide_image/public/content/slide/blue.jpg?itok=Q3sP5aF7	
4	https://www.iucn.org/sites/dev/files/styles/slide_image/public/content/slide/blue.jpg?itok=Q3sP5aF7	

Figure 2. Repetition Image

3. WEIDJ MODEL

This research proposes the information integration model for data extraction focus on semi-structured data such as image, video, audio and text by using Document Object Model (DOM) and JavaScript Object Notation (JSON). Based on the proposed integration models, a mediator tool called as a wrapper will be developed as experimental to extract semi-structured data from heterogeneous source like web pages. Experiments will be conducted on Setiu wetlands web site and biodiversity web pages dataset for testbed. The aim of this work is to find extraction approach that can identify and extract images. In this paper, we propose WEIDJ model to extract images from a web page as shown in Figure 3. It also mines images information and focuses on arranging the extracted data in a tabular format. This tool aims speed and performance of image extraction. Lots of applications worked on extracting information then arrange them into structured format [29, 30]. Mining information records in data regions plays important role. It becomes easier to extract data from data regions because it contains useful data such as images, text, audio and video. A technique is needed for the process of mining data area. This model proposed DOM tree to mine data regions in web page.

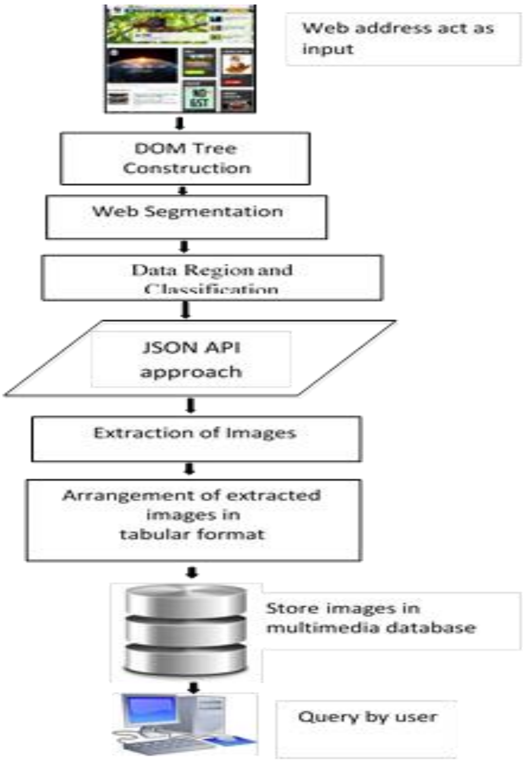

Web address act as input
DOM Tree Construction
Web Segmentation
Data Region and Classification
JSON API approach
Extraction of Images
Arrangement of extracted images in tabular format
Store images in multimedia database
Query by user

Figure 3. WEIDJ extraction model

3.1 DOM Tree Construction

Initially for information extraction from web page, web address or unified resource locator (*url*) of web page is required. When user inputs the *url*, single page is the extraction target but when user input multiple *url*, the extraction target aims at page-wide information. This paper only focuses on record level extraction task. This approach focuses on the image extraction from surface web. Most website are developed based on the html format rather than xml. Basically DOM will be able to define and manipulate html documents into a tree structure [9]. It defines the logical structure of documents and the way it can be accessed and manipulated. It is also known as node tree. Everything in a web page contains html tags and elements, text nodes and others.

In this research, DOM is applied in html documents to form a web page into a pattern tree structure. Although objects in the DOM tree can be manipulated using certain methods but in this case, we transform the structure of web page into tree structure to recognize data regions. This process is important because data region consists useful information that is will be retrieved based on html element.

Every web page is developed using html element. Each module of html element is useful for several data records. This module discovers the whole element in web documents such as <html>,<table>,<div>,<img>,<form>,<video> and so on. All the tags will be extracted from root node to its child node in designing web page. So, tags can be recognized data region by using html element.

3.2 Data Region and Classification

Multimedia data such as images, video clips, animations, graphics and audio have increased rapidly over the past several years. Users have begun to expect that multimedia contents should be easily to access. They want to find relevant images that appear in web page, see video clip related to text articles they read and listen to the audio. It is important to provide integrated access to diverse types of multimedia semi-structured data stored in disparate data sources. Many web data extractors today deal with multimedia data. Data classification is important as it categorized data based on required need. The class of different objects will be identified in data classification.

Classification of data patterns is important for data extraction from the web page. Figure 4 illustrates the process of data extraction and classification. There are four classes that have been identified; image, text, audio and video. These multimedia data will be identified when the parser has found the word “src=” in the data structure during the extraction process. This is a keyword for multimedia data source reference to locate the source data that has been used.

When location of the required source is determined, the parser will identify its data type. Table 3 shows data source for multimedia data that contains in html documents.

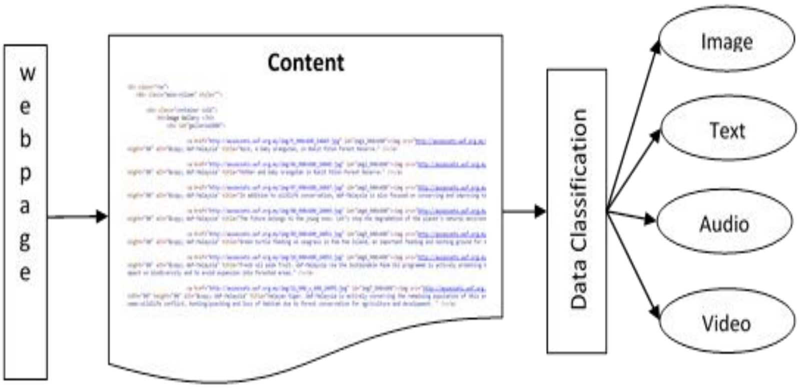


Figure 4. Data classes

Table 3. Data Source of Media

Type	‘src’- source reference
Text	<a><p> <font><size>
Image	“src=*.jpg,gif,png,bmp”
Audio	“src=*.wav,mp3,raw,midi”
Video	“src=*.flv,wmx,mp4,avi”

3.3 Content Structure Development Process using Visual Segmentation

A visual segmentation is developed using each leaf node as an object. Visual segmentation has been proposed because it is easier to understand the structure of web page visually compared to texting in details. This segmentation is important to check whether there are information that are required in each block. When conducting the experiment on image extraction using DOM and JSON method, it has been found that not all images can be extracted. As a solution, each block must be checked whether they have images or not. Figure 5 shows an example of visual segmentation of layout structure for www.wwf.org.my.

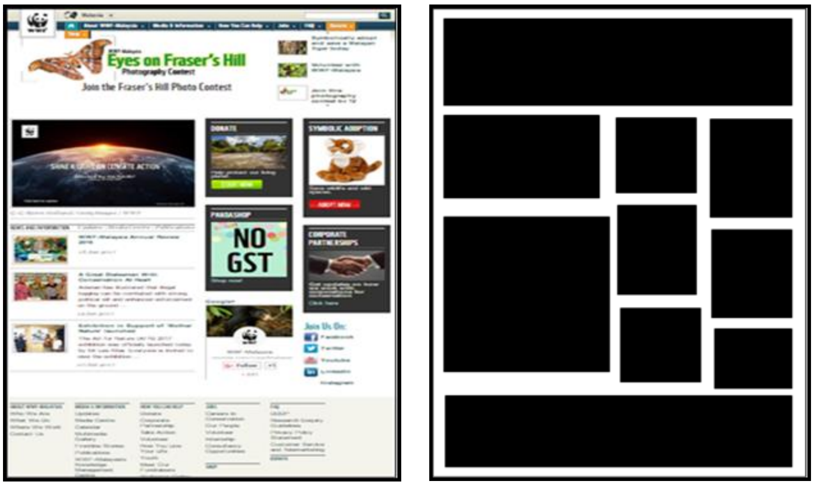


Figure 5. Visual segmentation of layout structure

In this level, we aim to find all suitable visual block contained in the current web page. Basically, every node in the DOM tree can be presented as a visual block but nodes such as <TABLE> and <P> are not suitable to be represented as a single visual block. This is because they are commonly used for organization purpose. Several rules are considered in order to extract the visual block as below:

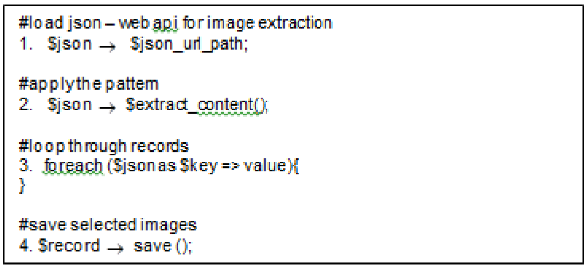
Tags cue such as <hr> usually displayed as a horizontal rule in visual browser. If each DOM node contain this tag, then we will partition the section.

If a DOM node has different background colour between from one of its child node. It will not be divided.

When appropriate blocks are extracted, the rest of invalid nodes are ignored. Separator can be used as indicator to divide different section within a page. These visual blocks segmentation is applied to check every single multimedia element so that all required information can be retrieved.

3.4 JavaScript Object Notation Data (JSON)

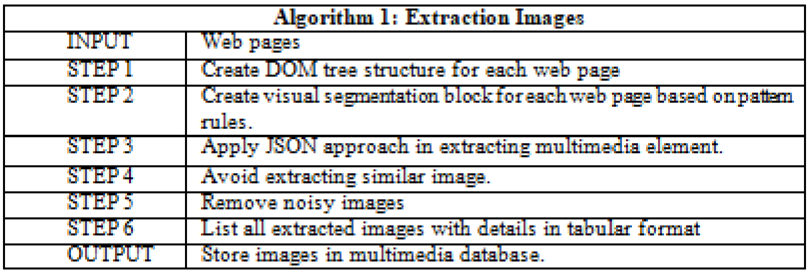
JSON is a syntax for storing and exchanging data which originates from Java Script Object Notation. The advantage of JSON is that it is an open-standard format that uses human readable text to transmit data objects. [Yusof and Man [12], 31] stated that JSON is the best choice for storage and speedy in query information. The output can be ranged from simple to complex structure and highly nested. Figure 6 shows how JSON data set treats in column. $json_url_path is used as constructor to inform the JSON data set to include the nested structures of JSON object. In this particular example people need to input the *url* as json path. We specify the path using ‘src’ value which is simply find the information of image from the image nested structure.



```
#load json – web api for image extraction
1.  $json → $json_url_path;

#apply the pattern
2.  $json → $extract_content();

#loop through records
3.  foreach ($json as $key => value){
}

#save selected images
4. $record → save ();
```

Figure 6. Basic step JSON of WEIDJ programming

Figure 7 shows WEIDJ algorithm that has been proposed in our research. This algorithm apply DOM to structure the html documents in hierarchical structure. Then visual segmentation blocks are developed for html page to check available element of images in each blocks. This approach is important to make sure all required images can be extracted without failure. JSON environment approach is applied in extracting images. Other rules like filtering similar filename and removing noisy images such as logo and button are to be considered to make sure images that has been extracted are valuable information. At last, final images and their details will be display in tabular format before users can store them into multimedia database.



Algorithm 1: Extraction Images	
INPUT	Web pages
STEP 1	Create DOM tree structure for each web page
STEP 2	Create visual segmentation block for each web page based on pattern rules.
STEP 3	Apply JSON approach in extracting multimedia element.
STEP 4	Avoid extracting similar image.
STEP 5	Remove noisy images
STEP 6	List all extracted images with details in tabular format
OUTPUT	Store images in multimedia database.

Figure 7. The WEIDJ algorithm

Bhardwaj and Mangat [26] discussed that elements of tags whose size greater than 120000 have height value for extraction. As example, (size of tags 300x400 or 400x300). In contrast to our experimental testing on Youtube channel (Figure 8), we found that size for each image of video has been set to 196x110 and size for each icon has been set to 88x88. This can be used as a guideline that we must consider tag size below than 120000 for data extraction. For image extraction rules, we set that image tags whose size smaller than 50x50 will not be extract and be considered as noisy information. The rules for avoiding extraction repetitive files of images also will be added. The reason why image size larger than 50x50 must be considered to be extracted is because in certain web page, there are valuable images that have been set to size 70x70 as shown in Figure 9 such as in biodiversity explorer web page, images have been set to 70x70.

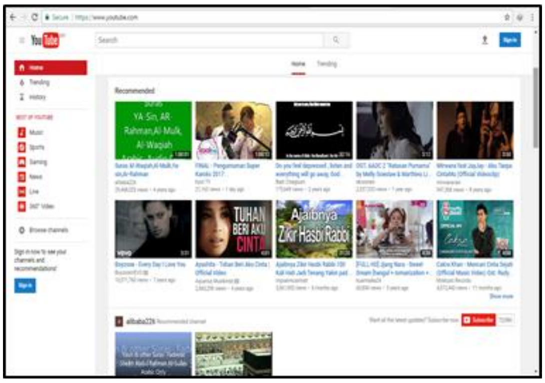
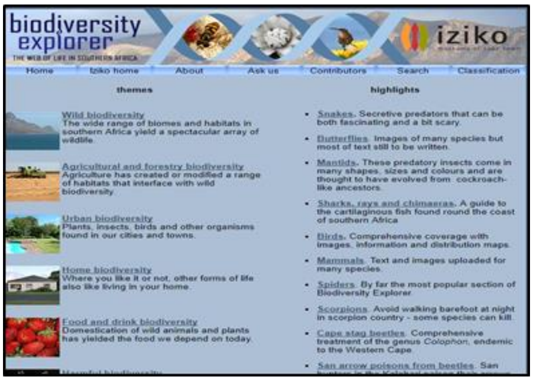


Figure 8. Youtube channel



themes

Wild biodiversity
The wide range of biomes and habitats in southern Africa yield a spectacular array of wildlife.

Agricultural and forestry biodiversity
Agriculture has created or modified a range of habitats that interface with wild biodiversity

Urban biodiversity
Plants, insects, birds and other organisms found in our cities and towns.

Home biodiversity
Where you like it or not, other forms of life also like living in your home.

Food and drink biodiversity
Domestication of wild animals and plants has yielded the food we depend on today.

Harmful biodiversity

highlights

- Snakes. Secretive predators that can be both fascinating and a bit scary.
- Butterflies. Images of many species but most of text still to be written.
- Mantids. These predatory insects come in many shapes, sizes and colours and are thought to have evolved from cockroach-like ancestors.
- Sharks, rays and chimaeras. A guide to the cartilaginous fish found round the coast of southern Africa
- Birds. Comprehensive coverage with images, information and distribution maps.
- Mammals. Text and images uploaded for many species.
- Spiders. By far the most popular section of Biodiversity Explorer
- Scorpions. Avoid walking barefoot at night in scorpion country - some species can kill.
- Cape stag beetles. Comprehensive treatment of the genus *Colophon*, endemic to the Western Cape
- San arrow poisons from beetles. San

Figure 9. Biodiversity Explorer Web Page

4. RESULTS

The main motivation for this paper is to extract images and mine image details such as images, links of images, size of images and store selected images in single multimedia database. In ideal scenario, if people want to save image, it can be extracted manually. People can extract them manually by saving each image as many as possible. But how to extract and mine images manually if there are large of volume images? Therefore, another solution must be developed to extract images automatically to reduce time consuming. The important part of extraction system is database of records. This is because the records that have been extracted and saved can be used for beneficial purpose such as documentation, analyze reports and so on.

A data extraction engine need to be able to extract all the data that are required from web page. We need to define the unified resource locator (*url*) of the web page where the objective data is located. This is initial process to extract data from a specific web page. Figure 10 shows Setiu Wetlands web page namely WWF- Malaysia. WWF stands for World Wide Fund for Nature. It was formerly known as the World Wildlife Fund but adopted its current name to show that it also works on other environmental issues, and not just wildlife.

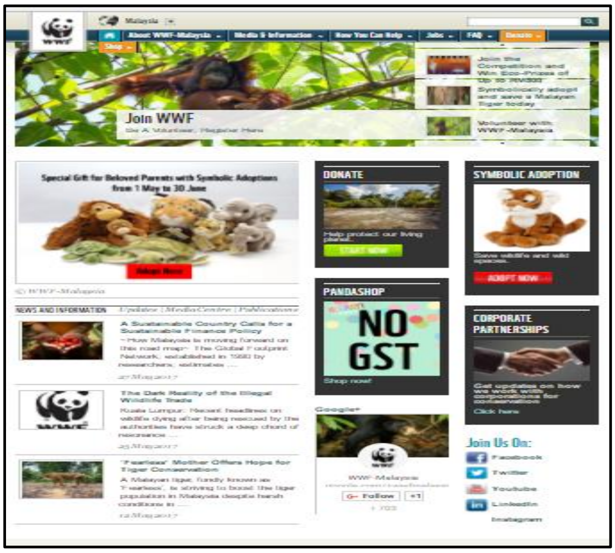
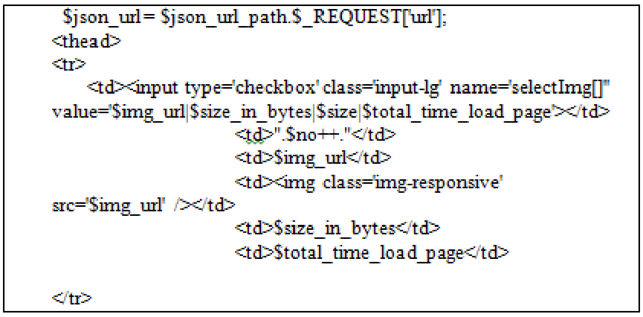


Figure 10. A snapshot images of the WWF web page

In this experimental works, sample of wwf web pages were taken and the content extraction experiments was performed on the sampled data using html source file. This file contains images information that are going to be extracted. We can see that within brackets '{' and '}' there are list of commands that consist of image and image url from sample extractor specification of file. Most of the images in .jpg format file. Our wrapper are able to extract images in various format such as .jpg, .gif, .bmp and others. Looking at Figure 11, it shows extracted information which is arranged in structured way. The syntax shows the information will be display in tabular format. The extraction process in this example is performed by table definition. The initial command $json_url fetches the contents of the source file whose *url* is given in ['*url*']. After the file has been fetched, the contents will be specified into specific criteria such as $no, $img_url, image, $size_in_bytes and $total_time_load_page. The extraction results will be represented in tabular format.



```
 $json_url= $json_url_path.$_REQUEST['url'];
<thead>
<tr>
    <td><input type='checkbox' class='input-lg' name='selectImg[]"
value='$img_url|$size_in_bytes|$size|$total_time_load_page'></td>
              <td>".$no++."</td>
              <td>$img_url</td>
              <td><img class='img-responsive'
src='$img_url' /></td>
              <td>$size_in_bytes</td>
              <td>$total_time_load_page</td>

</tr>
```

Figure 11. The extracted information in JSON format

In this paper, we have worked on the dataset called Science and Technology Resources on the Internet, "Biodiversity Web Resources", which is having 43 online databases [32]. This experimental focus on five web pages on single page to form as input in image extraction. Table 4 shows web address for selected web pages and their domain that have been used for extraction purpose. The dataset is composed of a collection of web domain with different page structures. The different page structures allows us to study the performance of image extraction in different contexts.

Table 4. Domain for Web Pages

url	Unified Resource Locator (URL)	Domain
1	http://tolweb.org/tree/	Tree of Life Project (ToL)
2	http://www.amnh.org/exhibitions/permanent-exhibitions/biodiversity-and-environmental-halls/hall-of-biodiversity	American Museum of Natural History (AMNH) Hall of Biodiversity
3	http://ocean.si.edu/	Ocean Portal: Smithsonian Institution
4	http://www.iucn.org/	International Union for Conservation of Nature
5	http://www.endangeredspeciesinternational.org	Endangered Species International

Table 5 shows few familiar sections that are total extracted images in single page and time required to complete each extraction process. This table summarizes the results of the performed experiments. This extraction involves three approach DOM, JSON and WEIDJ. First column contains number of unified resource locator (*url*) that can be refer from Table 4. Column total of images shows the number of images in single page. Column image extracted shows the number of successfully images that has been extracted. Column time required in second represent time processing for extracting images in DOM, JSON and WEIDJ approach.

Table 5. Extraction Results

url	Total of images	Image Extracted DOM	Time required in second	Image Extracted JSON	Time required in second	Image Extracted WEIDJ	Time required in seconds
	3	3	1.892	3	1.8814	3	0.004
	27	24	21.5508	25	3.42	26	0.007
	13	9	13.35	9	9.58	9	0.006
	15	13	8.23	11	6.34	11	0.006
	22	0	59.63	22	11.7421	22	0.004

An experimental has been done to extract image using different approach. This experimental is important to identify the characteristics of DOM and JSON such as the ability to extract images and time taken for extraction process. Figure 12 and Figure 13 shows the experimental of performance image extraction using WEIDJ approach. The extraction of semi-structured data, images involves five web pages which have different of structure. It shows number of images can be extracted and time taken for extraction. The graphs show significant differences in image extraction. Time taken for image extraction using WEIDJ approach are speedy than both approaches, DOM and JSON.

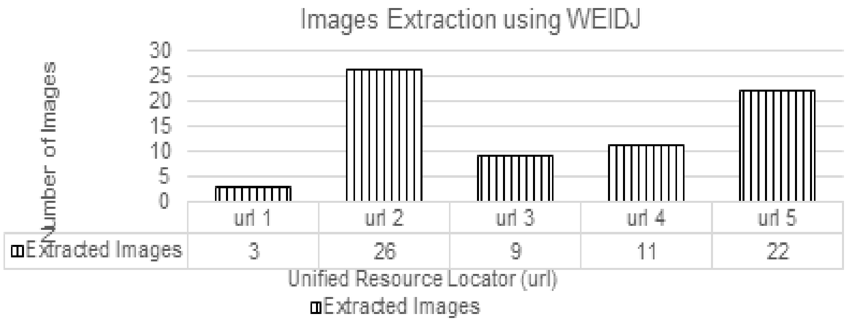


Figure 12. Image extraction using WEIDJ approach

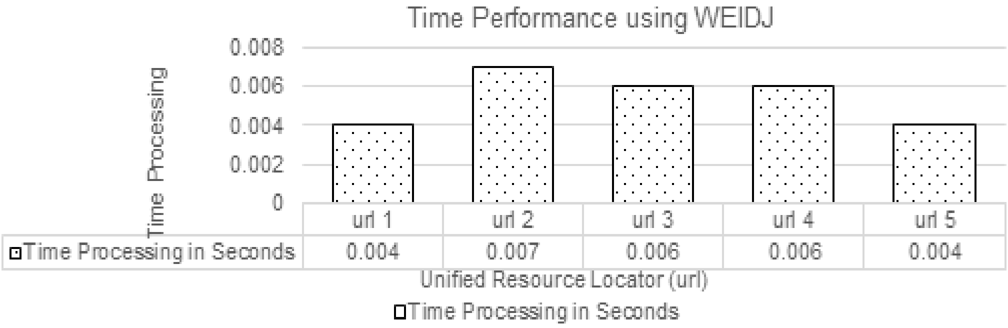


Figure 13. Time performance using WEIDJ approach

Figure 14 shows comparison of time performance between DOM, JSON and WEIDJ approach. The graph consist of web address and time required for extraction process in seconds. The web address are represented by unified resource locator (*url*) that can be referred from Table 4. The graph shows time for image extraction is speedy rather than JSON and DOM.

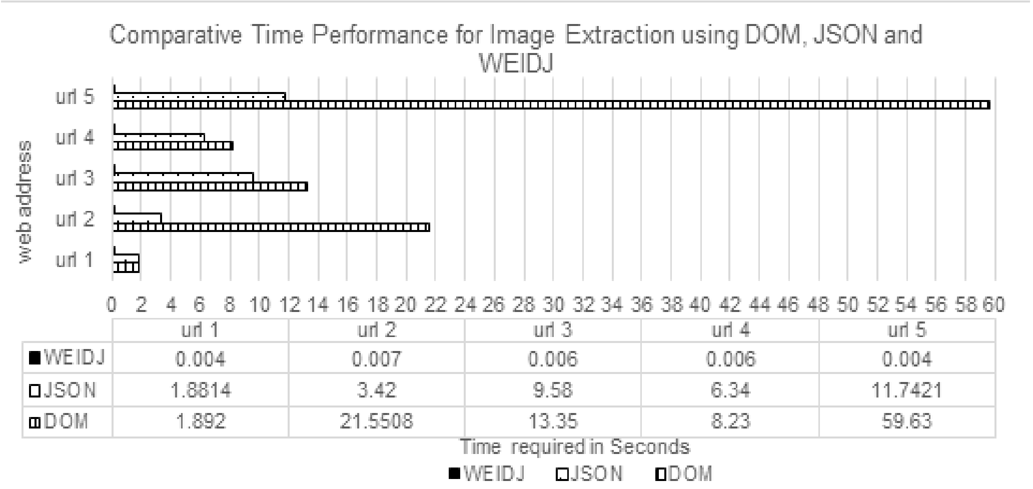


Figure 14. Time performance for image extraction using DOM, JSON and WEIDJ

5. CONCLUSION

In this work, we presented a new approach to study how JSON and DOM can be composed together into web extraction applications. From the experiment findings, the implementation of extraction using DOM and JSON proves that the extraction of images can be done in efficient way. This indicates the efficiency of extraction process. Complementary to this, we intend to combine both approaches to get the best performance. This wrapper has been developed based on proposed model, WEIDJ. In this paper, we do experimental by extracting images from single web page using three approaches DOM, JSON and WEIDJ. Finally, selected images can be saved in single multimedia database. For further purpose, user can query all images that has been saved in multimedia database. WEIDJ is proposed to make the extraction process more accurate and efficient as possible. In addition, this model motivates leads to the increase performance of time and space complexities.

REFERENCES

[1] M. Man, W. A. W. A. Bakar, N. H. Ali, and M. A. Jalil, "Hybrid federated data warehouse integration model: Implementation in mud crabs case study," *The Journal of Science and Technology*, vol. 36, pp. 28-38, 2015.

[2] S. Palaniapan and N. Y. Huey, "A tool for healthcare information integration," *Journal of ICT*, vol. 5, pp. 29-44, 2006.

[3] P. Scheuermann, "Report on the workshop on heterogenous database systems held at Northwestern University, Evanston, IL," *SIGMOD Record,* vol. 19, no. 4, pp. 23-31, December 11-13, 1989 1990.

[4] J. Ronk. (2014, 29 July 2015). *Structured, semi-structured and unstructured data*. Available: https://jeremyronk.wordpress.com/2014/09/01/structured-semi-structured-and-unstructured-data/

[5] W. Kim, S.-S. Park, and H. H. Kim, "A framework for the integration of multimedia data," *Journal of Object Technology,* vol. 4, pp. 27-35, 2005.

[6] P. Rawat, S. Sayyad, S. Surinder, and S. Shelke, "Application for web data extraction and analysis," *Imperial Journal of Interdisciplinary Research,* vol. 2, 2016.

[7] M. Man, I. A. A. Sabri, M. M. A. Jalil, N. H. Ali, and S. Muhamad, "Information integration architecture system for empowering rural woman in setiu wetlands," presented at the Seminar Ekosistem Setiu 2016: Sains Marin & Sumber Akuatik Untuk Kelangsungan Hidup, Universiti Malaysia Terengganu, 2016.

[8] I. A. A. Sabri and M. man, "Multiple types of semi-structured data extraction using WEID," presented at the Regional Conference on Sciences, Technology and Social Sciences (RCSTSS), Copthorne Hotel Cameron Highlands, 2016.

[9] S. M. Narawade, N. M. Prabhakar, N. S. Maruti, S. M. Bhagwat, and B. Burghate, "A web based data extraction using hierarchical (DOM) tree approach," *International Journal for Innovative Research in Science and Technology,* vol. 2, pp. 255-257, 2016.

[10] M. K. Sangeetha, "Component based information retrieval using DOM," *International Journal of Software Engineering and Its Applications,* vol. 10, pp. 117-126, 2016.

[11] B. Mehta and M. Narvekar, "DOM tree based approach for web content extraction," in *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*, 2015, pp. 1-6.

[12] M. K. Yusof and M. Man, "Efficiency of JSON approach for data extraction and query retrieval," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 4, pp. 203-214, 2016.

[13] K. Kanaoka, Y. Fujii, and M. Toyama, "Ducky: a data extraction system for various structured web documents," in *Proceedings of the 18th International Database Engineering & Applications Symposium*, 2014, pp. 342-347.

[14] K. Kanaoka and M. Toyama, "Effective web data extraction with ducky," in *Proceedings of the 19th International Database Engineering & Applications Symposium*, 2015, pp. 212-213.

[15] K. Kanaoka and M. Toyama, "Browser GUI for generating web data extraction rules in Ducky," in *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, 2015, p. 79.

[16] G. Wang, "Improving data transmission in web applications via the translation between XML and JSON," in *Communications and Mobile Computing (CMC), 2011 Third International Conference on*, 2011, pp. 182-185.

[17] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowledge-Based Systems,* vol. 70, pp. 301-323, 2014.

[18] A. H. Laender, B. Ribeiro-Neto, and A. S. da Silva, "DEByE–data extraction by example," *Data & Knowledge Engineering,* vol. 40, pp. 121-154, 2002.

[19] J. Alarte, D. Insa, J. Silva, and S. Tamarit, "Analysis of hyperlinks and DOM comparison for site-level web template extraction⋆," 2015.

[20] S. Z. Abidin, N. M. Idris, and A. H. Husain, "Extraction and classification of unstructured data in WebPages for structured multimedia database via XML," in *Information Retrieval & Knowledge Management,(CAMP), 2010 International Conference on*, 2010, pp. 44-49.

[21] S. Sarawagi, "Information extraction," *Foundations and trends in databases,* vol. 1, pp. 261-377, 2008.

[22] S. Flesca, G. Manco, E. Masciari, E. Rende, and A. Tagarelli, "Web wrapper induction: a brief survey," *AI Communications,* vol. 17, pp. 57-61, 2004.

[23] M. Raza and S. Gulwani, "Automated Data Extraction using Predictive Program Synthesis," 2017.

[24] J. G. Waghmare and V. B. Maral, "Implementation of novel web based data extraction using template extraction technique and non information filtering," in *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, 2016, pp. 2023-2026.

[25] D. Song, F. Sun, and L. Liao, "A hybrid approach for content extraction with text density and visual importance of DOM nodes," *Knowledge and Information Systems,* vol. 42, pp. 75-96, 2015.

[26] A. Bhardwaj and V. Mangat, "An improvised algorithm for relevant content extraction from web pages," *Journal of Emerging Technologies in Web Intelligence,* vol. 6, pp. 226-230, 2014.

[27] V. B. Kadam and G. K. Pakle, "DEUDS: Data Extraction Using DOM Tree and Selectors," *International Journal of Computer Science and Information Technologies,* vol. 5, pp. 1403-1410, 2014.

[28] S. López, J. Silva, and D. Insa, "Using the DOM tree for Content Extraction," *arXiv preprint arXiv:1210.6113,* 2012.

[29] J. Wang and F. H. Lochovsky, "Data extraction and label assignment for web databases," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 187-196.

[30] Y. Zhai and B. Liu, "Web data extraction based on partial tree alignment," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 76-85.

[31] M. K. Yusof and M. Man, "Efficiency of JSON for data retrieval in big data," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 7, pp. 250-262, 2017.

[32] J. Creech. (2012, 31 May 2017). *Biodiversity web resources*. Available: http://www.istl.org/12-fall/internet.html