ABSTRACT
Web Data Extraction has become a very serious problem especially having vision based features. We have studied different approaches in a lane range of application domains. Many approaches to extracting vision based data from the Web have been designed to solve specific problems and operate in web application domains. Other techniques reuses in the meadow of Information Extraction. This paper aims in providing structured and comprehensive vision-based features of research efforts made in the field of Web Data Extraction. The overview of our work is to present a cataloguing of vision based approaches in terms of the applications for which they have been working. This differentiates our work from other researcher’s efforts to organize vision-based approaches on the basis of the algorithms, techniques and tools they use. We divide vision based Web Data Extraction approaches into different kinds along with an application how the data regions are extracted from a deep web page.

Keywords: vision-based features, web data extraction, Enterprise applications, Social web applications.

1. INTRODUCTION
Vision-based Web Data Extraction system can be done with various web sources using different techniques and extract the data regions stored in the deep web page [6]. Consider, if the source is a HTML Web page, the extracted information could consist of elements in the page as well as the full-text of the page itself. The deep web data region has to be again convert into a Structured format.[Zhao 2007; Irmak and Suel 2006].Vision-based web data extraction has useful data extraction from the deep web pages which are hidden web pages. The consequence of Vision based Web Data Extraction systems depends large (and quickly growing) amount of information is continuously produced, shared and consumed online: Web Data Extraction systems allow to efficiently collecting this information with a limited human effort [18].

Now a day’s, World Wide Web has become one of the most major information possessions. Most of the information is in the form of unstructured text, a huge amount of semi-structured objects, called data records, are enclosed on the Web [5]. The quantity of Web information has been rising rapidly, mainly with the evolution of Web 2.0 environments, where the users are encouraged to provide rich content. A large amount of Web information is presented in the form of a Web record, which exists in both detail and list pages. Extraction of Web information is a significant process for information integration, but many web pages may provide the same or analogous information using entirely diverse formats or syntaxes, which makes the integration of information a challenging task. Due to the heterogeneity and lack of structure of Web information, automated discovery of relevant information becomes a difficult task [1]. The Deep Web is the content on the web not accessible by a search on general search engines, which is also called as hidden Web or invisible Web. Deep Web contents are accessed by queries submitted to Web databases and the retrieved information i.e., query results is enclosed in Web pages in the form of data records. These special Web pages are generated dynamically and are difficult to index by conventional crawler based search engines, namely Google and Yahoo.

In this paper, we describe this kind of special Web pages as deep Web pages [12]. In general, Web information extraction tools are divided into three categories: (i) Web directories, (ii) Meta search engines, and (iii) Search engines.

Normally, a Web page contains several blocks or areas, such as main content areas, navigation areas, advertisements, etc. A block is a semantic part of a web page that has its own text content, style and functionality. Generally, a web page comprises two blocks: main content blocks and noise blocks. Only the main content blocks describe the informative portion that most users are interested in. Even though other blocks are supportive in improving functionality and guiding browsing, they negatively affect such web mining tasks as web page clustering and classification by reducing the precision of mined results and speed of processing. Hence, these blocks are called noise blocks in this context. For instance, a CNN web page contains a sports news report in the center of the page, which is the main content of this page. Also, there are advertisements, navigation bars, and others, situated around the main content, which are called as noise blocks [2].

Semantically related content is usually grouped together and the whole page is divided into regions for diverse contents by means of explicit or implicit visual separators namely lines, blank areas, images, font sizes, colors, and more [4]. In several Web pages, the main content information is present in the center block and the rest of the page has ads, navigation links, and privacy statements as noisy data. Web pages are often cluttered with
disturbing features around the body of an article that divert the attention of the user from the actual content they are interested in. These “features” may comprise pop-up advertisement, showy banner advertisements, search and filtering panel, superfluous images, or links scattered around the screen. However, these noisy data are present in various patterns in diverse Web sites. Such irrelevant items should be removed for extracting only the significant information [3].

2. VARIOUS VISION–BASED FEATURE EXTRACTION APPROACHES

The first part of this paper is dedicated to the discussion of the techniques adopted in the field of the Web Data Extraction. The first trials to extract data from the Web are dated back in early nineties, as reported by [Kaiser and Miksch 2005; Sarawagi 2008]. In the early stage, this discipline borrowed approaches and techniques from Information Extraction (IE) literature. In particular, two classes of strategies emerge [Kaiser and Miksch 2005]: learning techniques and knowledge engineering techniques also called as learning-based and rule-based approaches, respectively [Sarawagi 2008]. These classes share a common rationale: the former was thought to develop systems that require human expertise to define rules (for example, regular expressions or program snippets) to successfully accomplish the data extraction. These approaches require specific domain expertise: users that design and implement the rules and train the system must have programming experience and a good knowledge of the domain in which the data extraction system will operate and, possibly, the ability of envisaging potential usage scenarios and tasks assigned to the system. On the other hand, also some approaches of the latter class involve strong familiarity with both the requirements and the functions of the platform, so the human engagement is essential. To reduce the commitment of human domain experts, a number of strategies have been devised. Some of them have been developed in the context of Artificial Intelligence literature, by means of the adoption of specific algorithms using the structure of Web pages to identify and extract data. Some others are borrowed from the Machine Learning discipline, thought as supervised or semi-supervised learning techniques which allow the design of systems capable of being trained by examples and then to autonomously extract data from similar (or even different) domains. In the following we will discuss separately scrutinize.

2.1 Partial Tree Alignment

We now present the partial tree alignment technique for data extraction. The primary role is how to match corresponding data items or fields from all data records, There are two process involved in PAT[19] where one is to produce one rooted tag tree for each data record: After all data records are identified, the sub-trees of each data record are rearranged into a single tree. Each data record may be contained in more than one sub-tree of the original tag tree of the page, and another is each data record may not be contiguous. Thus, this sub-process is needed to compose a single tree for each data record (an artificial root node may also need to be added). We will not discuss this further as it is fairly simple. The tag trees of all data records in each data region are aligned using our partial alignment method which is based on tree matching. It should be noted that in the matching process, we only use tags. No data item is involved.

2.2 Vision-based Page Segmentation Algorithm

Basically, the vision-based content structure of a page is obtained by combining the DOM structure and the visual cues. The segmentation process is illustrated in Figure 2. Here mainly three steps are involved: block extraction, separator detection and content structure construction. These three steps as a whole are regarded as a round. The algorithm is top-down[21]. The web page is firstly fragmented into several big blocks and the hierarchical structure of this level is recorded. For each big block, the same segmentation process is carried out recursively until we get sufficiently small blocks whose DoC values are greater than pre-defined PDoC.

![Figure 2: Implementation of VIPs in Three Stages](image)

2.2.1 Extraction of Visual Block

In the extraction of visual blocks, VIPs is in search of relevant visual blocks occurring in each of the sub-division web of page. While constructing the DOM tree each of the node is a visual block. In many of the situations, the current node should be further divided and replaced by its children. Using HTML specifications, DOM trees don’t exhibit correct relationship of different DOM node. For each extracted node that represents a visual block, its DoC value is set according to its intra visual difference. This process is iterated until all appropriate nodes are found to represent the visual blocks in the sub-division web of page.

2.2.2 Detection of Visual Separator

Once all the visual blocks are retrieved, all are placed in a buffer for the detection of Visual separator. After all blocks are extracted, they are put into a pool for visual separator detection. Separators are horizontal or vertical
lines in a web page that visually cross with no blocks in the buffer. From a visual perspective, separators are good indicators for discriminating different semantics within the page. A visual separator is represented by a 2-tuple: (Pstart, Pend), where Pstart is the start pixel and Pend is the end pixel. The width of the separator is calculated by the difference between these two values.

### 2.2.3 Content Structure Construction

When separators are detected and separators’ weights are set, the content structure can be constructed accordingly. The construction process starts from the separators with the lowest weight and the blocks beside these separators are merged to form new blocks. This merging process iterates till separators with maximum weights are met.

![Visual blocks identified using VIPS algorithm](image)

**Figure 1**: Visual blocks identified using VIPS algorithm

### 2.3 ViDE: Data record extraction

Data record extraction aims to discover the boundary of data records and extract them from the deep Web pages. An ideal record extractor should achieve the following: 1) all data records in the data region are extracted and 2) for each extracted data record, no data item is missed and no incorrect data item is included. We first identify the data region, and then, extract data records from the data region. PF1 and PF2 indicate that the data records are the primary content on the deep Web pages and the data region is centrally located on these pages [7]. The data region corresponds to a block in the Visual Block tree. We locate the data region by finding the block that satisfies the two position features. Though very simple, this method can find the data region in the Visual Block tree accurately and efficiently. Each data record corresponds to one or more subtrees in the Visual Block tree, which are just the child blocks of the data region. So, we only need to focus on the child blocks of the data region. In order to extract data records from the data region accurately, two facts must be considered. First, there may be blocks that do not belong to any data record, such as the statistical information (e.g., about 2,038 matching results for java) and annotation about data records (e.g., 1, 2, 3, 4, 5 (Next)). These blocks are called noise blocks in this paper. Noise blocks may appear in the data region because they are often close to the data records. According to LF2, noise blocks cannot appear between data records. They always appear at the top or the bottom of the data region. Second, one data record may correspond to one or more blocks in the Visual Block tree, and the total number of blocks in which one data record contains is not fixed. That is, we attempt to determine which blocks belong to the same data record. We achieve this in the following three phases:

1. Phase 1: Filter out some noise blocks.
2. Phase 2: Cluster the remaining blocks by computing their appearance similarity.

### 2.4 Vision based deep web page Extraction

This paper endeavour to identify and understand different information areas, or chunks, of a web page – where a chunk represents an area on a page. For example, a news article might be one block, and an advertisement might be another. After the chunks of a page are identified, each of it look at the characteristics or features of each chunk to determine how important each chunk. This paper use an VDEC[20] that builds a output based upon features of chunks, and upon URL’s given as input determining the importance of a number of chunks in a collection of URL’s. The VDEC help the search engine to determine which chunks are the most important, so that it can use to determine the important data regions in each of the deep web page by processing its URL.

The kinds of features that might be used in the VDEC include features such as the size of chunks or their locations or both, as well as “content” features such as the number of links within a block or the number of words within the chunk. A chunk located in the centre of a page might be considered more important that another chunk at the bottom of a page. The “content” features that are included as follows[9]:

- The number and size of images in the chunk.
- The number of links and the number of words in each link, in the chunk.
- The number of words in the text of the chunk.
- User interaction of the chunk, looking at things like the number and size of input fields.
- Forms within the chunk, again looking at number and size number and size of input fields.

#### 2.4.1 Chunk Segmentation

We see a chunk within a web page that signifies the principal topic of that page may aid a search engine choose which words are the most important ones on the page when it aims to associate the page with keywords that someone might search with to find that page. Determining the most important chunk on a page could influence
the weight and importance of links from different chunks on a page, so that a link from the most important chunk on a page has more value than a link from the least important chunk. Removing these noises will help in improving the mining of web. To assign importance to a region on a web page \( W_p \), we first need to segment a web page into a set of chunks. Hence, to clean a web page, a pre-processing step called Chunk Splitting Operation (fig.2) is performed [10].

Basically, the layout of many web pages follows a similar pattern in such a way that the main content is enclosed in one big <div> or <td> element which is HTML tags. In our paper, we are concentrating only the content inside the “div” tag. The <div> tag defines a division or a section in an HTML document and it is often used to group chunk-elements. In our approach <div> tag is consider as chunk. Normally, a <div>tag separated by many sub <div> tags based on the content of the deep web page. If there is no <div>tag in the sub <div>tag, the last <div>tag is consider as leaf node. Include a div element where flow content is expected.

Begin the div element with a starting <div> tag. The element name uses lower case letters and should be in the HTML namespace, which it will pick up automatically from the xmlns attribute on the <html> tag. Inside the div element, between the starting <div> tag and the ending </div> tag, code the inner HTML flow content.

End the div element with a matching </div> closing tag.

The Chunk Splitting Process aims at cleaning the local noises by considering only the main content of a web page enclosed in div tag. The main contents are segmented into various chunks[13].

In fig.1 we have taken an example of a tree sample which consists of main chunks and sub-chunks. The main chunks are segmented into chunks \( C_1 \), \( C_2 \) and \( C_3 \) using Chunk Splitting Operation and sub-chunks are segmented into \( C_{2,1}, C_{2,2}...C_{5,1} \) in fig 2.

2.4.2 Noisy Chunk Removal

A page from a news site on the Web might contain an article about an international political event and “noise information” like a diet advertisement, a legal notices section, and a navigation bar. A search engine attempting to index the full content of the page might choose keywords based upon the noise information instead of from text related to the primary topic of the page – the political event[11]. To crawl the web, a search engine service may use a list of root web pages to identify all web pages that are accessible through those root web pages. The keywords of any particular web page can be identified using various well-known information retrieval techniques, such as identifying the words of a headline, the words supplied in the metadata of the web page, the words that are highlighted, and so on. As a result, a search engine service may select keywords based on noise information, rather than the primary topic of the web page. A chunk of a web page represents an area of the web page that appears to relate to a similar topic. The importance system provides the characteristics or features of chunk to an importance function that generates an indication of the importance of that chunk to its web page. This paper asks users to provide an indication of the importance of blocks of web pages in a collection of web pages.

2.4.3 Chunk Weightage for Sub-Chunk: In the previous step, we obtained a set of chunks after removing the noise chunks and duplicate chunks present in a deep web page. Web page designers tend to organize their content in a reasonable way: giving prominence to important things and deemphasizing the unimportant parts with proper features such as position, size, color, word, image, link, etc. In our research work, we have concentrated on the three parameters Title word relevancy, keyword frequency based chunk selection, and position features which are very significant. Each parameter has its own significance for calculating sub-chunk weightage[21].

For each noiseless chunk, we have to calculate, Title Keyword, Keyword Frequency based chunk selection and Position features (PFs). In our experiments, the threshold of the ratio is set at 0.7, that is, if the ratio of the horizontally centered region is greater than or equal to 0.7, then the region is recognized as the data region.

2.4.4 Chunk Weightage For Main Chunk: We have obtained sub-chunk weightage of all noiseless chunks from the above process. Then, the main chunks weightage are selected .Thus, finally we obtain a set of...
important chunks and we extract the keywords from the above obtained important chunks for effective web document clustering mining. Many web pages contain more than one topical section, or blocks, which may make it difficult or a search engine to tell what a page is about when it is trying to index that page[14]. These blocks may include such things as a main content area, navigation bars, headings, footers, advertisements, and other content that may refer to other pages on a site, or on other sites. Here we have considered a web page in figure 3.

Figure 3: A sample Deep Web Page for data region extraction. Figure 4 : The sample data region extracted

The data region which we have extracted from the url defined http://news.engineering.iastate.edu/category/department-news/electrical-and-computer-engineering/ is shown in figure 4 using Vision based Deep Web Data Extraction For Web Document Clustering And Application

3. APPLICATIONS
In the Web Data Extraction area, majority of the work is taking place pertaining to solve some special issues related to a single or, sometimes, a set of fields of application. The aim of the this paper is to analyze a large number of applications that are strictly interconnected with Web Data Extraction tasks[17]. As per our experience, here we are discussing classification of applications based on Web Data Extraction techniques even if they have been originally designed to operate in specific domain and, in some cases, they can appear as unrelated [19]. The bunch of applications possibly benefiting from Web Data Extraction techniques is quite large and it encompasses applications designed to work in the business domain to applications developed in the context of the Social Web.

We categorize applications of Web Data Extraction techniques in two:

3.1 Enterprise Applications: Many of the applications come under this category for increasing a commercial purpose for increasing both the level of automating business process and its efficiency.

3.2 Social Web Applications: Some of the applications comes under this category mainly to designed to extract and collect data from a social web platform.(e.g., a Educational Website like http://news.engineering.iastate.edu/category/department-news/electrical-and-computer-engineering/etc ).

The categorization of the applications specified above is not projected to be tough likewise there are some applications which has been intended to design in the enterprise area of applications added with some shared features with applications working on social web platforms. For example many of the areas include providing services for customers to extract information. This information is valuable because it enables firm managers to design strategies to increase the quality of services/products delivered to customers.

4. PERFORMANCES OF VARIOUS VISION BASED EXTRACTION APPROACHES
We are going to compare the performances of various vision based extractions techniques. In vision based page segmentation, it has been observed that 600 web pages are frequent and used listed in 14 main categories. After executing the Vision based page segmentation algorithm it has found that 93% pages have their The VIPS algorithm is run on all the pages and the results are assessed by several individuals. As can be seen, 93% pages have their logical comfortable structures correctly detected. For such pages, the levels of hierarchies are not constructed properly due to insufficient visual cues to separate blocks. And then data record extraction and data item extraction are done based on visual features. Later visual wrappers create the wrappers to enhance the efficiency of both data record extraction and data item extraction. Thus finally generate accurate results to confirm the affluent visual features on deep Web pages to construct effectual data extraction algorithms. And Next is , ViDE can only process deep Web pages containing one data region, while there is significant number of multidata-region deep Web pages. The efficiency of ViDE can be enhanced. In the current ViDE, the visual information of Web pages is obtained by calling the programming APIs of IE, which is a time-consuming process. Next We concentrate VDEC where still the efficiency of extracting the visual features has been developed by increasing the threshold constant while calculating the important chunks and we extract the keywords from the above obtained important chunks for effective web document clustering mining.
5. CONCLUSION
In this paper we are discussing basics of web data extraction, going analyze various techniques. We have analyzed the PAT to extract the data with the matching process, only use tags. No data item is involved where in the VIPS. Implementation three Stages are involved to extract web data. ViDE we attempt to determine which blocks belong to the same data record by filter out some noise blocks. Cluster the remaining blocks by computing their appearance similarity, Discover data record boundary by regrouping blocks. VDEC where still the efficiency of extracting the visual features has been developed by increasing the threshold constant. We also discuss various applications involved using web data extraction.

REFERENCES
[18] Wei Liu, Xiaofeng Meng, Weiyi Meng;” Vision based Web Data Records Extraction”.