IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 3, 2020, accepted November 14, 2020, date of publication November 18, 2020, date of current version December 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039044

Automatically Discovering Relevant Images From Web Pages

ERDINÇ UZUN [1], ERKAN ÖZHAN [1], HAYRI VOLKAN AGUN [2], TARIK YERLIKAYA [3], AND HALIL NUSRET BULUŞ [1]

[1]Department of Computer Engineering, Çorlu Faculty of Engineering, Tekirdağ Namık Kemal University, 59860 Tekirdağ, Turkey
[2]Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Bursa Technical University, 16000 Bursa, Turkey
[3]Department of Computer Engineering, Faculty of Engineering, Trakya University, 22030 Edirne, Turkey

Corresponding author: Erkan Özhan (erkanozhan@gmail.com)

ABSTRACT Web pages contain irrelevant images along with relevant images. The classification of these images is an error-prone process due to the number of design variations of web pages. Using multiple web pages provides additional features that improve the performance of relevant image extraction. Traditional studies use the features extracted from a single web page. However, in this study, we enhance the performance of relevant image extraction by employing the features extracted from different web pages consisting of standard news, galleries, video pages, and link pages. The dataset obtained from these web pages contains 100 different web pages for each 200 online news websites from 58 different countries. For discovering relevant images, the most straightforward approach extracts the largest image on the web page. This approach achieves a 0.451 F-Measure score as a baseline. Then, we apply several machine learning methods using features in this dataset to find the most suitable machine learning method. The best f-Measure score is 0.822 using the AdaBoost classifier. Some of these features have been utilized in previous web data extraction studies. To the best of our knowledge, 15 new features are proposed for the first time in this study for discovering the relevant images. We compare the performance of the AdaBoost classifier on different feature sets. The proposed features improve the f-Measure by 35 percent. Besides, using only the cache feature, which is the most prominent feature, corresponds to 7 percent of this improvement.

INDEX TERMS Image classification, image retrieval, feature extraction, web crawlers, web mining.

I. INTRODUCTION

The internet provides us with a lot of valuable content among HTML tags. Nowadays, extracting this content automatically is one of the challenging tasks in information retrieval. In this task, the researchers mostly concentrate on obtaining textual contents including title [1], [2], main content [3], [5], summary [5], and reviews [6], [7]. Also, there are few studies [8], [9] that focus on automatically extracting relevant images. Moreover, the performance results of these studies are low. This study aims to improve these results through the proposed novel features.

Previous studies [8], [9] consist of around 1100 web pages which are standard news. However, a website consists of many different pages including standard news, gallery pages, link pages. A web page of standard news may have one or more relevant images but no images at all. A gallery web page contains too many related images. A link web page has no relevant images. In this study, all different pages are included in the data set. Moreover, while previous studies focus on one relevant image, namely a representative image, in the content, this study concentrates on all relevant images in the content. For our purposes, we developed a web crawler[1] to download web pages and their images for creating a new dataset.[2] With this crawler, 20,000 web pages and 635,015 images were downloaded from 58 different countries and 200 different websites. Unlike other studies, a very large dataset is created that generalizes the solution.

Features are needed for an automatic prediction process. For this task, the first features that come to mind are height,

The associate editor coordinating the review of this manuscript and approving it for publication was Maurizio Tucci.

[1]This crawler downloads the images and extracts features from these images for use in machine learning. It is available via the web page https://github.com/erdincuzun/ImageCrawler

[2]This dataset with 30 features is available via the web page https://adys.nku.edu.tr/Datasets/

width, file extension, and file size of an image. Unfortunately, these features are not enough for accurate prediction. In order to resolve this situation, we identified features that can be obtained from the web page by considering other features [5], [10] suggested in the literature. Furthermore, new features are suggested thanks to the additional modules added to our crawler in this study. As a result, 30 features were constructed for an image. To the best of our knowledge, 15 of these features have been mentioned for the first time in the literature. The contribution of these features to the prediction process is examined in the experimental section of this study.

Some of the approaches manually assign weights to simple features [8], [11]. Heuristics based on these weights are applied on datasets with limited samples. Increasing the number of samples is beneficial for the prediction accuracy of machine learning models. For determining the most representative image on a web page, Vyas and Frasincar [9] construct their prediction model on the machine learning method, Support Vector Machines (SVM). Nonetheless, these studies are not contributive in terms of the comparison of various classification methods. In this study, we report the best evaluation score along with the comparison of several different machine learning methods in discovering the relevant images.

Nowadays, Web pages have very complex and visually rich layouts. These layouts contain a lot of images and contents. However, the number of relevant images is very few. For this reason, the image datasets constructed for relevant image extraction is imbalanced. This is known as an imbalanced dataset problem that negatively affects the prediction model of machine learning methods. The problems caused by this imbalanced dataset to the prediction model can be overcome with ensemble methods, which are a special form of a machine learning method. Ensemble methods combine prediction models obtained from various machine learning algorithms in order to produce an optimal prediction model. The experiment section examines the contribution of these methods in our imbalanced dataset.

In a prediction model derived from a machine learning method, not all features have the same effect for the model. Some features may not contribute to the model at all. That is, having irrelevant features in your dataset can decrease the performance of the models constructed from machine learning methods. Therefore, finding the required features, namely feature selection, from a dataset has become a critical issue for enhancing this task. Feature selection is a practical solution to find out a minimal feature subset [12], [13]. You can apply this solution to select those features which contribute most to your performance of the model. In this study, subsets of features are constructed by taking information gain values into account, and then models obtained from these subsets are evaluated.

The rest of the study is organized as follows. The second section is about related studies. The third section gives information about the layouts and design on a web page. The fourth section introduces our web crawler and features extracted from this crawler. The fifth section covers the dataset constructed with our crawler, machine learning methods, performance metrics, and the experimental evaluations of these methods. The last section reports our conclusion and future studies.

II. RELATED STUDIES

Image retrieval addresses the issue of browsing, searching, and retrieving images from a large collection. Most of the studies are the issue of searching on web data which is the world's largest collection. Kherfi *et al.* [14] compare some existing image retrieval algorithms, which are based on textual user queries. They have emphasized that collecting relevant images in web content to obtain information from websites with a few texts has become an important issue. Nie *et al.* [15] propose an approach to automatically estimate the image search performance. Kennedy and Naaman [16] introduce an approach to extract representative images for landmark each from the Web. Wang *et al.* [17] use images of a web page in order to classify it as objectionable or benign. Hor and Fekri-Ershad [18] present an approach to retrieve images from a large database. In this approach, they use local texture information. Bani and Fekri-Ershad [19] utilize texture and colour information obtained from spatial and frequency domains. In our study, we focus on the retrieval of images from the web pages by using the information of the website. Besides, the entire website is utilized while discovering the relevant images on the web page.

Some studies focus on a single image, namely a representative image, that represents the content on a web page to users. This image is chosen among multiple images. Helfman and Hollan [20] rate images based on features including large, square, and colorful. Gali *et al.* [8] give scores for images using image size, aspect ratio, alt tag, title tag, image path, image format. They use a heuristic score for every feature. Hu and Bagga [21] introduce an algorithm for automatically classifying images into image categories including story and preview images. Bhardwaj and Mangat [22] propose a technique to extract image tags which size greater than 120,000 pixels (for example, the size of tags 300×400 or 400×300). Sabri and Man [23] introduce an algorithm that uses the Document Object Model (DOM) and JavaScript Object Notation (JSON) for extracting images from a web page. These studies use a single web page. Our study utilizes websites that contain a lot of web pages thanks to our Web crawler.

Web crawlers download a lot of data that can be transformed into information for different purposes. Noh *et al.* [24] utilize term frequency/document frequency, entropy, and compile rules for determining the relevance of a webpage for a topic. Pant and Srinivasan [25] uses texts around a hyperlink within a Web page for predicting corresponding hyperlinks. Batsakis *et al.* [26] use a web page content and link information for determining download priorities in order to improve the performance of their crawler. Uzun *et al.* [5] develop an intelligent crawler, namely iCrawler, automatically pulls content out of various layouts for improving the crawling process. Uzun [10] proposes a novel approach that extracts data quickly using the string functions and additional information including the starting position, the number of the inner tag, and tag repetition obtained from web pages. The data obtained through web crawlers can be used for many different purposes. There is a considerable amount of literature [27] on this issue. In this study, we utilize information derived from the crawling process.

Some studies employ machine learning methods in image problems. Hu and Bagga [21], Tong and Chang [28], Zang *et al.* [29] use SVM in order to find informative images based on textual queries from users. Vyas and Frasincar [9] focus on determining the most representative image on a web page. They also use SVM and new features to improve their f-Measure score. In this study, many machine learning methods are tested rather than focusing on just one machine learning method. Besides, the number of features is reduced with feature selection methods that are not recommended in other studies.

Companies such as Google, Bing, Yandex, and Yahoo which want to easily obtain the data on web pages, have a joint effort to suggest attributes for HTML tags. The best-known organization on this issue is schema.org. Schema.org creates a structured data markup schema supported by major search engines. Such organizations are important for constructing the Semantic Web. On the other hand, when the website concept changes, the proposed structure alters. For example, Facebook recommends the "Facebook Open Graph" structure. However, this structure is not enough for search engines. The standards on this issue are defined by W3C with the RDF (Resource Description Framework[3]) that is a framework for representing information on the Web. Although this issue has been studied for many years, Web designers must follow the rules. However, these designers and developers interpret the suggestions on their own or do not use them at all. In this study, we aim to present a general solution without using such suggestions. Our study can be used in a system that proposes automatic tag attributes for web designers.

III. MOTIVATION

There are a lot of different website layouts[4] that define a website's structure. Relevant and irrelevant images are included in these layouts. Relevant images are images related to the content of the page, while irrelevant images are images including advertisements, other links, headers, logos, etc. Images can be of many different sizes and in different parts of a web page. Fig. 1. contains some examples of these designs and images in these layouts.

[3]RDF is a W3C standard for data interchange on the Web. For more information, see https://www.w3.org/RDF/

[4]Web layouts consist of patterns that rule the structure of the document. For more examples about layouts, see https://www.w3schools.com/css/css_website_layout.asp and https://www.w3schools.com/css/css_templates.asp.

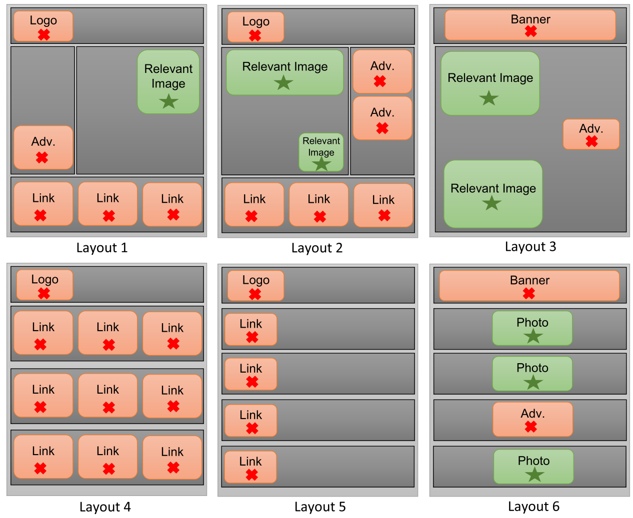

FIGURE 1. Some example layouts for relevant and irrelevant images.

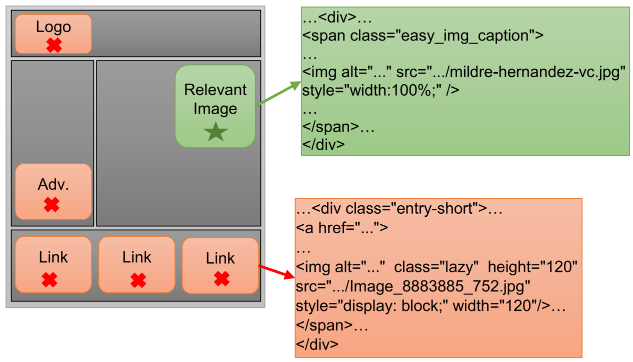

FIGURE 2. Relevant and irrelevant images in HTML tags.

Web designers can make very different layout designs using various tags. In this design, the content and images are located between HTML tags. This structure is called an HTML element that is defined by a start tag, some content, and an end tag. This content includes other elements or text viewed in the browser. In this case, we are dealing with an unstructured text as indicated in Fig. 2.

Web designers usually use the img element for embedding an image on a web page. The img element starts with the img tag and has many attributes that are a modifier of an HTML element type. That is, these attributes are appended to the img tag. Fig. 2 shows HTML tags of relevant and irrelevant images on a web page. In the img tag, the scr attribute is a required attribute for determining the URL of the image. All HTML attributes generally appear as name-value pairs. For example, the scr attribute inside the img element consists of name/value pairs like $src = \text{``}\ldots/mildre - hernandez - vc.jpg''$. All other attributes within the img element are optional to alter the default functionality of an element type. Deciding on an img element through attributes is a difficult process. For example, in Fig. 2, it cannot be decided over scr or alt attributes. Properties such as width, height, class can be distinguished

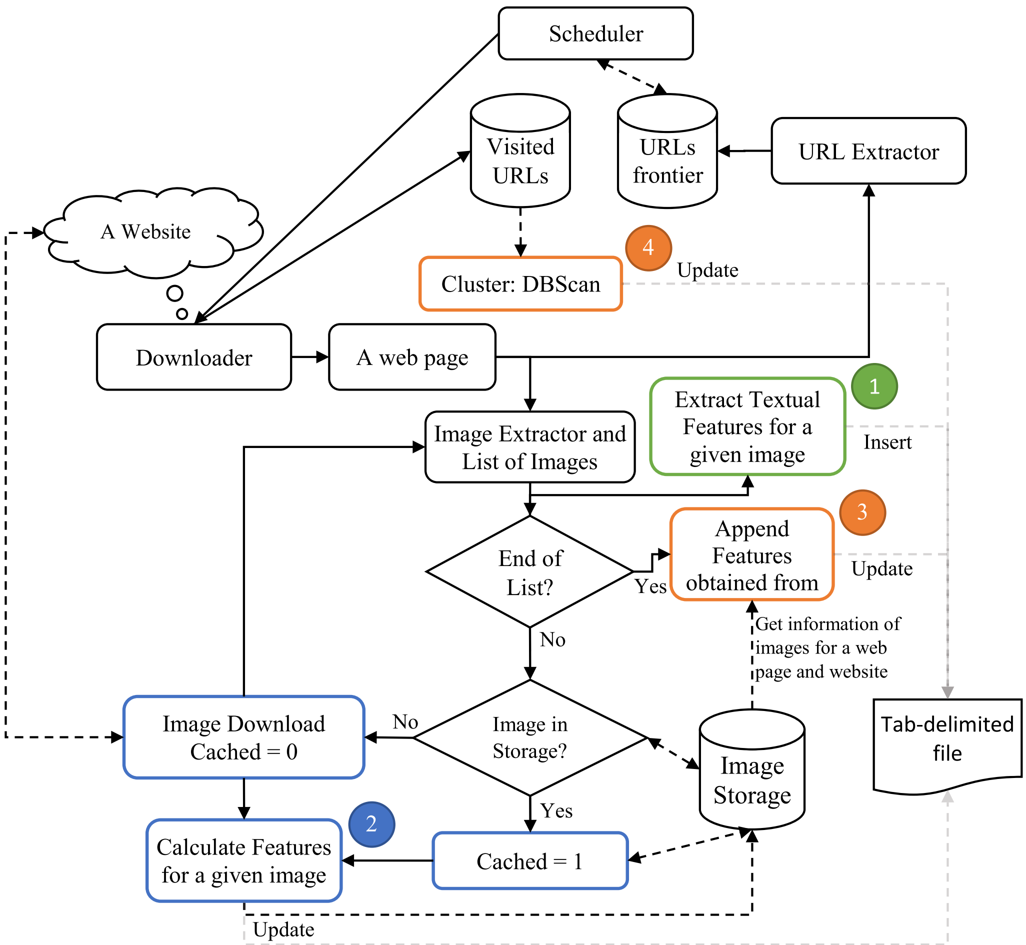

FIGURE 3. Flow chart of our crawler.

for these two images. However, all img elements should be considered when creating this rule. Parent elements can be considered as a solution if no distinguishing feature can be found. In layout design, several HTML tags such as div, span, figure, li, article, aside, etc. can be used to design the arrangement of visual elements on a page. These tags are used as the parent of an img element. For example, the class = "easy_img_caption" attribute might be a distinguisher for the relevant image in the example. When these possibilities are tested and the appropriate value is selected, an extraction rule is obtained for this web page. However, whether this extraction rule obeys all web pages on the website is a problem that needs to be investigated separately. As you can see, determining the relevant image only through HTML elements is an error-prone process.

A website consists of many different layouts such as homepage, link page, content page, gallery page, video page, etc. A web page, such as the homepage and link web page, does not contain any relevant images. A content web page may contain one or more images. A gallery web page usually contains more than one relevant image. For these reasons, it is another challenge to automatically identify relevant images on a website.

IV. CRAWLER AND FEATURES

Determining the features is one of the crucial issues for the prediction model produced through machine learning methods. Another crucial issue is to collect data for these features. In this study, the web crawler whose flow chart is given in Fig. 3 is developed for obtaining data and extracting features from this data.

Our crawler begins with one or more URLs given by a developer. Then, it fetches the web page at that URL. This page is parsed to extract both images (Image Extractor) and the links from the page (URL Extractor). Image Extractor returns all img elements in the web page. URL Extractor finds all links on the web page and stores these links to the storage of URLs frontier. In this storing process, URLs that are outside of the website are filtered. Besides, the crawler checks for Visited URLs / URLs frontier and does not append the existing URL to URLs frontier. The scheduler gets a URL

TABLE 1. Example Elements and Relevant Status.

No	Image	Parent Element 1	Parent Element 2	Relevant
1		<span class="easy_img_caption">	<div>	1
2		<a href="...">	<div class="footer-icons">	0
3		<a href="...">	<div class="entry-short">	0
4		<a href="..." itemprop="image">	<div>	1
5		<span class="field-value">	<dd class="field-entry">	0

from storage and calls the downloader to start the process again.

A classic crawler does not download images. The crawler we have developed downloads the images one by one and collects data for the features used for machine learning methods. These data are kept in a tab-delimited file that contains features separated by tab characters. In this study, these features are divided into three categories.

- Textual features
- Image features
- Last features

Textual features (1 in Fig. 3) are obtained directly from the HTML elements on the web page. Here, it is determined whether the attribute keys in the element exist or not. The attribute values are not used as they differ between websites. Firstly, the data of these features are added to the tab-delimited file.

Image features (2 in Fig. 3) are the data obtained from the image. If the image is not in the image storage, it will be downloaded. Otherwise, the image information is taken from this storage. This information keeps values such as height, width, file size, the file extension of the image. However, features for an img element are not kept here. Because features extracted from an img element may differ between web pages. For this reason, the src value is used as the key in the image storage process. All search operations are carried out on this key. That is, key/value (src/image information) pairs are stored in image storage. In storage, we use a hash map data structure so the time complexity of the searching process is O(1). If the scr value is in the image store, the cached feature is set to 0, otherwise 1. The rows according to the data of these features in the tab-delimited file are updated.

Last features (3 and 4 in Fig. 3) are generated after all images are downloaded. Except for one of these features, the numerical data obtained from all images are used. The cluster feature utilizes the textual data of Visited URLs. The data of last features are calculated after obtaining the data of textual and image features. Finally, the tab-delimited file is updated.

A. TEXTUAL FEATURES

An image element contains the img tag and attributes that provide additional information about this image. The most important feature is src that specifies the path to the image. Other attributes such as alt, class, id, height, width, style, sizes, style are optional. These attributes can have different values even on a website. Table 1 indicates image elements, two-parent elements, and their relevant status.

TABLE 2. Textual Features.

Feat. Name	Description	Type	Example
1. id	Is id attribute in the img element?	boolean	0
2. elm_class	Is class attribute in the img element?	boolean	0
3. style	Is style attribute in the img element?	boolean	1
4. alt	Is alt attribute in the img element?	boolean	1
5. align	Is align attribute in the img element?	boolean	0
6. itemprop	Is itemprop attribute in the img element?	boolean	0
7. ariahidden	Is ariahidden attribute in the img element?	boolean	0
8. attr_height*	Height value if present in the img element	integer	-1
9. attr_width*	Width value if present in the img element	integer	100
10. len_Attrs*	Number of attributes on the img element	integer	3
11. len_Page*	Number of characters on the web page	integer	49612
12. len_src*	Number of characters on the src attribute of the img element	integer	27
13. parent1_tag	first parent tag of the img element	nominal	span
14. parent2_tag	second parent tag of the img element	nominal	div

Example values are for Table I - No.1
* Proposed Features

Some of the attributes are given in Table 1. The web designer is free to set the feature names for himself. On the other hand, the value of the attribute can be different for each element on a web page, as well as from website to website. For this reason, when specifying features, if the attribute value is a string, it is selected whether the element contains the attribute or not, and if the attribute value is an integer, it is chosen numerically. Table 2 gives features obtained from elements.

Table 2 contains a boolean, integer, and nominal data types. Features that return a Boolean value, if the attribute name is found or not in the img element. To make this value suitable for machine learning, we convert Boolean values (True or False) to number (1 or 0). Nominal values are string, but they represent a fixed set. For example, the parent elements can have figure, html, div, body, a, td, span, li, noscript, etc.

TABLE 3. Image Features.

Feat. Name	Description	Type	Example
15. cached*	Is the src attribute of the img element in the Image Storage	boolean	0
16. file_ext	File extension of the src attribute	nominal	.jpg
17. img_Pos_Page*	The index of the img element in the web page	integer	1119
18. ratio_imgPos_Page*	= img_Pos_Page / len_Page	float	0.023
19. Width	The width attribute specifies the width of an image.	integer	750
20. Height	The height attribute specifies the height of an image.	integer	470
21. WidthHeight	=Width * Height	integer	352500
22. ratioWH	= Width / Height	float	1.596
23. FileSize	A measure of how much storage it consumes in a file system.	integer	216427

Example values are taken from the dataset for Table I - No.1.
* Proposed Features

TABLE 4. Last Features.

Feat. Name	Description	Type	Example
24. ratio theimg to Pageimgs*	= FileSize of the image / Mean size of images files in the web page	float	2.967
25. ratio theimg to websiteimgs*	= FileSize of the image / Mean size of images files in the Web site	float	5.770
26. orderFileSize*	Sorting images in a web page by file size	integer	1
27. orderWidth*	Sorting images in a web page by width	integer	2
28. orderHeight*	Sorting images in a web page by height	integer	1
29. orderWidth-Height*	Sorting images in a web page by width * height	integer	1
30. cluster*	Cluster of the URL in visited URLs	integer	2

Example values are taken from the dataset for Table I - No.1.
* Proposed Features

tags. We list the nominal values, and then prepare them for machine learning by giving an index. That is, we convert nominal values to a numerical value. Integer features provide numerical values about the element.

In the literature, the size and height of the image file are generally used. However, a web designer can change the width and height attributes within the img element to change the size of the image shown in the web browser. In this study, the attr_width and attr_height features are proposed. Also, the number of attributes in the element, the number of characters on the entire web page, and the number of characters on the scr attribute have been added to our feature set. Finally, Table 2 presents the values obtained for the first example in Table 1.

B. IMAGE FEATURES

Our crawler has a cache mechanism to store locally information of images. The crawler uses the src attribute as a key. If the src attribute is not in the cache storage then it is downloaded from the website. If the src attribute was previously cached, the crawler loads the image information. The value of the cached feature, proposed in this study, is determined in this case. Moreover, this mechanism adds performance gains and minimizes bandwidth consumption. Table 3 gives image features obtained from this stage.

Features of Width, Height, WidthHeight, ratioWH, FileSize, and file_ext are commonly used features in the literature. The file_ext is nominal attribute that can have.jpg,.jpeg, .jfif,.pjpeg, .pjp,.png,.gif,.svg,.ico, etc. extensions. Features img_Pos_Page and ratio_imgPos_Page, proposed in this study, provide information about the position of the image on the web page. In particular, the ratio_imgPos_Page feature III gives information about the approximate position of an image on the web page. This feature has a float value. Width and Height features are the values obtained from the image file. These features should not be confused with attr_width and attr_height. The web designer has a chance to alter the attr_width and attr_height values on the web page. But it cannot change the width and height of the image file without using an image processor.

C. LAST FEATURES

After all images are processed for a web page, new features are computed for those images. Since this is the final stage in feature extraction, it is named as last features given in Table 4.

In Table 4, ratio features are the comparison of the image with other images in terms of file sizes. Values of order features obtain from sorting files according to various criteria. The sorting is in descending order. The cluster feature has a value derived from visited URLs for a website. To produce these values, the Levenshtein distances [30] are calculated for all visited URLs. The Levenshtein distance is a well-known string metric for measuring the difference between two strings. A comparison matrix is created that contains all distances for visited URLs. Then, we use DBSCAN (Density-based spatial clustering of applications with noise) [31] which groups together points that are close to each other.

A crawler begins with a seed/seeds (URL / URLs) for the crawling process. A seed's web page contains images, but none of these images are relevant images. A crawler keeps collecting URLs from other web pages. There are no relevant images on the seeds-like pages. Here, distance and DBSCAN are used to detect page similarity. In this study, instead of using data of web pages, only URL is used for this task. The information gain of all features will be examined in the experimental section.

V. EXPERIMENTS

In this section, firstly information about the dataset and annotating of relevant and irrelevant images are given. The second subsection presents the performance metrics used in this study. The third subsection gives brief information about machine learning methods. The last subsections cover the performance results of experiments.

TABLE 5. Information About Dataset.

	Training Dataset	Testing Dataset	Total / Average
Number of websites	150	50	200
Number of images	500284	134731	635015
Number of Relevant images	5582	17100	22682
The average of web page sizes	269KB± 285KB	167KB± 191KB	244KB± 268KB
The average number of images per a web page	28±32	36±30	34±31
The average of image file sizes	58.48KB± 384.51KB	29.99KB± 183.33KB	37.12KB± 233.62KB

A. DATASET

Previous studies were on a single web page and well-known websites. In this study, we develop a web crawler that downloads web pages from 200 websites. These websites belong to 58 different countries including Albania, Australia, Austria, Azerbaijan, Bahrain, Bangladesh, Belarus, Bolivia, Bosnia and Herzegovina, Botswana, Brazil, Bulgaria, Cameroon, Canada, China, Cuba, Czechia, Egypt, Finland, France, Germany, Greece, Guatemala, India, Indonesia, Iran, Italy, Japan, Jordan, Kazakhstan, Kyrgyzstan, Laos, Latvia, Liberia, Macedonia, Madagascar, Malaysia, Mexico, Montenegro, Nepal, New Zealand, Pakistan, Philippines, Romania, Russia, Slovakia, Spain, Tanzania, Turkey, Uganda, Ukraine, Ukraine, USA, Uzbekistan, Venezuela, Vietnam, Zambia, and Zimbabwe. 100 web pages are downloaded for each website. So 20,000 web pages are collected. A dataset consisting of 635,015 images is created from these web pages. 22,682 of these images are relevant images. To apply machine learning methods, a dataset should be annotated. We have utilized a double-blind annotation technique for constructing a reliable dataset. To annotate relevant images, 5 web experts for 200 websites have set up rules to get relevant images. All images except the relevant images are marked as irrelevant. Besides, at least 2 web experts checked each site whether the rules were correct. Table 5 gives information about the dataset used in this study.

In this study, our dataset is divided into training and testing datasets to evaluate the performance of our model. The training dataset is a subset to train a prediction model employing machine learning methods. The testing dataset is a subset to predict data with the trained model. However, machine learning methods can be overfitting because there are too many similar web pages for websites. To prevent overfitting, 150 of the websites are used as a training dataset and the other 50 websites as testing dataset.

B. EVALUATION METRICS

There are several metrics to evaluate the performance of machine learning methods. This study focuses specifically on predictive performance on relevant images. Because our dataset is imbalanced. In other words, the number of relevant images in the dataset is much higher. A balanced dataset contains an equal number of samples for each class. However, only 3.57% of the images are included in the relevant class in our dataset. For this reason, our metrics are about evaluating the relevant image correctly.

Accuracy is the ratio of correct predictions to total predictions. However, since we have an imbalanced dataset, we get an accuracy of 0.964 when we annotate all images as irrelevant. Therefore, f-Measure is used to understand the prediction performance of the related images. The f-Measure is the weighted average of Precision and Recall. Precision is the ratio of correctly predicted relevant images to the total predicted relevant images. Recall is the ratio of correctly predicted relevant images to all images in actual class - relevant. Let's define equations of Precision(1), Recall(2), F-measure(3) and Accuracy(4) metrics, respectively.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$f - Measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

where TP (True Positives) is the number of correctly predicted relevant images, TN (True Negatives) is the number of correctly predicted irrelevant images, and FP (False Positives) and FN (False Negatives) are the numbers of error predictions. In FP, when actual class is an irrelevant image but the model prediction is a relevant image. In FN, when actual class is a relevant image but the model prediction is an irrelevant image.

Moreover, Log Loos [32] is a useful metric to evaluate the performance of machine learning methods. The output of this metric is a probability value between 0 and 1. Log Loos can be calculated as (5).

$$LogLoss = -(ylog(p) + (1 - y)log(1 - p)) \tag{5}$$

where y is a binary indicator (0 or 1) if class is the relevant image in prediction for observation and p is predicted probability observation is of class. Log loss measures the uncertainty of the probabilities of the prediction model. A lower log loss value means better predictions.

C. MACHINE LEARNING METHODS

In this study,primarily traditional machine learning methods including Naive Bayesian (NB) [33], Support Vector Machines(SVM) [34], K-Nearest Neighbours (KNN) [35], and Decision Tree (J48) [36] are used to create the prediction model. Then, we use two popular ensemble methods, Random Forests (RF) [37] and Adoboost [38], to improve the performance results of the prediction model. In this section, general information is given about these methods. In experiments, we give the performance results of these methods.

NB is a simple supervised machine learning method based on performing Bayes' theorem with naive independence assumptions between the features. In the implementation of

NB, we employ Gaussian NB (GNB) that follows Gaussian (normal) distribution. GNB supports continuous-valued features and models each as conforming to this distribution. SVM is a discriminative machine learning method formally expressed by a separating hyperplane. Different kernel functions, namely the linear, polynomial, and the Gaussian radial basis function (RBF), can be applied to SVM. We use the RBF kernel that is especially useful when the data-points are not linearly separable. Moreover, this function has been implemented and proposed by Vyas and Frasincar [9] to find the representative image. KNN is a non-parametric method that calculates weights for features. This method depends on distance for classification, normalizing the training dataset can enhance its accuracy score. J48 (C4.5) is a popular classifier to construct a decision tree that is a map of the possible classes. This tree is used for the prediction of class and observable in a model. With this tree, the explanation for the conditions is easily indicated by boolean logic.

We have an imbalanced dataset where classical methods can have some problems. Recently, ensemble methods provide very successful solutions in addressing this problem [39]. In machine learning, ensemble methods utilize multiple machine learning methods to generalizability/robustness over a machine learning method. In this study, we use RF and AdaBoost which decrease the variance of a single machine learning method as they combine several machine learning methods. As a result, the performance of the learning model increases, and the predictions are much more robust and stable. RF classifier is an ensemble learning method that uses multiple learning algorithms to create a better prediction model. RF classifier is a classification method that contains a combination of tree predictors. Boosting is a sequential ensemble method that decreases the bias error and constructs well-built models. The Boosting term means to turn a weak learner into a strong learner. In this study, we utilize the popular boosting algorithm AdaBoost [40], developed by Schapire and Freund that won the prestigious 2003-Gödel Prize for this algorithm.

In experiments, feature selection [41] is the process of choosing a subset of relevant features by eliminating redundant features. In the present study, we use information gain [36] which can be used in feature selection, by evaluating the gain of each feature in the context of the target feature. This gain shows us the importance of a feature.

D. PERFORMANCE OF MACHINE LEARNING METHODS AND SIMPLE HEURISTIC TECHNIQUES

The number of irrelevant images on websites is huge. For this reason, an imbalanced dataset has been created for this task. In our first experiment, traditional and ensemble machine learning methods are evaluated to understand the impact of this imbalanced dataset. Table 6 shows the performance results for the methods. Moreover, we try some simple heuristic techniques including the biggest filesize, widths, heights, width*height, and width*height greater than 120,000 [22] to highlight the difficulty of the problem.

TABLE 6. The Performance of Machine Learning Methods and Some Heuristic Techniques.

Method/s	Precision	Recall	f-Measure	Accuracy	Log Loss
AdaBoost	0.827	0.788	0.807	0.984	0.538
RF	0.795	0.792	0.794	0.983	0.590
J48	0.488	0.695	0.573	0.957	1.481
KNN	0.413	0.630	0.499	0.948	1.811
SVM	0.569	0.344	0.429	0.962	1.312
NB	0.245	0.592	0.346	0.907	3.198
Biggest Image File	0.466	0.437	0.451	0.956	1.524
Biggest Widths	0.318	0.624	0.422	0.939	2.450
Biggest Heights	0.291	0.607	0.393	0.922	2.684
Biggest WH	0.312	0.622	0.415	0.928	2.505
WH >= 120,000	0.211	0.958	0.345	0.850	5.199

In Table 6, it can be seen that simple heuristic methods are not sufficient for this task, as websites consist of many different web pages. Bhardwaj and Mangat's [22] proposed width*height greater than 120,000 is not suitable for determining the relevant image on many web pages. This technique has the worst performance results. Selecting the largest file size as the relevant image gives the best performance results compared to other techniques.

When the performances of machine learning methods are examined through accuracy, it is seen that the accuracy results are close to each other. However, the numbers of relevant and irrelevant images are very disproportionate. In other words, it is an imbalanced dataset. For this reason, it is better to interpret the results in terms of f-Measure and log loss.

NB is the worst classifier in terms of performance compared to others. On the other hand, it is even lower than the performance of heuristic methods. NB assumes that all the features are mutually independent. These results indicate that features are not independent. SVM is dramatically the second-worst machine learning method, although it has been suggested in the literature. Considering that Vyas and Frasincar [9] reaches an F-Measure of 0.439 to find the representative image in a different dataset, our performance results are very similar for a different dataset and task. As seen in Table,SVM is insufficient for this task. KNN and Decision Tree methods do not give good performance results. The performance of KNN depends on the quality of the data. Although decision tree classifier gives successful performance results from web data extraction studies [5], [42]–[44], this classifier is inadequate in terms of the prediction model in this study. Unfortunately, our imbalanced dataset prevents constructing the desired prediction model. It also shows the difficulty of enhancing the prediction model with classical machine learning methods due to the imbalanced dataset. In this case, the ensemble methods are recommended in the literature.

According to Table 6, AdaBoost and Random Forests give the best results for this task. Recent studies on web data extraction have shown that AdaBoost [13] and Random Forests [45] give good results in different imbalanced datasets. Although the aim of these studies is not to find the relevant image, it is seen that there are similar results.

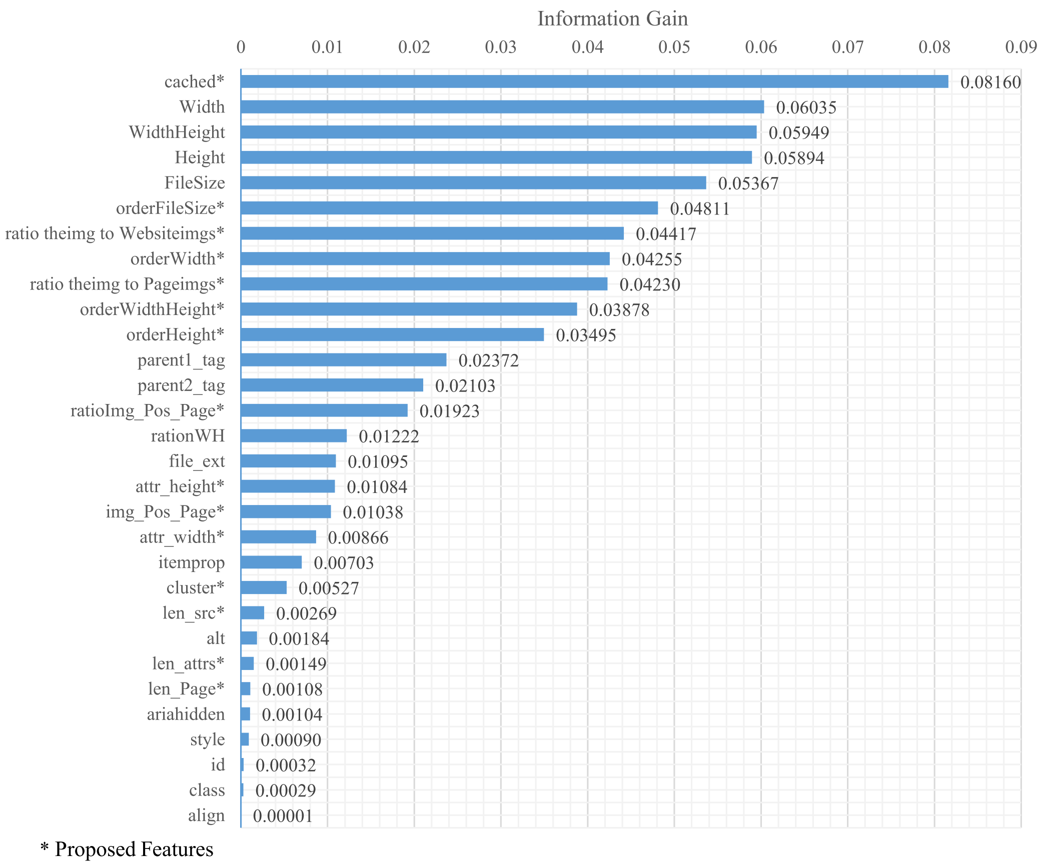

FIGURE 4. Information gain scores of training dataset.

Ensemble methods appear to have a positive impact on performance results. In this way, especially the f-Measure score is improved for the prediction of relevant images. These methods change the habit of overfitting to their training set.

E. COMPARISON OF FEATURE CATEGORIES

In Section IV, features are divided into three categories including textual features, image features, and last features. This section examines the contributions of these three categories. In this experiment, firstly, the prediction model is created using only features in the category and evaluating on the testing dataset. For the prediction model, the AdaBoost classifier is applied, which gives the best performance results in Experiment 1. Then, features in categories are combined order to try to discover the importance of groups. Table 7 indicates the performance results obtained with these categories.

The performance of the prediction model constructed by textual features are poor. It is seen that textual features do not contribute to the model due to the large variability of HTML elements. The prediction model based on the image features gives very good results. In particular, this category covers most of the features such as height, width, and file size recommended in previous studies. These features positively affect the performance of the prediction model. According to the performance results, it is seen that the Last features contribute more than textual features to the prediction model. That is, the image and last features are very necessary and important for this task. Interestingly, the sub-dataset containing the image and last features yield the best results with a score of 0.822 f-Measure. In other words, there is no need to use textual features when creating a prediction model. The next section examines the features in more detail.

TABLE 7. Comparison of Feature Groups.

Group/s	Precision	Recall	f-Measure	Accuracy	Log Loss
C1	0.232	0.146	0.179	0.945	1.915
C2	0.759	0.771	0.765	0.980	0.678
C3	0.709	0.324	0.445	0.966	1.158
C1 - C2	0.818	0.661	0.731	0.980	0.696
C2 - C3	0.836	0.809	0.822	0.986	0.500
C1 - C3	0.609	0.539	0.572	0.967	1.155
C1 - C2 - C3	0.827	0.788	0.807	0.984	0.538

C1: Textual features, C2: Image Features, C3: Last Features

F. EVULATION OF FEATURE SELECTION

In this study, we use the scores of information gain to determine the most influential features of our models. Information gain is widely applied for the feature selection process. Fig. 4 shows the information gain scores for each feature.

According to Fig. 4, six of the features including cached, orderFileSize, ratio theimg to websiteimgs, orderWidth, ratio

TABLE 8. Comparison of Experiments in Feature Selections.

Group/s	Precision	Recall	f-Measure	Accuracy	Log Loss
Top 5 features	0.634	0.881	0.738	0.974	0.898
Top 10 features	0.725	0.677	0.700	0.976	0.831
Top 15 features	0.812	0.738	0.773	0.982	0.620
Top 20 features	0.803	0.765	0.784	0.982	0.604
Top 25 features	0.836	0.788	0.811	0.985	0.525
Except Cached	0.820	0.714	0.763	0.982	0.633
Except New Features	0.848	0.478	0.611	0.975	0.869
Only New Features	0.655	0.753	0.701	0.973	0.920
All Features	0.827	0.788	0.807	0.984	0.538

theimg to Pageimgs, orderWidthHeight proposed in this study are among the top 10 features. In particular, the most prominent feature on a website that keeps whether the image is downloaded during the crawling process is the cached feature. The features including Width, Height, WidthHeight, and FileSize recommended in previous studies come after the cached feature. The ratio theimg to websiteimgs and ratio theimg to Pageimgs features, which show the importance of an image according to images on the website and web page, are among the important features. The order features suggested in our study support the construction of a better prediction model.

To examine the performance contribution of the features, the features are divided into groups of 5 according to the information gain scores. These groups are added to the prediction model, starting with the top 5 features. The AdaBoost classifier is used in the creation of prediction models. Performance scores obtained by evaluating these models with the same testing dataset are given in Table 8.

The performance results in Table 8 indicate that the prediction model created by the top 25 features is better than others. In other words, these results show that the last five features are unnecessary for this task. The f-Measure score of the prediction model derived from traditional features is 0.611. In terms of performance results, even the prediction model established with new features is better than the prediction model established with traditional features. The f-measure score of the model created with new features is 0.701. Thanks to the new features proposed in the present study, the performance of the prediction model reaches an f-Measure score of 0.807. Using the last and image features, the prediction model performance reaches 0.822 f-Measure. In other words, using new features enhances performance results by 35%. If the cached feature, which is the most important feature among the information gain values, is not used, the f-Measure score of the model is 0.753. In other words, this feature increases the f-Measure by 0.059 points according to the best model performance result. That is, using this feature provides a 7% improvement in the prediction model.

VI. CONCLUSION

In web data extraction studies, finding suitable features, creating large datasets, and evaluating appropriate machine learning methods are important issues. Conventional studies focus only on a web page. Vyas and Frasincar [9] state that the performance results could get better by using multiple web pages. In our study, a web crawler was developed that can easily create a dataset and extract new features from web pages. A crawler offers a lot of information about web data extraction [10]. With this crawler, a large dataset of 20,000 web pages from 200 websites was created. These new features have increased the performance of the prediction model. In particular, the cached feature obtained from the cache mechanism contributes to the prediction model.

In the present study, we have an imbalanced dataset of 635,015 irrelevant and 22,682 relevant images. This study shows that ensemble methods in machine learning are suitable for constructing accurate models in the imbalanced dataset. It is seen that these methods give much better performance results than conventional methods. Especially the Adaboost classifier gives the best performance results. When all the features are used, the performance results of the prediction model obtained from this classifier has an F-Measure score of 0.807.

The performance results of a prediction model can be increased by removing unnecessary features. In machine learning literature, this process is called feature selection. In this study, feature selection is applied to two different approaches. The first approach examines the feature categories presented in the study separately. In this approach, our experiments indicate that textual features do not contribute much to the classification performance. The f-Measure score of the prediction model created without using textual features is 0.822. In the second approach, the features are listed according to their information gain values. Then, the performance results of models created from sub-datasets are evaluated. In this second approach, a 0.811 f-Measure score is achieved with the top 25 features.

Semantic web technologies enable us to write rules for handling data. The schema.org community sets standard rules to make it easier for search engines in the web data extraction. However, most of the web designers do not follow these rules. Some of them interpret these rules on their requirements. For this reason, the desired development in the Semantic web has not been achieved despite the years of efforts. Automatic web data extraction works can speed up the Semantic web creation process. This study contributes to the identification of relevant images from web pages.

Our web crawler is an open-source project and available via the web page https://github.com/erdincuzun/ImageCrawler. Besides, the dataset in this study is available via the web page https://adys.nku.edu.tr/Datasets/. With the web crawler, the dataset can be enlarged or new datasets can be created for studies that need image datasets. For future works, we plan to perform three different studies. The first study is to develop a regular expression generator that uses negative and positive examples of images. In the second study, we aim to find the relevant image only through img elements without downloading images. In the last study, we will try to determine the relevant image through the texture and colour information used in different studies [18], [19], [46]. While doing these studies, the best f-Measure score in our study will be considered as a baseline.

REFERENCES

[1] Y. Xue, Y. Hu, G. Xin, R. Song, S. Shi, Y. Cao, C.-Y. Lin, and H. Li, "Web page title extraction and its application," *Inf. Process. Manage.*, vol. 43, no. 5, pp. 1332–1347, Sep. 2007.

[2] Y. Hu, G. Xin, R. Song, G. Hu, S. Shi, Y. Cao, and H. Li, "Title extraction from bodies of HTML documents and its application to Web page retrieval," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2005, pp. 250–257.

[3] F. Sun, D. Song, and L. Liao, "DOM based content extraction via text density," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 245–254.

[4] Q. Liu, M. Shao, L. Wu, G. Zhao, G. Fan, and J. Li, "Main content extraction from Web pages based on node characteristics," *J. Comput. Sci. Eng.*, vol. 11, no. 2, pp. 39–48, Jun. 2017.

[5] E. Uzun, E. S. Güner, Y. Kılıçaslan, T. Yerlikaya, and H. V. Agun, "An effective and efficient Web content extractor for optimizing the crawling process," *Softw., Pract. Exper.*, vol. 44, no. 10, pp. 1181–1199, Oct. 2014.

[6] E. Ucar, E. Uzun, and P. Tüfekci, "A novel algorithm for extracting the user reviews from Web pages," *J. Inf. Sci.*, vol. 43, no. 5, pp. 696–712, Sep. 2017.

[7] K. Suleman and O. Vechtomova, "Discovering aspects of online consumer reviews," *J. Inf. Sci.*, vol. 42, no. 4, pp. 492–506, Aug. 2016.

[8] N. Gali, A. Tabarcea, and P. Fränti, "Extracting representative image from Web page," in *Proc. 11th Int. Conf. Web Inf. Syst. Technol.*, 2015, pp. 411–419.

[9] K. Vyas and F. Frasincar, "Determining the most representative image on a Web page," *Inf. Sci.*, vol. 512, pp. 1234–1248, Feb. 2020.

[10] E. Uzun, "A novel Web scraping approach using the additional information obtained from Web pages," *IEEE Access*, vol. 8, pp. 61726–61740, 2020.

[11] H. K. Azad, R. Raj, R. Kumar, H. Ranjan, K. Abhishek, and M. P. Singh, "Removal of noisy information in Web pages," in *Proc. Int. Conf. Inf. Commun. Technol. Competitive Strategies (ICTCS)*, 2014, pp. 1–5.

[12] A. Stevanovic, B. Xue, and M. Zhang, "Feature selection based on PSO and decision-theoretic rough set model," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2840–2847.

[13] E. Uzun and E. Ozhan, "Examining the impact of feature selection on classification of user reviews in Web pages," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 430–437.

[14] M. L. Kherfi, D. Ziou, and A. Bernardi, "Image retrieval from the world wide Web: Issues, techniques, and systems," *ACM Comput. Surv.*, vol. 36, no. 1, pp. 35–67, Mar. 2004.

[15] L. Nie, M. Wang, Z.-J. Zha, and T.-S. Chua, "Oracle in image search: A content-based approach to performance prediction," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, pp. 1–23, May 2012.

[16] L. S. Kennedy and M. Naaman, "Generating diverse and representative image search results for landmarks," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, p. 297.

[17] J. Z. Wang, J. Li, G. Wiederhold, and O. Firschein, "Classifying objectionable websites based on image content," in *Interactive Distributed Multimedia Systems and Telecommunication Services* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1483. Berlin, Germany: Springer, 1998, pp. 114–124.

[18] N. Hor and S. Fekri-Ershad, "Image retrieval approach based on local texture information derived from predefined patterns and spatial domain information," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 6, pp. 246–254, Nov./Dec. 2019.

[19] N. T. Bani and S. Fekri-Ershad, "Content-based image retrieval based on combination of texture and colour information extracted in spatial and frequency domains," *Electron. Library*, vol. 37, no. 4, pp. 650–666, Aug. 2019.

[20] J. I. Helfman and J. D. Hollan, "Image representations for accessing and organizing Web information," in *Internet Imaging II*, vol. 4311. Bellingham, WA, USA: SPIE, 2000, pp. 91–101.

[21] J. Hu and A. Bagga, "Functionality-based Web image categorization," in *Proc. 12th Int. World Wide Web Conf.*, Budapest, Hungary, May 2003, pp. 1–9.

[22] A. Bhardwaj and V. Mangat, "An improvised algorithm for relevant content extraction from Web pages," *J. Emerg. Technol. Web Intell.*, vol. 6, no. 2, pp. 226–230, May 2014.

[23] I. A. A. Sabri and M. Man, "WEIDJ: An improvised algorithm for image extraction from Web pages," in *Proc. 8th Int. Conf. Inf. Technol. (ICIT)*, May 2017, pp. 512–517.

[24] S. Noh, Y. Choi, H. Seo, K. Choi, and G. Jung, "An intelligent topic-specific crawler using degree of relevance," in *Intelligent Data Engineering and Automated Learning—IDEAL 2004* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3177. Berlin, Germany: Springer, 2004, pp. 491–498.

[25] G. Pant and P. Srinivasan, "Link contexts in classifier-guided topical crawlers," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 107–122, Jan. 2006.

[26] S. Batsakis, E. G. M. Petrakis, and E. Milios, "Improving the performance of focused Web crawlers," *Data Knowl. Eng.*, vol. 68, no. 10, pp. 1001–1013, Oct. 2009.

[27] M. Kumar, R. Bhatia, and D. Rattan, "A survey of Web crawlers for information retrieval," *WIREs Data Mining Knowl. Discovery*, vol. 7, no. 6, p. e1218, Nov. 2017.

[28] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, p. 107.

[29] L. Zhang, F. Lin, and B. Zhang, "Support vector machine learning for image retrieval," in *Proc. Int. Conf. Image Process.*, vol. 2, 2001, pp. 721–724.

[30] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[31] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[32] V. Vovk, "The fundamental nature of the log loss function," in *Fields of Logic and Computation II* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9300. Cham, Switzerland: Springer, 2015, pp. 307–318.

[33] H. Zhang, "The optimality of Naive Bayes," in *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf.*, vol. 2, 2004, pp. 562–567.

[34] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[35] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[36] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[38] B. Soediono, "Bias, variance, and arcing classifiers," *J. Chem. Inf. Model.*, vol. 53, no. 1, p. 160, 1989.

[39] P. Lopez-Garcia, A. D. Masegosa, E. Osaba, E. Onieva, and A. Perallos, "Ensemble classification for imbalanced data based on feature space partitioning and hybrid metaheuristics," *Appl. Intell.*, vol. 49, no. 8, pp. 2807–2822, Aug. 2019.

[40] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*, vol. 42, no. 1. Cambridge, MA, USA: MIT Press, 2013.

[41] S. Sarangi, M. Sahidullah, and G. Saha, "Optimization of data-driven filterbank for automatic speaker verification," *Digit. Signal Process.*, vol. 104, Sep. 2020, Art. no. 102795.

[42] Z. Chen and J. C. Lv, "Web news pages extraction method based on DOM and decision tree," in *Recent Advances in Computer Science and Information Engineering* (Lecture Notes in Electrical Engineering), vol. 124. Berlin, Germany: Springer, 2012, pp. 149–154.

[43] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 441–450.

[44] E. Uzun, H. V. Agun, and T. Yerlikaya, "A hybrid approach for extracting informative content from Web pages," *Inf. Process. Manage.*, vol. 49, no. 4, pp. 928–944, Jul. 2013.

[45] E. Ozhan and E. Uzun, "Performance evaluation of classification methods in layout prediction of Web pages," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–7.

[46] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods—A review," *Int. Online J. Image Process. Pattern Recognit.*, vol. 2, no. 1, pp. 1–29, 2019.

ERDINÇ UZUN received the degree from the Computer Engineering Department, Trakya University, Edirne, Turkey, in 2001, and the master's and Ph.D. degrees from Trakya University, in 2003 and 2007, respectively. In 2007, he completed the Ph.D. thesis on the development of a web-based system that can automatically learn subcategorization frames. It is a thesis that combines the fundamental fields of computer science such as information retrieval, machine learning, and natural language processing.

In 2001, he started his academic career at the Computer Engineering Department, Trakya University, where he was a Research Assistant from 2001 to 2007. In 2007, he started his career at the Computer Engineering Department, Tekirdağ Namık Kemal University. From 2008 to 2010, he was the Vice Dean with the faculty. After 2007, he worked in web search, web mining, and web content extraction. He supports education not only with courses but also with lecture notes and blog posts shared on erdincuzun.com. He has also been a member of the Board of Directors of the Faculty since 2017. In 2019, he started to work as the Director of the Çerkezköy Vocational School. He was also a referee and panelist in various TÜBİTAK programs. He has more than 30 publications and more than 150 citations. He is currently supervising five graduate students. His research interests include developing a career that combines teaching and research while maintaining interest in information retrieval, data mining, and natural language processing.

ERKAN ÖZHAN received the degree from Technical Education Faculty, Firat University, Elazig, Turkey, in 2000, the master's degree in computer engineering and the Ph.D. degree from Trakya University, Turkey, in 2007 and 2013, respectively. He has taken many positions as a manager and a referee in research projects in artificial intelligence.

In 2001, he started his academic career at Trakya University, with a focus on studies on artificial intelligence and network security. Since 2013, he has been a full time Academic Member of the Corlu Faculty of Engineering, Computer Engineering Department, Tekirdağ Namık Kemal University. He conducted a project on the application of high performance computing to artificial intelligence and data mining. He has been a speaker at many meetings on artificial intelligence. He teaches undergraduate and graduate courses such us data mining, machine learning, big data, and artificial intelligence techniques. He has authored 14 articles in Turkish and English. His current research interests include artificial intelligence, data mining, big data, machine learning, high performance computing, and network security.

HAYRI VOLKAN AGUN was born in Erzurum, Turkey, in 1982. He received the B.S. degree in computer engineering from Anadolu University, Eskisehir, in 2001, the M.S. degree in computer engineering from Trakya University, Edirne, Turkey, in 2008, and the Ph.D. degree in computer engineering from Eskisehir Technical University, Eskisehir, in 2019. From 2005 to 2008, he was a Research Assistant with the Computer Engineering Department, Trakya University. From 2006 to 2008, he has collaborated in the projects of the Trakya Cognitive Science Society. He was a Data Scientist for the research projects of Dilisim - a big data and search services company located in Eskisehir. In 2019, he became an Assistant Professor (Dr.) with the Computer Engineering Department, Bursa Technical University. His research interests include natural language processing, text classification, and information extraction.

TARIK YERLIKAYA was born in Edirne, Turkey, in 1977. He received the from the Electronics & Communications Engineering Department, Yıldız Technical University, Istanbul, Turkey in 1999, and the master's and Ph.D. degrees from the Computer Engineering Department, Edirne, in 2002 and 2007, respectively. In 2007, he completed his Ph.D. thesis in cryptography.

In 1999, he started his academic career at the Computer Engineering Department, Trakya University. From 1999 to 2007, he was a Research Assistant with Trakya University. In 2008, he became an Assistant Professor (Dr.) with the Computer Engineering Department, Trakya University. His research interests include cryptography, natural language processing, text classification, and information extraction.

HALIL NUSRET BULUŞ was born in Vize, Kirklareli, Turkey, in 1981. He received the B.S., M.S., and Ph.D. degrees in computer engineering from Trakya University, Edirne, in 2003, 2006, and 2010, respectively.

From 2004 to 2009, he was a Research Assistant with the Computer Engineering Department, Engineering and Architecture Faculty, Trakya University. In 2010, he was an Assistant Professor with the Computer Engineering Department, Tekirdağ Namık Kemal University. He has authored more than 20 articles. His research interests include data compression and data processing.