Information Sciences 512 (2020) 1234–1248

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Determining the most representative image on a Web page

Krishna Vyas, Flavius Frasincar*

Erasmus University Rotterdam, PO Box 1738, Rotterdam, 3000 DR, the Netherlands

ARTICLE INFO

Article history:
Received 14 July 2019
Revised 9 September 2019
Accepted 25 October 2019
Available online 25 October 2019

Keywords:
Image search
Representative image
Support vector machines
Feature selection

ABSTRACT

We investigate how to determine the most representative image on a Web page. This problem has not been thoroughly investigated and, up to today, only expert-based algorithms have been proposed in the literature. We attempt to improve the performance of known algorithms with the use of Support Vector Machines (SVM). Besides, our algorithm distinguishes itself from existing literature with the introduction of novel image features, including previously unused meta-data protocols. Also, we design and attempt a less-restrictive ranking methodology in the image preprocessing stage of our algorithm. We find that the application of the SVM framework with our improved classification methodology increases the F_1 score from 27.2% to 38.5%, as compared to a state-of-the-art method. Introducing novel image features and applying backward feature selection, we find that the F_1 score rises to 40.0%. Lastly, we use a class-weighted SVM in order to resolve the imbalance in number of representative images. This final modification improves the classification performance of our algorithm even further to 43.9%, outperforming our benchmark algorithms, including those of Facebook and Google. Suggested beneficiaries are the search engine community, image retrieval community, including the commercial sector due to superior performance.

1. Introduction

We are now at the dawn of the digital era and the Web is abundantly filled with data. In order to handle this, research has been conducted on how to present appropriate information from the Web to the user. This information comes in many forms, such as text, video, audio, and last, but certainly not least, images. Images are a vital part of the user-experience on the Web, since they allow individuals to transfer a large amount of information at a mere glance. Besides, images are more likely to capture the attention of an audience than just text alone. Given the fast-paced environment we live in, Web users often do not have much time to carefully read a Web page, let alone watch a video or listen to an audio track. Frequently, Web users quickly scan over a Web page to see whether it interests them. Images are the key to resolving such issues since they get the message across at the blink of an eye. Finally, it can be expected that the role of images will rise even further in the coming years with the growth of e-commerce and digital marketing [35].

An important area of application is summarising a large amount of information, such as a Web page, using a single image. For clarity, we define such a *representative image* as the image that best represents the content on a Web page to users, in accordance with [5]. This is a highly subjective task and it needs a proper definition. There are various approaches, such as ones focused on summarisation [26], interestingness [7], memorability [13] and diverseness [12]. Our research focuses

* Corresponding author.
E-mail address: frasincar@ese.eur.nl (F. Frasincar).

https://doi.org/10.1016/j.ins.2019.10.045
0020-0255/© 2019 Elsevier Inc. All rights reserved.

on determining the image that best *summarises* the content of the Web page. Accordingly, we refer to this image as the *representative image*.

It is important here to make the distinction between a Web page and a website, namely that a website comprises a set of several Web pages. In our research, we aim to determine the representative image from a specific Web page, and not an entire website. Extracting a representative image is of use when, for example, restrictions on band-width come into play, in which case it is not an option to extract all images [8]. Next to this, the rise of social media has triggered research in obtaining the representative image of a Web page. More concretely, applications such as Facebook and Google+ attempt to obtain the representative image from a Web page when a hyperlink is entered. This is done in order to promptly inform users of the contents of the Web page the hyperlink refers to, summarising the search result. Another major area of application is location-based services (LBS), which are applications that use geographical location to facilitate users in finding nearby services, such as restaurants and sport facilities. Well-known examples of such services are Foursquare, Yelp, and MOPSI, to name a few. Next to this, many large corporations, such as Groupon, Facebook Places, and Google Latitude, have also introduced LBS to their services. Selecting the representative image is of great importance for a LBS service, since images are highly informative when a user is, for example, looking for a nearby restaurant.

This research responds to the rising importance of retrieving a representative image from a Web page. We use work previously conducted in this field as a starting point. Previous work has focused on rule-based and Web-category specific methods, but has disregarded statistical learning methods and recent developments in the Semantic Web. We attempt to improve the known methodologies by increasing the accuracy with which the correct representative image is selected. The novelty of our framework lies in the fact that we make use of statistical learning algorithms in combination with the introduction of novel image features, previously unused meta-data protocols and an improved ranking methodology in the image preprocessing stage of our algorithm. To our knowledge, the application of machine learning algorithms (ML) in obtaining the representative image from a Web page has not been studied in literature yet. We propose a method that is, unlike almost all of the previous research, template-independent and is not specific to a certain Web category. More concretely, we attempt to improve upon the methodological framework of previous literature and the most prominently used algorithms for finding the representative image, namely the algorithms used by Google+ and Facebook. This leads us to the following main research question: *How can we most accurately determine the representative image on a Web page, that best summarises the page content?* We approach the problem from a statistical learning perspective, using ML algorithms. Consequently, an aspect that further increases the complexity is that ML algorithms require certain input variables, which, in our case, relate to the image. These input variables attempt to capture information that could be related to the representativeness of an image including intrinsic features such as height, width, and format, and extrinsic features such as the location of the image in the Document Object Model structure (DOM). However, unlike prior research, we do not limit ourselves to these and expand the scope to, amongst others, include previously unused meta-data protocols, intrinsic image features such as colour compositions and creative novel features such as the duration the image has been presented on the Web page, using a repository of cached Web pages. After applying the ML algorithm, information regarding the importance of the variables for selecting a representative image can be retrieved.

2. Related work

Studies have long advocated research in the area of image retrieval from the Web. A prominent example is the paper by Kherfi et al. [18], which outlines both the importance of Web image retrieval and a collection of developed image retrieval algorithms, which are based on textual user queries. Essentially, these algorithms attempt to search the Web for images that are most representative of the textual query. With increasingly complex queries, studies such as [29] have explored the usage of visual- and semantic concepts for image search. Furthermore, the challenge of performance prediction for image search has been explored by Nie et al. [28]. However, it is only in the past few years that the importance of determining and retrieving a representative image from specific Web pages has been recognised, rather than the entire Web [41]. On the one hand, this problem is more complex, due to the fact that the algorithms are not given a search query to use and must make use of other features to extract the most representative image. On the other hand, however, the search area is only one Web page and is thus much more limited than considering the entire Web. It is important to note, however, that a given Web page might not have a representative image. That is, even though a Web page contains images, none of them might summarize the content adequately. The problem is complicated even further due to the inevitable fact that there is subjectivity involved when deciding which image is representative, implying that it is simply impossible to obtain an algorithm that would perform perfectly as judged by every individual. Researchers have attempted to tackle this problem from several perspectives. In this section we discuss three of the most common approaches in literature and practise. An extensive overview of the state of the art in content-based image retrieval is presented by Lew et al. [21].

2.1. Rule-based methods

Academic research in this field started by employing rule-based methods, which make use of expert knowledge in selecting representative images. Essentially, rule-based methods are approaches where the scheme used for deciding upon the representativeness of an image is defined a priori by researchers based on their understanding of what characteristics such an image should have. One of the earliest rule-based methods developed for selecting the representative image has come from [8]. They approach the problem from an ordinal perspective, using an empirically obtained rating-scheme. More specifically, they found larger, squarer, and more colourful images to be representative, and thus rated images based on these features.

Most of the current research has generated fragmented solutions, searching for the representative image based on a single-sided perspective. More concretely, most studies either produce methods that are template- or Web-category specific [2,14] or use an incomplete approach by discarding certain useful information, such as the text surrounding the images [5]. The paper by Gali et al. [5] is, in fact, one of the most recent papers on this topic that did attempt to combine perspectives. The key idea applied was to combine the categorical approach introduced by Hu and Bagga [10] with the parsing of the website's DOM structure as described in [6]. In essence, the images are ranked in two steps, after which the highest ranked image is selected as representative. Firstly, the images are classified into five categories with a defined priority, after which the images are ranked within these categories based on descriptive features such as size, format, and HTML attributes. In this case, the expert knowledge incorporated was that the category that an image related to was of great importance when selecting the representative image. The lower image categories were only considered if no image was present in the higher categories. Next to this, the features that were chosen to be decisive, as well as their respective weights, were also based on expert knowledge, such as the choice of giving images of the `jpeg` format greater weight than images of the `svg`, `png`, and `gif` formats. The paper by Gali et al. [5] was able to produce strong, robust results, and it was, in fact, the first to formally quantify and compare the performance with the very few other algorithms that exist in this field. More concretely, the algorithm of [5] had an accuracy of 64% in contrast to the modest 48% and 38% achieved by Google+ and Facebook, respectively, on a self-collected dataset.

2.2. Machine learning-based methods

It is only recently that ML algorithms have been introduced for finding the representative image on a Web page. These innovative algorithms were first applied in the field of image selection based on user queries. The papers by Tong and Chang [37], Zhang et al. [43], for example, make use of Support Vector Machines (SVM) in order to classify images as informative based on textual queries from users. A few years later, research has been conducted on identifying representative images from a set of images by Kennedy and Naaman [16] using SVM techniques again, although this is still based on textual queries. SVM algorithms have proven to be highly effective for other meta-tasks too, such as prediction in the context of social networks as shown by Nie et al. [27], Song et al. [33]. To our knowledge, ML algorithms have not yet been applied to identify a representative image from a Web page. It is our working hypothesis that the application of ML algorithms could prove superior to rule-based, expert counterparts such as [5] for finding the representative image on a Web page. It is to be expected that the weighting given by ML algorithms will yield stronger results, since it will possibly discover patterns that are not obvious at first sight when creating a rule-based algorithm. That is, ML algorithms have the ability to learn functions of an arbitrary form, of which some may be highly intricate and out-of-reach for rule-based methods. Next to the expected increase in performance, ML algorithms also allow us to employ features without having to decide upon their performance beforehand. This is not possible when using rule-based methods, since these require us to decide on their significance a priori.

2.3. Semantic Web-based methods

Despite the fact that the problem of selecting a representative image from a Web page has not been adequately addressed yet, there have been recent developments that attempt to make the creators of websites themselves define the representative image. These developments attempt to promote common data formats and exchange protocols in order to bring structure to meaningful content of Web pages. The concept is easy to state, but inherently complex to realise. One of the most widespread developments is the Open Graph protocol (OGP) created by Facebook in 2010, which advocates the development of Web pages as graph objects, building on the ideas of the Resource Description Framework in Attributes (RDFa). This semantic approach is currently being used by influential organisations, such as Google, Facebook, and LinkedIn, and makes use of meta-tags to allow for better indexing of Web content. This is done in a straightforward manner by adding the `<meta property=''og:image'' content=''image_url''/>` meta-tag to the HTML file. Next to this, the competitors of Facebook have started a similar initiative which annotates Web pages with so-called Schema.org microdata. This framework is established from a collaboration between Google, Bing, Yandex, and Yahoo!, and is also built on the ideas of RDFa. More concretely, Web developers can add the `representativeOfPage` property to images, which are presented as objects in accordance with RDFa. This property requires a Boolean value and is thus informative of what Web developers view as the representative image of a Web page.

Although the added benefit of these initiatives is that the creators of the Web page are enabled to express their preference, the downside is that the representative image as chosen by the creators may actually not be representative. The problem here is that these meta-tags are designed for images that are to be presented as thumbnails, due to which the chosen images are often small and of insufficient quality. Next to this, not all Web developers are aware of this protocol, making it an unreliable tool to use by itself at this stage. This is most likely the reason why the algorithms of Google+ and Facebook do not perform adequately, as shown by Gali et al. [5]. Their algorithms attempt to make use of the protocols they have developed, but many websites have most likely not implemented these yet. The algorithm of Google+, for example, bases its choice of representative image on only size and aspect ratio, if a given website does not make use of OGP, Schema.org microdata, or ad-hoc meta-tags.

The reason commercial algorithms fail to accurately select the representative image is thus twofold. Firstly, these algorithms rely too much on initiatives such as OGP and Schema.org microdata, while they are not as widespread as one might expect. Secondly, the algorithms rely on very simple approaches, based on only features such as size and aspect ratio. It is the combination of these issues that most likely contributes to their inadequate performance. Despite the shortcomings of simple Boolean meta-tags defining images as representative or not, more informative meta-tags including photo, time, and location meta-data have been shown to enhance the retrieval of representative images on popular media-sharing websites such as Flickr [17]. This shows that the use of meta-tags is certainly of interest for the image retrieval community.

3. Methodology

We proceed by discussing the methodological framework. Overall, we build upon existing work and adhere to an idea similar to that in [5]. We approach the problem by three steps leading from a Web page to a selected represented image. Namely, extracting, filtering and classifying images. However, there are certain aspects in which we differ from [5], because we believe these could be improved upon. Essentially, the difference between our approach and that of [5] is threefold and pertains to the steps where we filter and classify the images. Firstly, we do not specify the classification scheme a priori, but allow the ML algorithm to determine the classification scheme itself using statistical analysis. We believe that the added benefit of such a statistical approach is its ability to identify complex schemes which might not be apparent at first sight. Secondly, we classify the images based not only on the features introduced in [5], but introduce several novel image features and employ meta-data protocols that we believe have predictive power. Thirdly, we believe that the categorisation that [5] apply in their framework to filter their images might be too restrictive. Therefore, we evaluate whether less restrictive filters yield better performance. For example, by selecting the representative image not only from the highest priority category, but also from the second-highest priority category. It is our working hypothesis that these modifications yield an algorithm with performance measures that are superior to that of the algorithm from [5].

3.1. Selection of image features

In order to fully understand the dynamics of what characteristics a representative image has, we collect data regarding both the representative images and all other non-representative images present on Web pages. We do so since it is not only valuable to understand what characteristics make an image representative, but equally important is to understand what characteristics are likely to make images less representative. The more interesting question now is *what* data we should collect. In order to apply a theoretically sound ML algorithm, we must carefully consider which image features to use as variables, as they are decisive for the quality of our algorithm. For the purpose of structure, we organise the image features in three overarching types. Firstly, we define *technical* features as those that relate to the composition of the image. Secondly, we define *structural* features as those that relate to the structural properties of the image in the HTML document. Third and lastly, we define *lexical* features as those associated with the textual contents related to the image. We now go over all image features type by type.

Technical. Inspired by the features as selected by Gali et al. [5], we believe that both the image file format and the image dimensions are informative. Firstly, we consider the Joint Photographic Experts Group (`jpg`), Graphics Interchange (`gif`), Portable Network Graphics (`png`), and the Scalable Vector Graphics (`svg`) image formats as a categorical feature. We differentiate between these formats since they often serve different purposes. For example, the `gif` format is very often used for low-resolution images and animations, while the `jpg` format is often used for real-life photographs [25]. Indeed, it could, for example, very well be the case that representative images are often real-life photographs, meaning that they are more likely to be of the `jpg` format. Secondly, the image dimensions are informative, since both the size and shape of the images play a role in representativeness as found by Helfman and Hollan [8]. It is, for example, likely that larger images are of greater importance on a Web page, while smaller images are often images of icons. Next to this, images with a very large aspect ratio have an irregular shape, which could indicate that they are formatting images or banners.

Besides, Gali et al. [5] also classifies images into the following five categories: *Representative, Logo, Banner, Advertisement*, and *Formatting and Icons*. This categorisation has been applied using both textual comparisons and considerations of the aspect ratio, since images that are either very wide or long are often banners or formatting images. However, we certainly believe that other intrinsic visual features of an image are also informative. Therefore, we introduce two additional intrinsic visual features of images, namely the colourfulness and colour coherence of an image. Firstly, we believe that images with a larger variety of colours are often photographs, which in turn are often of greater importance on Web pages [8]. For this purpose, we make use of the RGB colour model and construct this feature as the number of distinct colour-values p_{ij} as given in Eq. (1), where p_{ij} denotes the colour value of the ith colour component of the jth image pixel.

$$Colour\,fulness = |\{p_{ij}\}| \tag{1}$$

Secondly, what has been found is that images with more consistent colours are of greater importance for content-based image retrieval, since they would be more visually appealing [22]. For this purpose, we introduce an image feature that measures the variation of colours in each of the R, G, and B components of the RGB colour model. We define the colour coherence as the sum of the standard deviations of the colours in these components, as presented in Eq. (2) [36].

$$\textit{Colour coherence} = \sum_{i \in \{R,G,B\}} \left(\sqrt{\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - \mu_i)^2} \right) \tag{2}$$

Where N denotes the total number of image pixels and the mean μ_i is defined in Eq. (3).

$$\mu_i = \frac{1}{N} \sum_{j=1}^{N} p_{ij} \tag{3}$$

Structural. We found that [5] used a very limited number of structural image features, which relate to the way the images have been incorporated into the Web page, as they only consider whether an image is nested in the sub-tree of the <h1> or <h2> elements, or not. We strongly believe that there are additional novel structural properties to be exploited for the successful selection of representative images, which we will introduce below.

Let us start with the feature that we believe is the most interesting, namely the duration an image has been presented on the Web page. As far as we know, this statistic has not been exploited in prior literature, and we believe that this feature of the temporal dimension is highly informative. It is likely that the most important images of a given Web page have been presented on the Web page for much longer than those of lesser importance. We have been able to acquire these data by making use of a digital Web page archive that downloads and maintains Web caches of Web pages many times per year. More specifically, we have programmatically crawled through the Wayback Machine [40] digital Web page archive for a maximum of 10 years and found that the oldest images have been presented for 6 years on their corresponding Web pages. This temporal feature is defined as the number of years an image has been presented on the Web page. As far as we know, cached Web pages have not been used for retrieving representative images.

Next to this, the way the image is stored and located structurally on the Web page is informative. Therefore, we introduce the following novel image features. Firstly, the image order and distance from the root may indicate the relative importance of an image with respect to the other images present. These are constructed as the rank of the image in the ordered list of <img> tags and the depth at which the <img> tag of the image is nested in the HTML document respectively. If an image is, for example, placed first in the HTML document, while there are over hundred images present, then this might signal it is of relatively greater importance than the other images. The same holds for the distance from the root element of the HTML document, which indicates how deeply the image is nested in the HTML document. The deeper it is nested, the less important we believe it is. Secondly, the way an image is stored is also of interest. An image may either be stored on an external server, such as a commercial image hosting server, or it may be stored on the Web page itself. It is our working hypothesis that externally stored images are of lesser importance. Moreover, images that *are* stored on the Web page itself, can be ranked in importance once more, namely in terms of the folder depth at which the image is stored. If an image is stored in a very specific, deeply nested folder, then we believe it may be of lesser importance. More concretely, we define the folder depth of an image based on the image URL. For example, the URL www.website.com/folder/image.jpg implies that the image is stored at a folder depth of two.

Furthermore, prior research in retrieving the representative image has not used image meta-data, such as the OGP protocol and Twitter Cards. Given the fact that these meta-tags allow Web developers to define the representative image themselves (as explained in Section 2.3), we define additional boolean image features for each of these, indicating whether or not an image has been specified as representative by the protocol.

There are a few additional novel features that we believe are also of interest to us. On the one hand, if an image is clickable (contains the <href> attribute) or contains clickable regions (contains the <ismap> or <hasmap> attribute), then it is likely that the image is, for example, a navigational button. On the other hand, the fact that an image contains a caption could imply it is relatively important, while the fact that it is contained in a large list of images may reduce its relative importance. We locate lists of images using the <ul> and <ol> tags, which stand for unordered- and ordered list, and define the size of the list as the number of <li> tags within.

Lastly, there are many other HTML tags and attributes that can be informative regarding the importance of an image, namely, the <p>, <figure>, <article>, <section>, <button>, <footer>, and <form> tags, and the <hspace>, <vspace>, <align>, and <border> attributes. An image that is nested in a paragraph, figure, article or section is likely to be of importance, while images nested in a button, footer, or form are not. Next to this, it is likely that only important images are aligned in a certain way, using, for example, the <hspace>, <vspace>, or <align> attributes, while the same statement holds for images with borders.

Lexical. In contrast to features of the structural type, Gali et al. [5] did employ many lexical features. More concretely, they evaluated if image-related textual content, as defined in the <alt>, <title>, <class>, and <id> attributes of the image, match with the textual content of the overarching <title> and <h1> tags of the HTML document. Here the <alt> and <title> are, for example, descriptive for the content of the image and are therefore indeed of interest. Next to this, the <src> attribute has also been used for a similar type of textual comparison, since the image URL may be informative regarding its content. This procedure is indeed a valid assessment of the degree to which the image is relevant to the content of the Web page.

Yet, we believe that there is more useful lexical information for recognising representative images. The first is very similar to the textual matching done in [5], as we believe that textual matching of the image related <alt> and <title> attributes is also possible with the content of the OGP, TwitterCard, and standard meta-tags. The content of these is supposed to provide a general description of the Web page, and is therefore of interest. Thus, we introduce an additional feature, indicating the proportion of words from the image <alt> and <title> attributes that match the content of the OGP, TwitterCard, and standard meta-tags. In this way, we exploit textual meta-data too, besides the structural image meta-data discussed prior.

Secondly, we believe that the text surrounding the image in the HTML document is also of interest. Specifically, we believe that images with actual textual content surrounding it are of greater importance. For the purpose of analysing whether an image is indeed surrounded by text, we use the concept of text-to-content ratio (TTCR), as defined in [24], and use all textual content contained within the second parent of the <img> tag of the analysed image. We choose the second parent, since the first parent often only contains structural HTML tags, while the textual content related to the image is located outside, as illustrated below.

```
<p>
<a href=''http://www.website.com ''>
<img alt=''information about image'' src=''/folder/image>
</a>
Surrounding textual content
</p>
```

Based on the textual content contained within the second parent of the images analysed, we define the TTCR as in Eq. (4) below [24].

$$TTCR = \frac{L_{text}}{L_{DOM}} \tag{4}$$

where L_{text} denotes the total number of characters as based on the textual content and L_{DOM} denotes the total number of characters as based on both the textual content and the HTML mark-up text. In the example given above, L_{text} is the total number of characters in ''Surrounding textual content'' whereas L_{DOM} is the total number of characters in the entire structure.

Next to the TTCR, we believe that a measure of readability is also informative, since we believe this is indicative of whether the surrounding text is indeed information that is presented to users of the Web page. For this purpose, we use the Automated Readability Index (ARI) as introduced in [32] and presented in Eq. (5).

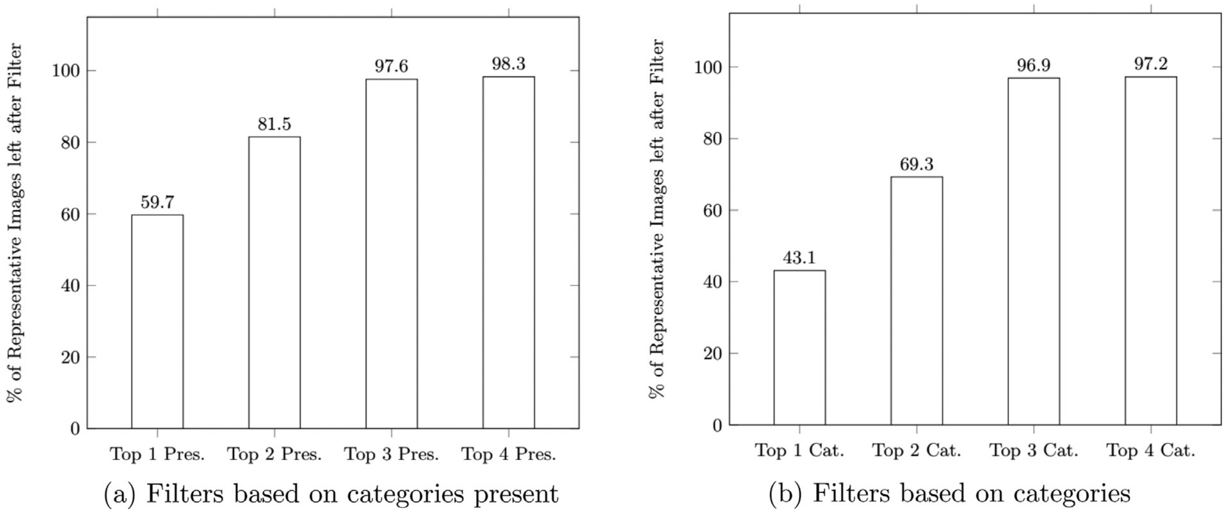
$$ARI = 4.71 * \left(\frac{characters}{words}\right) + 0.5 * \left(\frac{words}{sentences}\right) - 21.43 \tag{5}$$

Lastly, we believe certain additional features of the surrounding text can also be informative. Specifically, we believe that the proportion of surrounding capitalised text and the presence of typographical emphasisers in surrounding text, such as boldface and italics, are of interest too. Again, we only consider textual content nested in the second parent of the image under consideration. It is our working hypothesis that an image may also be of greater importance, if the proportion of surrounding text that is capitalised is high. The same holds for the use of typographical emphasisers, since they indicate textual importance. With regard to the former, we introduce a feature indicating the proportion of textual (non-HTML) content that is capitalised. For the latter, we introduce a feature indicating the presence of at least one of the following tags: <b>, <em>, <i>, <strong>, or <mark>.

4. Categorical filters

Now that we have elaborated on the image features, we proceed by discussing an idea by Gali et al. [5] that we wish to expand upon. We make use of the categorisation applied by Gali et al. [5], which stems from, firstly, width, height, and aspect ratio, and secondly, keywords in the image URL, as well as the class name of the <img> tag and of the parent element. More concretely, Gali et al. [5] only considers images from the highest priority category that contains any image, where the priority ordering is as follows, from highest to lowest: *Representative, Logo, Banner, Advertisement*, and *Formatting and Icons*. That is, Gali et al. [5] select images only from the highest category that contains at least one image.

We believe that this sort of categorisation to filter images might be too restrictive, since the categorisation might not be accurate. Next to this, Fig. 1a shows that, based on our data, representative images are, quite frequently, not located in the highest priority category containing an image. Here Top 2 Pres., for example, denotes the top two categories present and indicates the filtration where we only leave in images from the highest two categories containing at least one image. That is, if a Web page only contained images from the *Logo, Banner*, and *Advertisement* categories, then the Top 2 Pres. filtering would only filter images of the category *Logo* and *Banner*, since those are the top 2 categories present. As can be seen, this preprocessing step from [5], which only looks at the images from the Top 1 highest categories present and thus applies the Top 1 Pres. filter, filters only 59.7% of all representative images, dismissing 40.3% of the representative images. This is, therefore, indeed too restrictive, and we can see in Fig. 1a that filtering for the top 2, 3 and 4 highest priority categories containing an image, for example, can significantly decrease the number representative images that are filtered out.



(a) Filters based on categories present

(b) Filters based on categories

Fig. 1. Empirical results for different methods for filtering images.

Based on the previous considerations, we evaluate whether less restrictive filters yield better performance by, for example, selecting the representative image not only from the highest priority category, but also from the second highest priority category. Such a filtration is essentially a preprocessing step that is applied before selecting the representative image, reducing the area of search. More concretely, we attempt nine different filtration methods based on three ideas. The first idea is to apply no filter and allow all images to be selected. The second idea is to filter out only images from the top 1, 2, 3 and 4 categories *present* on a Web page, yielding four filtrations. We call these filtrations Top *n* Pres. filtrations, where $n \in \{1, 2, 3, 4\}$. Here we have to remember that the top category may very well not be the highest priority category, if there is simply no such image on the Web page. The Top *n* Pres. filter makes a ranking based on the images present on the Web page only. If there is only one image of the *Logo* category and one image of *Banner* category, then the Top 1 Pres. filtration would only select the image from *Logo* category, while the Top 2 Pres. filtration would select both. Lastly, the third idea is to filter based on the top 1, 2, 3 and 4 categories, independent of the category ranking on the Web page, yielding an additional four filtrations. These filtrations we call Top *n* Cat., where again $n \in \{1, 2, 3, 4\}$. In this case, the filter based on the top 1 category implies that you only look at images from the *Representative* category, the filter based on the top 2 category only looks at images from the *Representative* and *Logo* categories and so forth. Fig. 1a and b indicate that these less restrictive filters preserve many more representative images than that of [5]. The fact that the number of images preserved using the top 3 and top 4 filters is almost equivalent implies that very few Web pages contain images from more than three categories. We investigate which of these filters, all of which being novel variations of the filters as employed by Gali et al. [5], works most effectively as a preprocessing step.

5. SVM Framework

Since we are dealing with a binary classification problem, ML algorithms are specifically well-suited. We apply supervised learning algorithms, because we have labelled data at our disposal. We have chosen to apply the Support Vector Machine (SVM) framework since it is particularly suited to problems with a relatively large number of features [20].

5.1. Programming implementation

In order to apply the SVM algorithm, we made use of the widely employed LIBSVM library for SVM [3]. More specifically, we use the *C*-Support Vector Classification (*C*-SVM) algorithm as introduced in [39], which incorporates the penalty parameter *C*. We used the algorithm of [3], implemented in the Java programming language.

5.2. Data imbalance and data preparation

We are concerned by the limited number of representative images that we have available to use for training the SVM, namely, this might make the estimates of our weights imprecise since we have an unbalanced dataset where the number of non-representative images is much larger than the number of representative images. More concretely, we have 922 representative images, while we have 17,159 non-representative images, meaning that just over 5% of the images in our dataset is representative. The most obvious solution to resolve this problem of unbalanced data is to use oversampling on the representative images, in order to increase the number of representative images to be the same as the number of non-representative images. However, the problem here is that oversampling will actually not mitigate the problem, since we are not providing any additional useful information to the SVM classifier. That is, the separating hyperplane will not differ with or without oversampling, since the oversampled data points will lay at the exact coordinates of the respective initial data point. We approach this problem using a class-weighted SVM, where the misclassification penalty differs between classes [30]. In our case, this implies having different misclassification penalties C_r and C_n for the representative and non-representative image classes, respectively. We used the class-weighted SVM, which allows us to give a greater penalty to misclassifying a representative image, as compared to a non-representative image. We evaluate five multiples, namely, where C_r is 1x, 2x, 5x, 10x or 50x as large as C_n. We evaluate these and will select the multiple that yields the highest performance measure.

Next to this, it is important to realise that SVM requires the data to be represented as a vector in $\mathbb{R}^n$ where n denotes the number of variables. Therefore, we need to convert categorical attributes, such as image format, into numeric data. We apply a different transformation for categorical features, such as the number of years that an image has been presented on a given Web page. For this, we specify n variables in order to represent one categorical variable containing n categories. Each of these n variables is essentially an indicator function, stating whether an image is of that specific category or not. We do so since this approach has been empirically found to be more stable than just using a single categorical variable [9]. For example, for using the categorisation from [5] as image feature, we create five indicator variables one for every individual category respectively, rather than a single variable with elements ranging from 1 to 5.

5.3. Kernel selection

Lastly, we need to decide upon the kernel for the SVM. There are three well-known and frequently used kernels, namely the linear, polynomial, and the Gaussian radial basis function (RBF) kernel, and chose for the RBF kernel presented in Eq. (6) where $\gamma = \frac{1}{2\sigma^2} > 0$.

$$K_{RBF}(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{||\mathbf{x_i} - \mathbf{x_j}||^2}{2\sigma^2}\right) = \exp(-\gamma||\mathbf{x_i} - \mathbf{x_j}||^2) \tag{6}$$

We apply the RBF kernel, since it has been shown that under the right parameter configuration, there is no need to consider the linear kernel, because the performance of the RBF kernel will always be better [15]. Next to this, the linear kernel is only useful when there is a sufficiently large number of features which makes mapping into a different dimensional feature space redundant. We do not have a large enough number of features to warrant this. The RBF kernel is preferred since it is able to handle the non-linear relations that might prevail in our classification problem, as the data are most likely not linearly separable without the initial mapping into another dimensional space. The linear kernel is unable to capture this. Lastly, we do not consider the polynomial kernel since it has a large number of hyper-parameters, making it computationally intensive to solve [9]. Also, numerical difficulties tend to occur with the polynomial kernel when the degree d is large [9].

5.4. Backward feature selection

Due to the large number of features, it is of great importance to distinguish between relevant and irrelevant features. Even though all features have a theoretical grounding for being relevant (as explained in Section 3.1), the question remains whether they are indeed informative of the representativeness of an image in practice. Therefore, in order to narrow the focus of the SVM algorithm to the relevant features, we must provide the SVM algorithm a (sub)set of features that is truly relevant. For this purpose, we use the backward feature selection procedure as proposed in [19]. That is, we start by including all features proposed, and eliminate the feature of which the removal yields the highest increase, or lowest decrease, in the performance measure F_1, one at a time. We do so until there have been two subsequent rounds that have not led to an increase in F_1. We use this stopping condition, since it may be that the removal of one variable may reduce the F_1, but removing an additional variable would increase the F_1 sufficiently to mitigate the initial decrease. This procedure is of great importance for the performance of the classifier, since having irrelevant variables would most likely reduce the ability of the SVM to generalise.

6. Performance evaluation

Now that we have discussed the framework that underpins the method we use to select representative images, we proceed by elaborating on the choices we have made regarding performance evaluation. We start off by discussing our dataset and our method of parameter estimation for the SVM. Subsequently, we introduce the benchmark algorithms that we use to compare the performance of our algorithm with and present the evaluation metrics used to make the comparison with benchmark algorithms.

6.1. Ground truth dataset

The data that we use as our ground truth has been collected by the School of Computing at the University of Eastern Finland and consists of a list of Web pages, each page having at most three representative images selected by 117 voluntary contributors in September 2014 [5]. For the selection of Web pages to assess, the volunteers were allowed to either use a Web page of their own choice, or use a MOPSI search result [4]. In accordance with our research question, the contributors were instructed to select images that best *summarises* the content of the Web page, best representing the content of the Web page. That is, volunteers were instructed to perform the specific task of determining the image that best summarises the contents of the Web page [26], disregarding interestingness [7], memorability [13], and diverseness [12].

6.2. Parameter estimation

In order to apply the SVM algorithm, we are required to propose parameters C and $\gamma = \frac{1}{2\sigma^2}$. This can be seen in the C-SVM formulation [30] and the RBF kernel (as shown in Eq. (6)). Most importantly, these parameters need to be chosen based on training data, since none of the specifications of our algorithm are allowed to be based on data from the testing set. We have decided to divide the dataset of images into 80% and 20%, as training and testing data respectively. The most robust method for finding these parameters from the training set is k-fold cross-validation, since it limits frequent problems in ML such as overfitting. Therefore, we apply cross-validation on the training data for evaluating the performance of parameter configuration (C, γ), as proposed by Hsu et al. [9]. In order to search for a well-tuned parameter configuration, we apply a two-step grid search. Before we discuss this search framework, we would like to clarify the method we use to compare parameter configurations, and our approach to cross-validation.

For the purpose of comparing parameter configurations, we define the cardinal ordering of parameter configurations as based on two configurations i and j, shown in Eqs. (7)–(9). The ordering is determined by the F_1 performance measure.

$$(C_i, \gamma_i) \prec (C_j, \gamma_j) \Leftrightarrow F_{1,i} < F_{1,j} \tag{7}$$

$$(C_i, \gamma_i) \succ (C_j, \gamma_j) \Leftrightarrow F_{1,i} > F_{1,j} \tag{8}$$

$$(C_i, \gamma_i) \sim (C_j, \gamma_j) \Leftrightarrow F_{1,i} = F_{1,j} \tag{9}$$

With respect to the cross-validation, we have decided to stipulate $k = 5$, since larger values would lead to problems regarding computation time. That is, we divide the training dataset of images into 80% and 20%, for training and validation respectively, and vary the split five times with a specific parameter configuration (C, γ). We use the average F_1 as performance measure based on which parameter configurations are compared. It is important to ensure that the proportion of representative images is approximately equal in the training and validation data. We do so by computing the proportion of representative images in the complete training dataset at first, and afterwards ensuring that the representative images appear in the same proportions in the training and validation subsets. After identifying the optimal parameters using cross-validation, we use these to train the SVM using the whole training dataset.

6.2.1. Two-step grid search

We discuss the two-step grid search framework towards parameter estimation. In the first step of this framework, we apply a grid-search for C and γ on the sets $\{10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4\}$ and $\{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$, respectively. After we have obtained the best parameter combinations (C_1, γ_1) from the first step, we apply another search to a finer grid around (C_1, γ_1). More specifically, this finer grid for both C_1 and γ_1 consists of five steps up and down with step-size 0.0025, 0.025, 0.25, 2, 10, 100 and 1000 for 10^{-2}, 10^{-1}, 10^{-0}, 10^1, 10^2, 10^3 and 10^4 respectively, based on the parameter we obtain in the first step. It is important to note that we apply this second step in order to increase our chances of arriving at a local optimum for the parameter configuration. We resort to such a local optimum, because obtaining a global optimum for the parameter configuration is computationally difficult. In order to find this local optimum, we vary the step-size of these sets so as to account for the magnitude of the parameters. This finer grid allows us to locate the final parameters (C_2, γ_2) from the second step. For example, if we find the parameters ($C_1 = 1000, \gamma_1 = 10$) in the first step, then we search for an improved C_2 and γ_2 in sets $\{500, 600, \ldots, 1400, 1500\}$ and $\{0, 2, \ldots, 18, 20\}$, respectively. Overall, the two-step nature of our algorithm helps us to achieve tuned parameters for the RBF kernel, while the 5-fold cross-validation prevents SVM from overfitting.

6.3. Benchmark algorithms

The performance of our algorithm is compared with the methods from [5], Facebook, and Google+. The performance of these has been given in [5]. However, in order to make a fair comparison, we reimplement the algorithms once more based on our new dataset.

Next to reimplementing the basic algorithm from [5], we also suggest two slight modifications to this algorithm, which we expect to perform better. Firstly, Gali et al. [5] match keywords from, for example, the image title attribute with the Web page URL, and increment the score of an image if they match. However, a problem we recognised was that often one of the two was capitalised, while the other was not. Images for which this is the case would thus not have the increment in score they should have, since we believe it is the content of the text that matters for this purpose and not the layout. Therefore, we suggest also implementing the algorithm by Gali et al. [5] with the additional feature that the matching is case-insensitive, simply by casting all textual features to the lower case. If these image features, which are based on matching textual characteristics, are indeed informative, then we expect the performance to increase, since we would be able to identify textual matches more accurately. Secondly, we also have concerns regarding the fact that [5] choose the representative image *only* from the category with the highest priority that contains any image. That is, their algorithm disregards images from four out of the five categories defined in [5]. We believe that the assumption that the representative image lies in the highest category is too restrictive, since it could very well be that the categorisation is inaccurate (as explained in Section 4). Therefore, we suggest inspecting whether a category-insensitive version of the algorithm by Gali et al. [5] is better in terms of performance, and use that as a benchmark, too.

Lastly, next to these well-developed algorithms, we also use the performance of three greedy algorithms as benchmarks. The first greedy algorithm is one in which the largest image is always chosen as representative. We expect this approach to perform reasonably since important images are often large in size. The second greedy algorithm is one in which the first image that appears in the HTML document is selected. The reason for choosing this is that we expect Web developers to place crucial images at the very start of the HTML document. Lastly, the third greedy algorithm combines the categorisation from [5] by greedily selecting the largest image from the highest category present on the Web page.

6.4. Evaluation metrics

In order to evaluate the performance of our algorithms, we have narrowed down the performance measures to four metrics which are particularly well-suited to evaluating binary classifiers: accuracy, precision, recall and F_1. Comparing performance using the harmonic mean F_1 has our preference over merely precision or recall, since we believe that the performance must incorporate both the number of correctly identified representative images, and how many truly representative images the algorithm is able to identify. If, for example, the algorithm were to correctly label one representative image out of 100 images whilst labelling the rest of the images as non-representative, then the precision would be 100%, although this is most certainly not a reliable indicator of performance as we miss 99% of the representative images. We want both precision and recall to be high, which is precisely what the F_1 metric measures. Therefore, we use the F_1 metric to evaluate the performance of the algorithms. We also present the performance measure used by Gali et al. [5], which we call the WP Accuracy (from Web Page Accuracy). This is an accuracy measure indicating the proportion of Web pages for which the algorithm selects a ground truth image as representative.

7. Results

Now that we have discussed the methodological framework that we use to determine representative images, we proceed by presenting the results that we have obtained. We start this section by discussing the results from the benchmark algorithms in Section 6.3, including the improvements made on the framework from [5], after which we conclude this section by presenting the results of our methodology using SVM in Section 7.2. We have implemented all the algorithms in the Java programming language and ran our experiments on a PC with an Intel Core i7-3520M processor at 2.90 GHz, with 8GB of RAM. We used the dataset provided by Gali et al. [5] from 2015, with minor adaptions made for missing Web pages and images. The data we use comprises of 1162 Web pages and 18,081 images, from which 922 are labelled as representative.

7.1. Benchmark algorithms

Let us start by presenting the results from the benchmark algorithms in Table 1, as introduced in Subsection 6.3. We call the algorithm by Gali et al. [5] WebIma (from Web image), in accordance with their own nomenclature. The runtime of all algorithms presented in this table is only a couple of seconds and, thus, negligible.

The most surprising observation is that, contrary to our expectation, the algorithms used by Facebook and Google+ perform relatively poor in terms of image selection. In fact, the results even indicate that they perform worse than the three greedy heuristic algorithms. Out of these three, we find the algorithm that simply selects the first image from the HTML document to perform the best in terms of all five performance measures. As can be seen, the performance of this algorithm even comes very close to that of WebIma, since all the performance measures differ by at most 1 percentage point between

Table 1
A comparison of the performance of algorithms, rounded to one decimal.

Algorithm	Acc. (%)	Prec. (%)	Rec. (%)	F_1 (%)	WP Acc. (%)
First Image	92.0	25.1	28.6	26.7	51.9
Largest Image	91.7	22.7	25.9	24.2	47.0
Largest Image in Highest Category	91.8	23.7	27.0	25.2	48.9
Facebook	**93.7**	**30.1**	17.9	22.4	32.4
Google+	91.9	22.1	23.6	22.8	42.8
WebIma [5]	92.1	25.6	29.2	27.2	52.9
Case-insensitive WebIma	92.2	26.7	30.5	28.5	55.2
Category-insensitive WebIma	92.3	27.4	**31.2**	**29.2**	**56.6**

Table 2
SVM results using only the features from WebIma, rounded to one decimal.

Filter	Acc. (%)	Prec. (%)	Rec. (%)	F_1 (%)	WP Acc. (%)	Runtime (hr)
No filter	**95.2 (2.1)**	66.7 (2.3)	15.1 (2.3)	24.6 (2.4)	16.4 (1.9)	25.4
Top 1 Cat.	94.1 (2.0)	45.8 (2.1)	13.8 (2.1)	21.2 (1.9)	15.5 (1.9)	**20.9**
Top 2 Cat.	93.3 (1.5)	59.6 (1.8)	22.0 (1.7)	32.2 (2.0)	23.7 (1.7)	21.2
Top 3 Cat.	94.0 (1.9)	67.2 (1.9)	23.2 (1.9)	34.5 (1.5)	25.9 (1.9)	21.1
Top 4 Cat.	93.5 (2.3)	59.2 (1.5)	16.1 (1.5)	25.3 (2.0)	19.0 (1.5)	23.7
Top 1 Pres.	92.6 (2.1)	47.9 (2.2)	20.9 (1.6)	29.1 (2.2)	22.8 (1.7)	21.0
Top 2 Pres.	93.4 (1.8)	**70.2 (2.1)**	**26.5 (1.9)**	**38.5 (2.1)**	**29.9 (1.8)**	23.1
Top 3 Pres.	93.6 (2.0)	56.5 (2.5)	21.7 (1.5)	31.3 (2.2)	23.9 (2.3)	22.9
Top 4 Pres.	94.2 (1.7)	65.5 (2.1)	19.8 (2.2)	30.4 (1.9)	21.8 (1.7)	24.6

the two. Next to this, we can also see that the improvements that we have suggested on WebIma are indeed beneficial with respect to performance. More concretely, we find that the algorithm that results in the highest F_1 score, which is the score we deem most important, as discussed in Subsection 6.4, is obtained when applying the WebIma algorithm without categorisation as a preprocessing step (whilst also ensuring that textual matching is case-insensitive). It appears that the filter which was applied in WebIma is indeed too restrictive for their own algorithm, as we expected, and thus counter-productive. The WebIma algorithm performs better without the use of the category-based filter.

7.2. SVM Algorithms

Now that we have shown the results from the benchmark algorithms, we proceed by presenting the results from our SVM framework. We start with Subsection 7.2.1 by discussing the results of the SVM framework with only the features introduced in [5]. Subsequently, we discuss the results of the SVM framework with the addition of the new image features in Subsection 7.2.2. In all the tables that follow, the metrics presented are the average as found by the 5-fold cross validation on the whole dataset, and we have included the standard deviations in brackets.

7.2.1. SVM With features from [5] only

Firstly, let us look at the performance of the SVM framework when we use the exact same features as that in WebIma, excluding the new features introduced in Section 3.1. Table 2 shows the performance of the SVM framework on all nine preprocessing filters introduced in Section 4.

Overall, we find that with respect to all performance measures except accuracy, the preprocessing filter where we pick the top two categories *present* performs the best. We find that this outperforms WebIma by 11.3 percentage points in terms of the F_1 score, with an F_1 measure of 38.5%. Lastly, it is noticeable that often the less restrictive filters have a larger runtime compared to more restrictive filters. This phenomenon appears to persist through all the results that follow, and can be explained by the fact that less restrictive filters provide the SVM with more training data, which lengthens the process of learning.

7.2.2. SVM with new features

Lastly, we present the results of the SVM algorithm framework with the addition of our novel features as introduced in Section 3.1. After applying backward feature selection, we find that the features presented in Table 3 are most informative for finding the representative image.

Firstly, we observe that all but one of the features introduced in WebIma are informative for the representativeness of an image. That is, only the feature that regards textual matching of keywords in the image URL (src) with the <title> and <h1> tags of the Web page is not regarded as relevant. A possible explanation for this is the lack of informative keywords in an image URL. More concretely, an image URL often mainly comprises rather random sequences of letters and numbers in order to make the URL unique. Therefore, it might be the case that keywords from an image URL may not be as informative as [5] had expected. Secondly, it is interesting to note that many of our novel image features of the structural type appear to be informative. Gali et al. [5] introduced almost no features of the structural type, while it seems that these features are of interest as we had expected.

Using the features presented in Table 3, we proceed by presenting the results of the SVM framework using the two-step grid search approach. In the results that follow, we have made use of the non-normalised data, since that yielded superior performance measures in all cases. We start by presenting the results of the two-step grid search *without* using class weightings in Table 4.

The best parameter configuration is found using a filter that only considers images of the highest category present on the Web page. We find that that the Top 1 Pres. filter is able to achieve an F_1 performance measure of 40%.

Lastly, we present the results when using a class-weighted SVM in Table 5, as introduced in Subsection 5.2, using the two-step grid search. We have found that a class-weighted SVM where the cost parameter for representative images (C_r) is 10x as large as the cost parameter for non-representative images (C_n) yields the highest F_1 performance measure. Therefore,

Table 3
An overview of the features obtained after backward feature selection, with new features compared to WebIma in italics.

Type	Feature	Explanation
Technical	Image Format	The format of the image (e.g., jpg)
	Width	The width of the image in px
	Height	The height of the image in px
	Size	The product of the width and height in px
	Aspect Ratio	Determines the shape of the image
	Image Category	What the image category is in [5]
	Colourfulness	How colourful an image is
Structural	Subtree of <h1> or <h2>	Whether the image is the subtree of <h1> or <h2>
	Duration Online	How long the image has been presented on the Web page
	Image Rank	The rank of the image as it appears in the HTML
	External Server	Whether the image is hosted on an external server
	Clickable	Whether the image has the <href> attribute
	Article	Whether the Image has a <article> tag as parent
	Button	Whether the Image has a <button> tag as parent
	Footer	Whether the Image has a <footer> tag as parent
	Form	Whether the Image has a <form> tag as parent
Lexical	Alt	Textual content of the image its <alt> attribute
	Image Title	Textual content of the image its <title> attribute
	Class	Textual content of the image its <class> attribute
	Parent Class	Textual content of the <class> attribute of an image its parent
	ID	Textual content of the <id> attribute of an image
	Parent ID	Textual content of the <id> attribute of an image its parent
	H1	Textual content of the <h1> tag
	Title	Textual content of the <title> tag
	Textual Importance	Whether the surrounding text contains emphasisers

Table 4
SVM results of the two-step grid search using the features in Table 3, rounded to one decimal.

Filter	Acc. (%)	Prec. (%)	Rec. (%)	F_1 (%)	WP Acc. (%)	Runtime (hr)
No filter	**94.3 (1.9)**	58.7 (2.1)	24.4 (2.2)	34.5 (2.2)	28.6 (2.5)	24.9
Top 1 Cat.	90.6 (1.7)	32.7 (2.6)	23.7 (3.0)	27.5 (2.5)	27.3 (2.8)	21.7
Top 2 Cat.	90.8 (1.8)	50.0 (2.7)	29.8 (2.3)	37.4 (2.4)	33.0 (2.1)	22.1
Top 3 Cat.	89.2 (2.0)	34.5 (1.9)	28.0 (2.3)	30.9 (2.6)	32.0 (2.1)	21.9
Top 4 Cat.	91.3 (1.7)	47.0 (2.6)	**31.2 (2.5)**	37.5 (2.2)	**34.6 (2.5)**	23.4
Top 1 Pres.	90.2 (1.9)	**66.0 (2.5)**	28.7 (1.7)	**40.0 (2.0)**	31.0 (2.2)	**19.8**
Top 2 Pres.	90.5 (2.0)	51.3 (2.5)	26.9 (2.1)	35.3 (2.2)	30.0 (2.1)	20.3
Top 3 Pres.	91.9 (2.1)	53.3 (2.9)	27.8 (2.0)	36.6 (2.4)	32.9 (2.0)	23.9
Top 4 Pres.	91.3 (2.3)	42.0 (2.4)	28.2 (2.4)	33.8 (1.9)	31.9 (1.1)	23.5

Table 5
Class-weighted SVM results of the two-step grid search using the features in Table 3, rounded to one decimal.

Filter	Acc. (%)	Prec. (%)	Rec. (%)	F_1 (%)	WP Acc. (%)	Runtime (hr)
No filter	**93.2 (1.2)**	55.3 (2.2)	28.5 (2.5)	37.6 (1.5)	31.2 (1.4)	24.7
Top 1 Cat.	91.3 (1.8)	36.9 (2.5)	24.9 (2.3)	29.7 (1.2)	29.2 (1.1)	20.6
Top 2 Cat.	90.1 (1.5)	58.2 (1.3)	31.7 (3.2)	41.5 (2.3)	33.9 (2.6)	**20.3**
Top 3 Cat.	91.6 (2.0)	31.3 (2.3)	29.4 (1.3)	30.3 (1.5)	33.2 (1.9)	22.9
Top 4 Cat.	92.4 (2.2)	45.9 (1.4)	**35.5 (2.1)**	40.0 (2.4)	**36.2 (1.9)**	24.1
Top 1 Pres.	90.8 (2.1)	**71.1 (1.5)**	32.4 (1.9)	**43.9 (1.3)**	32.2 (1.3)	21.2
Top 2 Pres.	91.3 (1.3)	52.4 (2.1)	32.1 (1.5)	39.8 (2.4)	31.9 (2.1)	22.1
Top 3 Pres.	90.4 (1.9)	52.9 (2.3)	31.4 (2.7)	39.4 (2.1)	32.7 (1.9)	23.7
Top 4 Pres.	92.2 (1.5)	44.2 (1.8)	30.9 (1.5)	36.4 (2.0)	31.2 (2.2)	23.8

all results that follow are based on a class-weighted SVM where $C_r = 10C_n$. We find that, as expected, the use of a class-weighted SVM is able to improve the performance of our algorithm even further by resolving the imbalance in our dataset. Once again, we find that the Top 1 Pres. filter yields the best performance. Hence, it appears that the restrictive filter in WebIma does work well within the SVM framework. More specifically, we find that the Top 1 Pres. filter combined with a parameter configuration as found with the two-step grid search yields the highest F_1 performance measure of no less than 43.9%.

In order to assess the statistical significance of the difference in performance in terms of F_1, we make use of a two-sample one-tailed t-test. We compare the performance of our SVM algorithms and present the results of our statistical

Table 6
Statistical comparison of the difference in F_1 performance between the SVM algorithms using a two-sample one-tailed t-test, rounded to three decimals. Difference in performances with *, ** and *** are statistically significant at $p < 0.05$, $p < 0.01$ and $p < 0.001$, respectively.

SVM comparison	WebIma features only	With new features	Class-weighted
WebIma features only	0.000	1.156	4.890**
With new features	−1.156	0.000	3.656**
Class-weighted	−4.890**	−3.656**	0.000

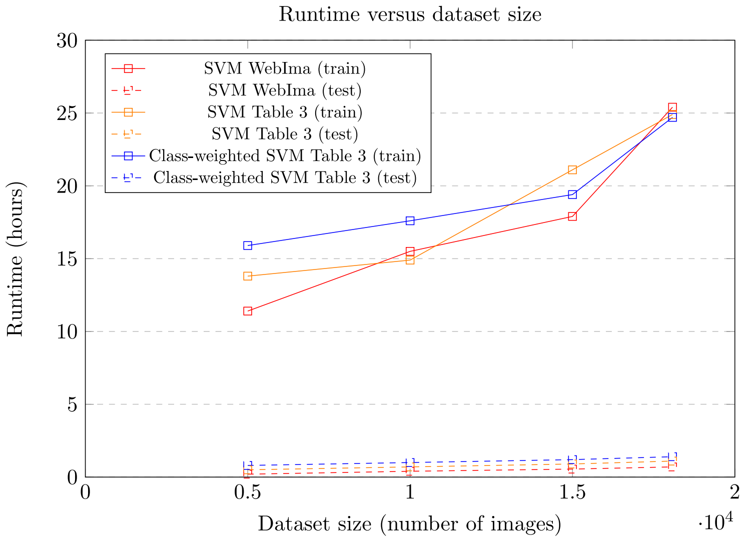


Fig. 2. Analysis of runtime versus dataset size. For readibility, test runtimes are offset by 0.5 hours.

comparison in Table 6, with H_0: $\mu_1 \le \mu_2$ and H_a: $\mu_1 > \mu_2$. We see that particularly the class-weighting of the SVM is able to significantly improve the F_1 performance.

To conclude, we find that using the image features as presented in Table 3, we consistently find that the Top 1 Pres. filter is able to yield the best performing algorithm. We find that a class-weighted SVM works effectively to remove the imbalance between representative- and non-representative images in our dataset, which enables our algorithm to achieve an F_1 performance measure of no less than 43.9%.

7.3. Time complexity

In order to evaluate the scalability of our approach, we briefly discuss the time complexity of our algorithm. This is of interest given the fact that our analysis is done in a low-scale environment, with a controlled number of train- and test instances. We focus on practical estimates of runtime against dataset size. For a theoretical overview of the time complexity of SVM in the LibSVM library [3], we refer to [1]. In order to assess the runtime for smaller datasets, we use stratified sampling, grouped based on images being representative or not as based on our ground truth dataset [34]. Fig. 2 presents the runtime plotted against the dataset size for SVM with only features from WebIma, SVM using features in Table 3, and class-weighted SVM using features in Table 3. We do so for both the training- and testing phase.

As we can see, the training phase of the algorithm is the largest contributor to runtime. This can be attributed to the computationally intense two-step grid search paired with the $\mathcal{O}(n^3)$ time complexity of training the SVM, where n denotes the number of images [38]. Testing is very fast as it is based on one function application to each test instance, which yields an $\mathcal{O}(n)$ time complexity for testing the SVM.

7.4. Failure analysis

Now that we have found an algorithm that outperforms WebIma by 16.7 percentage points, it is interesting to analyse its shortcomings. For this purpose, we briefly perform a failure analysis in order to understand what representative images our algorithm misclassifies as non-representative (false negatives), and what non-representative images our algorithm misclassifies as representative (false positives). This is of interest for future work, since it reveals image features that our algorithm is not able to handle well. We find three features to be skewed for the misclassified images.

Firstly, we find that out of all false negatives, 80.6% have been online for a duration of less than 3 years. This indicates that, most likely, images that have been presented on the Web page for longer are given greater weight, due to which images with a shorter duration online are more likely to be classified as non-representative. With regard to the false positives, this image feature is not skewed towards either a long or a short duration online.

Secondly, it appears that 62.6% of the false negatives images have an image rank of less than 5. That is, many images that are falsely classified as non-representative appear relatively early in the HTML document. This is surprising, since we would expect images at the top of the page to be more likely to be deemed representative. Possibly, this image feature has not been employed sufficiently due to, for example, overfitting. We find that the false positive images do not reveal this characteristic, namely, only 17% of those have a image rank of less than 5.

Lastly, we do, however, find that the false positives are skewed towards images of the `jpg` format. More concretely, 94.9% of the false positive images are of the `jpg` format, while this is only 60% for the false negative images. This could imply that images of the `jpg` format are given relatively greater weight compared to the other image formats (`png`, `svg`, `gif`), while it appears that not all images of the `jpg` format are representative.

8. Conclusion

Given the fast-paced environment that we live in, images are highly useful for enabling Web users to quickly understand the content of a Web page. Determining the representative image, which best summarizes the content of a Web page, is of great importance for many applications, such as Facebook and Google+. In this paper we have improved upon the methodological framework from [5] for the selection of the most representative image of a Web page, which is the best performing known algorithm for this purpose up to today. We have done so using, amongst others, the SVM framework, various novel image features, and previously unused image meta-data protocols. Besides, we have discovered several novel image features which are shown to be relevant.

We find that the application of our SVM framework to solely the image features as defined in [5] increases the F_1 score by 11.3 percentage points, from 27.2% to 38.5%. After we employ our novel image features, exploit meta-data protocols, and use backward feature selection in order to simplify the model, we find that many image features of the structural type appear to be of interest, while features of this type have not been used by Gali et al. [5]. After using a two-step grid search for parameter estimation, we find that the F_1 performance measure rises to 40.0%. Lastly, after we employ the class-weighted SVM in order to resolve the fact that only 5% of our data comprises of representative images, we find that this final adjustment improves the classification performance of our algorithm even further to an F_1 score of 43.9%. This result implies that our algorithm outperforms the state-of-the-art [5] by no less than 16.7 percentage points.

Suggested beneficiaries of our algorithm would be the search engine and image retrieval community, including the commercial sector due to the superior performance of our algorithm relative to equivalents from industry. Besides, our research brings together several different approaches to performing image ranking on Web pages, which is of interest for the recommender systems community. Lastly, we believe that several novel image features that we have introduced are of interest to a wider research community focusing on image retrieval, with most notably our temporal image feature based on the duration an image has been presented on a given Web page.

We wish to discuss ideas that could inspire future work in the field of determining the representative image of a Web page. Firstly, a fundamental limitation of our algorithm concerns the computational complexity, which leads to lengthy runtime in the training phase. Alternative methods can be applied in the training phase, such as the Core Vector Machine (CVM) as introduced in [38], in order to speed up computation. Their approach yields a training phase time complexity which is linear in the number of instances.

Secondly, it might be interesting to broaden the scope and not only consider the images present on a Web page as the exhaustive set from which the representative can be chosen. If, for example, a given Web page contains no image that is deemed representative, it might be better to look for images on the Web that are related to the Web page, rather than selecting an unsatisfactory image from the Web page itself. Similarly, another extension to our research could target extracting the representative image from a website, comprising multiple Web pages, in contrast to extracting the representative image from a single Web page.

Thirdly, we could analyse the quality of the Web images using a multi-dimensional image quality prediction model [42], and propose as candidates only the images deemed of high quality. In addition, we could extract image semantic features [11] and match these against Web page text or genre [23] to further narrow down the candidates. For representing the text of a Web page one can make use of natural language processing techniques for normalization and summarization.

Lastly, given the high level of subjectivity in the task of determining the image that best summarises the content of a Web page, it may be of interest to use crowdsourcing approaches in order to both enhance the quality and increase the size of our ground truth dataset. An example of a crowdsourcing marketplace is Amazon Mechanical Turk, which has been shown to be applicable for learning preferences for visual summarisation of image collections [31].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Abdiansah, R. Wardoyo, Time complexity analysis of support vector machines (SVM) in LibSVM, Int. J. Comput. Appl. 128 (3) (2015) 28–34.

[2] G. Adam, C. Bouras, V. Poulopoulos, Image extraction from online text streams: a straightforward template independent approach without training, in: IEEE 24th Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA 2010), 2010, pp. 609–614.

[3] C.C. Chang, C.J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 1–27.

[4] P. Franti, J. Kuittinen, A. Tabarcea, L. Sakala, MOPSI Location-based search engine: concept, architecture and prototype, in: Proc. 25th ACM Symp. on Applied Computing (SAC 2010), ACM, 2010, pp. 872–873.

[5] N. Gali, A. Tabarcea, P. Franti, Extracting representative image from web page, in: Proc. 11th Int. Conf. on Web Information Systems and Technologies (WEBIST 2015), 2015, pp. 411–419.

[6] S. Gupta, G. Kaiser, D. Neistadt, P. Grimm, DOM-based content extraction of HTML documents, in: Proc. 12th Int. ACM Conf. on World Wide Web (WWW 2003), 2003, pp. 207–214.

[7] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, L. van Gool, The interestingness of images, in: IEEE Int. Conf. on Computer Vision (ICCV 2013), IEEE Computer Society, 2013, pp. 1633–1640.

[8] J. Helfman, J. Hollan, Image representations for accessing and organizing web information, in: Proc. Int. Soc. for Optical Engineering (SPIE 2000), Internet Imaging II, 4311, 2000, pp. 91–101.

[9] C.W. Hsu, C.C. Chang, C.J. Lin, A Practical Guide to Support Vector Classification, Technical Report, National Taiwan University, 2003.

[10] J. Hu, A. Bagga, Functionality-based web image categorization, in: Proc. 12th Int. Conf. on World Wide Web (WWW 2003), 2003.

[11] C. Huang, H. Xu, L. Xie, J. Zhu, C. Xu, Y. Tang, Large-scale semantic web image retrieval using bimodal deep learning techniques, Inf. Sci. 430 (2018) 331–348.

[12] B. Ionescu, Working Notes of the 2016 MediaEval Retrieving Diverse Social Images Task, http://www.multimediaeval.org/mediaeval2016/diverseimages/, 2016.

[13] P. Isola, J. Xiao, D. Parikh, A. Torralba, A. Oliva, What makes a photograph memorable? IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1469–1482.

[14] P.M. Joshi, S. Liu, Web document text and images extraction using DOM analysis and natural language processing, in: Proc. 9th ACM Symp. on Document Engineering (DocEng 2019), 2009, pp. 218–221.

[15] S. Keerthi, C.J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Comput. 15 (7) (2003) 1667–1689.

[16] L.S. Kennedy, M. Naaman, Generating diverse and representative image search results for landmarks, in: Proc. 17th Int. ACM Conf. on World Wide Web (WWW 2008), 2008, pp. 297–306.

[17] L.S. Kennedy, M. Naaman, S. Ahern, R. Nair, T. Rattenbury, How Flickr helps us make sense of the world: context and content in community-contributed media collections, in: Proc. 15th ACM Int. Conf. on Multimedia (MM 2007), in: MM '07, ACM, 2007, pp. 631–640.

[18] M.L. Kherfi, D. Ziou, A. Bernardi, Image retrieval from the world wide web: issues, techniques, and systems, ACM Comput. Surv. 36 (1) (2004) 35–67.

[19] R. Kohavi, G. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.

[20] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, Emerg. Artif. Intell. Appl. Comput. Eng. (2007) 3–24.

[21] M. Lew, N. Sebe, C. Djerabe, R. Jain, Content-based multimedia information retrieval: state of the art and challenges, Trans. Multimed. Comput. Commun. Appl. 2 (1) (2006) 1–19.

[22] W.-Y. Ma, H.J. Zhang, Benchmarking of image features for content-based retrieval, in: Asilomar 11th Conf. on Circuits, Systems and Computers, 1, 1998, pp. 253–257.

[23] G. Madjarov, V.V. abd Ivic Dimitrovski, D. Kocev, Web genre classification with methods for structured output prediction, Inf. Sci. 503 (2019) 551–573.

[24] J. van der Meer, F. Boon, F. Hogenboom, F. Frasincar, U. Kaymak, A framework for automatic annotation of web pages using the Google rich snippets vocabulary, in: Proc. 26th ACM Symp. on Applied Computing (SAC 2011), 2011, pp. 765–772.

[25] J. Miano, Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP, ACM, 1999.

[26] A. Money, H. Agius, Video summarisation: a conceptual framework and survey of the state of the art, J. Vis. Commun. Image Represent. 19 (2) (2008) 121–143.

[27] L. Nie, X. Song, T.-S. Chua, Learning from multiple social networks, Synth. Lect. Inf. Concepts Retr. Serv. 8 (2) (2016) 1–118.

[28] L. Nie, M. Wang, Z.-J. Zha, T.-S. Chua, Oracle in image search: a content-based approach to performance prediction, ACM Trans. Inf. Syst. 30 (2) (2012) 1–23.

[29] L. Nie, S. Yan, M. Wang, R. Hong, T.-S. Chua, Harvesting visual concepts for image search with complex queries, in: Proc. 20th ACM Int. Conf. on Multimedia (MM 2012), 2012, pp. 59–68.

[30] E. Osuna, R. Freund, F. Girosi, Support Vector Machines: Training and Applications, Technical Report AIM-1602, 1997.

[31] S. Rudinac, M. Larson, A. Hanjalic, Learning crowdsourced user preferences for visual summarization of image collections, IEEE Trans. Multimed. 15 (6) (2013) 1231–1243.

[32] R. Senter, E. Smith, Automated readibility index, Aerosp. Med. Res. Lab. (1967) 1–14.

[33] X. Song, L. Nie, L. Zhang, M. Akbari, T.-S. Chua, Multiple social network learning and its application in volunteerism tendency prediction, in: Proc. 38th ACM Int. Conf. on Research and Development in Information Retrieva (SIGIR 2015), 2015, pp. 213–222.

[34] S. Thompson, Sampling, third ed., John Wiley & Sons, Inc., 2012.

[35] M. Tiago, J. Veríssimo, Digital marketing and social media: why bother? Bus. Horiz. 57 (2014) 703–708.

[36] D.P. Tian, A review on image feature extraction and representation techniques, Int. J. Multimed. Ubiquitous Eng. 8 (4) (2013) 385–395.

[37] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: Proc. 9th ACM Int. Conf. on Multimedia (ICM 2000), 2001, pp. 107–118.

[38] I.W. Tsang, J.T. Kwok, P.-M. Cheung, Core vector machines: fast SVM training on very large data sets, J. Mach. Learn. Res. 6 (2005) 363–392.

[39] V. Vapnik, C. Cortes, Support vector networks, Mach. Learn. 20 (1995) 273–297.

[40] WaybackMachine, A Digital Web Page Archive, 2018. http://archive.org,

[41] M. Wynblatt, D. Benson, Web page caricatures: multimedia summaries for WWW documents, in: Proc. IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS 1998), 1998, pp. 194–199.

[42] Y. Yang, X. Wang, T. Guan, J. Shen, L. Yu, A multi-dimensional image quality prediction model for user-generated images in social networks, Inf. Sci. 281 (2014) 601–610.

[43] L. Zhang, F. Lin, B. Zhang, Support vector machine learning for image retrieval, in: Proc. Int. Conf. on Image Processing (ICI 2001), 2, 2001, pp. 721–724.