Searching images in a web archive

André Mourão*†, Daniel Gomes†
FCT:Arquivo.pt
Avenida do Brasil 101, Lisbon, Portugal
†daniel.gomes@fccn.pt
NOVA LINCS, NOVA School of Science and Technology
Quinta da Torre, Caparica, Portugal
*a.mourao@campus.fct.unl.pt

***Abstract*—This article presents the research and development of a large-scale image search system applied to launch a word-wide innovative service that enables searching billions of historical images archived from the web since the 1990s. Contributions of this work were applied to enhance the Arquivo.pt web archive with an image-search service where users submit text queries, through a web user interface or an API, and immediately receive a list of historical web-archived images. However, supporting image search over web archives raised new challenges. The volume of data to be processed was big and heterogeneous, summing over 530TB of historical web data published since the early days of the web. The main contributions of this work are a toolkit of algorithms that extracts textual metadata to describe web-archived images, a system architecture and workflow to index large amounts of web-archived images considering their specific temporal features and a ranking algorithm to order image-search results by relevance. This research was applied to launch an enhanced image-search service that is publicly available since March 2021. All the developed software is fully available as free open-source software.**

***Index Terms*—Image search, web archive, web archive information retrieval**

I. Introduction

Arquivo.pt is a Big Data research infrastructure that preserves historical web data and provides public free tools to process it such as full-text search, APIs or data sets of derived results. The research work described in this article was applied to enhance Arquivo.pt with an image-search service, where users submit text queries and immediately receive a list of historical web-archived images and corresponding metadata. A service optimized to process and enable search over historical images archived from the web is a unique and precious tool useful to perform task such as web mining analysis of temporal data to derive market trends, extract information for data journalism or perform fact-checking about past-events. Supporting image search over the historical web-data preserved by web archives raised new challenges that live-web search engines do not need to address. These challenges include dealing with multiple versions of images and pages referenced by the same URLs, handling duplication of web-archived images over time, or ranking search results considering the temporal features of historical web data published over decades. In addition, the volume of data to be processed was big and heterogeneous summing over 530 TB of web data published and archived since the early days of the web.

This article describes the creation of the current Arquivo.pt's image search service deployed into a production system since March 2021 [1]. The goal was to develop a system which addresses the challenges raised by the inherent temporal properties of web-archived data, but at the same time provided a familiar look-and-feel similar to a live-web image search engine such as Google images to facilitate its adoption by common Internet users. The research questions that guided this work were:

- How to extract relevant textual content in web pages that best describes images (RQ1)?
- How to de-duplicate billions of archived images collected from the web over decades (RQ2)?
- How to index and rank search results over web-archived images (RQ3)?

The main innovative contributions from the presented research arise from the way it extracts metadata by identifying relevant textual content in web pages, deals with temporal web data by reducing index sizes by de-duplicating web-archived images, handles the big volume of information (1.862 billion web images) and it is completely open by freely providing public access to source code[1], image search API[2] and Web UIs[3].

II. Background and related work

Brin et Page [2] seminal article provided the first glimpse into what became the largest search engine in the world. It described the initial Google Page Rank algorithm, indexing data structures and workflow, metadata extraction techniques (e.g. anchor text extraction), system architecture and infrastructure. However, Google and other commercial search engines provide few details on how their systems work at scale. Existing publications are either outdated [3, 4] or provide high-level presentations [5]. In 2000, a green dress lead to the creation of Google Images [6]. Google Image Search has a share of 22.6% [7] of the Internet searches performed in the USA among the major image search engines but little is known about its applied features for ranking image search results or indexing architecture. Once again, the published information is

[1] https://github.com/arquivo/

[2] https://github.com/arquivo/pwa-technologies/wiki/ImageSearch-API-v1.1-(beta)

[3] https://arquivo.pt/

TABLE I
SEARCHABLE WEB-ARCHIVED IMAGES IN ARQUIVO.PT (CONTRIBUTION OF THIS WORK).

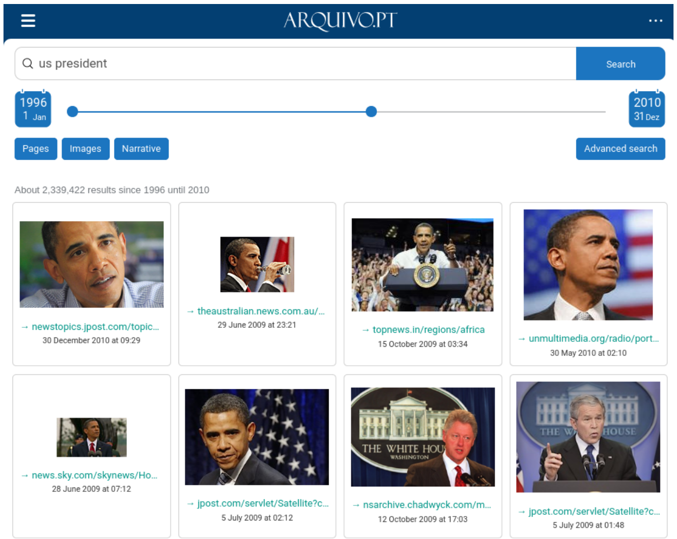
Oldest web-archived image	15 April 1994
Newest web-archived image	14 November 2020
Searchable images w/initial prototype (2020)	22,881,688
Searchable images w/this work (2022)	**1,862,311,456**

limited to high-level keynotes and presentations [8]. Live-web search engines update their *corpora* by the second, replacing old versions of images and other web files with the most recent information available online, removing the historical web data from the indexes, or making them unavailable to the public. Considering that 80% of web content is not available in its original form after only one year (e.g. updated or removed from the web) [9, 10], this constitutes a concerning continuous loss of historical data which consequently originates knowledge loss.

Web archives complement live-web search engines because they provide a temporal perspective about web data, raising challenging questions not addressed when dealing with the live-web. How to index web content collected over multiple years? How to deal with the resulting duplicated content? Should web images and pages which prevail along time be considered redundant? The Royal Danish Library blog contains a detailed description of the technical usage of SolrCloud at large scale (16 billion documents and 70 TB of data) to search archived web pages [11]. SolrWayback 4 [12] provides a system and pipeline, that goes from processing captured data to a fully-fledged web search UI. However, as search is focused on web pages, none of these systems was designed specifically to find images and little advantage has been taken from web-archived documents which are not text-based, such as images or videos. On the other hand, Costa's [13] user study on information needs of web-archive users identified image search as a user requirement which was not supported at the time of publication.

In 2009, Yahoo announced it was closing down the United States version of the Geocites. In 2016, the project *GifCities: The GeoCities Animated GIF Search Engine* was released[4]. It is a text-based animated GIF images search engine that processes the directory path and image filename to derive descriptive meta-data about the image. *Gifcities* contains 4.5 million animated GIF images (1.6 million unique images after de-duplication) extracted from archived Web resources. The Wayback Machine provides a search interface focused on images[5], limited to about 4 million donated images, not the full corpus of archived web data (828 billion web pages).

Arquivo.pt is a publicly available research infrastructure that provides access tools over historical web data, focused in the preservation of online information of general interest to the Portuguese community and research and education content at international level [1]. Arquivo.pt includes information in several languages and half of its users come from other countries than Portugal. Figure 1 presents a screenshot of the Arquivo.pt image search user interface[6]. Arquivo.pt image search goal is to enable finding images embedded in web pages considering their context and position in the page (e.g. which text is displayed close to embedded images). There is significant research in web page segmentation [14–16] and its application to caption extraction. However, from our experimental results these techniques were computationally heavy, for example they require parsing the full DOM hierarchy to determine the cut-off thresholds between page sections [17] . Thus, we developed a technique that scales to the billions of web pages we need to inspect and contributes to complement the limited literature published about live-web image search (described in Section III-C). Even less research was performed on how to search images in web archives with the additional challenges raised by the peculiar features of historical web data [18]. Müller-Budack et al. [19] analysed images from news articles in the Internet Archive and demonstrated the feasibility of deep learning techniques for the identification of public personalities in politics and entertainment. Mourão et Melo [20] describe an earlier version of Arquivo.pt image search system that enabled search on a subset of 24 million images.



About 2,339,422 results since 1996 until 2010

→ newstopics.jpost.com/topic...
30 December 2010 at 09:29

→ theaustralian.news.com.au/...
29 June 2009 at 23:21

→ topnews.in/regions/africa
15 October 2009 at 03:34

→ unmultimedia.org/radio/port...
30 May 2010 at 02:10

→ news.sky.com/skynews/Ho...
28 June 2009 at 07:12

→ jpost.com/servlet/Satellite?c...
5 July 2009 at 02:12

→ nsarchive.chadwyck.com/m...
12 October 2009 at 17:03

→ jpost.com/servlet/Satellite?c...
5 July 2009 at 01:48

Fig. 1. Arquivo.pt image search results for the query "US president" from 1996 to 2010.

Most articles in the academic literature focus on addressing peculiar and narrowed research problems, instead of the deployment of search in a fully working search service, despite this is the existing knowledge gap. Table I presents a comparison between the number of searchable web-archived images in Arquivo.pt between the initial prototype launched in 2020 and the result of this work in 2022. The number of searchable images increased from 22 to 1 862 million.

[4] https://gifcities.org/

[5] https://archive.org/details/image

[6] https://arquivo.pt/image/search?l=en

III. Algorithms to associate textual metadata to web images (RQ1)

The main goal of Arquivo.pt's image search is to enable finding images through textual queries. To build such a system for a large web archive, one first needs to find these images and associate related textual metadata extracted from all types of web-archived data along time (e.g. files following different versions of HTML or CSS formats). Images in HTML pages are used to illustrate the content of the page or subsection of a page, either as an inline image (<img> tag) or as a link (<a> tag). Web authors code textual descriptions for images (e.g ALT attributes) to be presented if the image cannot be displayed. This is useful if there is a server/connection error that prevents the image from loading or for users that rely on screen readers to browse the web. Extracting metadata from HTML using data captured for archival purposes poses specific challenges: separating images from the remaining web content, matching HTML pages to images and finding which page textual excerpts and metadata better represent each embedded or linked image. The following sections detail how we approached these challenges, taking into account the billion file scale of the web archive.

A. Identify pages related to images

The WARC (Web ARChive) format [21] is the standard to store information collected from the web, which aggregates and compresses multiple web resources together into a single WARC file. It can store any type of web resource content (e.g. web page, image, video, PDF) and related information (e.g. server response codes, headers). The ARC format was the predecessor of WARC and both formats co-exist in most web archives. Therefore, it is required to parse WARC and ARC files to find images and extract their corresponding metadata to generate indexes that support search over web-archived images using textual queries. The differences between ARC and WARC files do not impact most techniques described in this article. So, the term WARC will stand in for both ARC and WARC files. Web archives use crawlers to collect online information to be archived. Web crawlers iteratively collect and parse HTML pages to find links to additional resources. Linked resources such as associated images are added to a crawl queue to be archived later. For a longer description of the challenges behind this process, see [18], Part II. However, web pages and their corresponding embedded images are not instantly crawled and may be stored in different WARC files. This leads to our first challenge: matching images to pages so that we may associate relevant textual excerpts from web pages with embedded or linked images.

In WARC records, the image URL is stored as part of the image record or <*a*> tag *href* attribute (e.g. if the *href* ends with an image file extension). For HTML pages, one can find image references by parsing the HTML, extracting the relevant tags (<*img*>, <*a*>, CSS with *background-image*) and extracting the URL from the tags.

Algorithm 1 presents an high-level view of the process. To avoid processing WARCs multiple types (once for image, we first check if the current record matches an image or an HTML page, using the MIME type. URLs are normalized to SURT[7], a canonical URL/URI representation, to match similar URLs with different, equivalent capitalization and formats. If the record is HTML, we parse and transverse the HTML tree to find the relevant image tags (<*img*>, <*a*>, CSS with *background-image*), extract their matching *metadatas* (e.g. img ALT text) and add it to the metadata record set matching its SURT. If the record is an image (i.e. the JPEG of the image itself), we extract image specific metadata (e.g. image pixel size), filter them by size and add them to the metadata record set matching its SURT. After processing all WARC files, entries in *metadatas* will consist of a set of image metadata and page metadata.

An important note is that we exclude images that are too small (smaller than 50 pixels in height or width) or too large (area larger than 15000x15000 pixels). Our experimental results showed that smaller images were mainly icons or navigational images which were considered relatively irrelevant by the users when were displayed in a search engine results page. On the other hand, very large images were usually malformed files possibly due to errors during the crawling process, or caused unpleasant browser slowdown when they were presented in the search results page which was perceived by the users as a search engine malfunction.

The following section details what metadata is processed for images and pages. As images may be archived multiple times over time, and may be embedded on more than one web page, there may be several image or page metadata entries for the same SURT. Section IV-A will detail how we merge these records.

B. Assign HTML attributes to images

Each web page may contain multiple embedded images. Thus, it is important to identify the HTML attributes which reference each image so that we may extract the correct metadata. Web images can be published as a part of an <*img*> tag, <*a*> tag or CSS *background-image*. The metadata extraction includes the <img> *title* and *alt* attributes, an image caption extracted from the HTML page and the *anchorText* for <*a*> tags.

Page information extracted includes the title, crawl timestamp, weight, width, and a *imgDigest* generated using an SHA-256 hash from the image content. Parsing such a set of diverse web data ranging from 1992 to 2020, archived using different formats (ARC and WARC) and using different crawlers (e.g. Heritrix, Brozzler or even created manually from HTML files) resulted in unexpected problems in tasks such as decompression or character set encoding detection. We used Tika's text encoding detection function to identify the correct character set encoding used in the HTML page. Malformed records in WARC and ARCs (e.g. file closed unexpectedly, disk corruption, decompress error) were retried for recoverable errors (e.g. use different decompressor) or skipped if no useful page information was found.

[7] http://crawler.archive.org/articles/user_manual/glossary.html#surt

Algorithm 1: Matching images to pages

```
Input: Set of WARC files $warcs$;
Output: Map of metadata $metadatas$;
$MIN\_HEIGHT = MIN\_WIDTH = 50$;
$MAX\_HEIGHT = MAX\_WIDTH = 15000$;
$metadatas$ = {};
forall WARC $warc$ in warcs do
    forall Record $record$ in warc do
        if $record$ is HTML then // process page
            $imageRef$ = findImageTags($record.html$);
            forall ImageRef $imageRef$ in $imageTags$ do
                $imageSURT$ = getSURT($imageRef.url$);
                $m$ = extractPageMetadata($imageRef$);
                $metadatas[imageSURT]$.add($m$);
        else if $record$ is Image then // process image
            $imageSURT$ = getSURT($record.url$);
            $metadata$ = extractImageMetadata($record$);
            if $metadata.width > MIN\_WIDTH$ & $metadata.height > MIN\_HEIGHT$ & $metadata.height * metadata.width < MAX\_HEIGHT * MAX\_WIDTH)$ then
                $metadatas[imageSURT]$.add($metadata$);
return metadatas
```

C. Extract image captions from HTML

We found that only 18% of the web-archived images had an associated attribute *imgTitle* and 52% had *imgAlt*. There was no textual metadata for 45% of the web-archived images. Thus, assigning textual information to these images so that we may support search became a challenge. The approach adopted to address it was to process image URLs and page titles to obtain additional textual metadata. As a complement, we index text visually close to the image on the page (according to the HTML DOM hierarchy). This approach was derived from web page segmentation [14–17] and caption extraction [22] research. Our goal was to find a lightweight technique able to scale to billions of pages and images. Our method applies to <*img*> tags, as we found them to be assigned to about 90% of the images. Algorithm 3 is based on the HTML DOM hierarchy, with <*img*> tags in the leaves.

Figure 2 shows a simplified example of an HTML page tree with two images as leaf nodes (<*img*>) and potential captions as siblings of the image nodes. Images are encompassed in a <*div*> which also contains an adequate description of what is present in the image (abbreviated as "Foot...", "Cristiano..." and "Foot...", "Messi..."). However, the *parent* text technique fails for pages with "flat" structures, where most HTML elements are placed at the same level in the DOM hierarchy. This is especially prevalent on pages with lists of posts such as blogs, where image tags are presented at the same level as textual tags and information. Using the *parent* text technique will result in a caption with all the text in the page, losing the desired locality of our image caption extractor.

In HTML pages with this structure, the relevant text is present in the <*img*> *sibling* nodes (next to the tag in the DOM). Thus, if we can detect this structure, we could run the *get_text()* method on *sibling* nodes to get text that is close to the image in the page (*sibling text* technique). We need to find the level of the DOM hierarchy which has the highest amount of nodes to detect if pages have a "flat structure". This algorithm is detailed in Algorithm 2 which starting at an image node, goes up the DOM hierarchy and counts the number of children at each level, storing this "max child" depth. The algorithm ends when the top node is reached and the "max child depth" is returned.

We can now fully describe the caption extraction algorithm, algorithm 3: For each image node in a page, we run the *parent text* for each <*img*> tag in the HTML tree. If the first parent node is not the one with the most children, the page has a "regular" structure and we can use the *parent* method. If the first parent node with text is the one with the largest number of children, the page has a flat structure, and we resort to getting the text from the closest siblings that have *text content*. Some of these web pages contained over 10 000 <*img*> which caused the finding parents with text algorithm to be very expensive. As these pages often did not contain useful caption information, we also added a per page timeout on caption extraction tasks (default is 60 seconds per page).

Algorithm 2: Get Depth with Max Child Count

```
Input: HTML <img> $node$;
Output: $imgCaption$;

$depthMaxNodes$ = -1;
$depthMaxNodesCount$ = -1;
$parent$ = $node.parent()$;
while $parent$ is not null do
    if $parent.children().size()$ > $depthMaxNodesCount$ then
        $depthMaxNodes$ = $parent.depth()$;
        $depthMaxNodesCount$ = $parent.children().size()$;
return $depthMaxNodes$
```

IV. Indexing web-archived images (RQ2)

After obtaining textual metadata for the web-archived images, we must create index structures to support fast search service. However, in the context of web-archives, the prevalence of duplication and the constantly increasing volume of data raise new challenges.

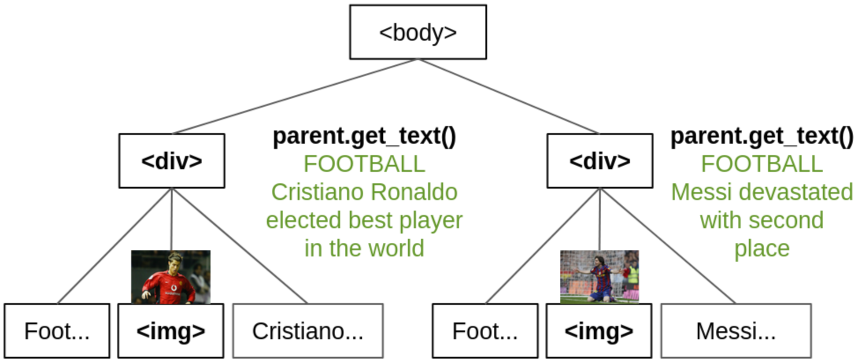


Fig. 2. Example of a HTML DOM hierarchy with image nodes and potential captions

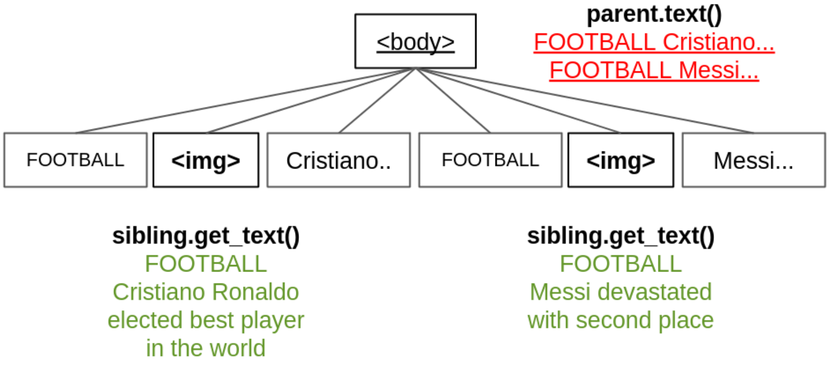


Fig. 3. Example of an unstructured HTML DOM hierarchy with image nodes with wrong parent captions and correct sibling captions.

Algorithm 3: Extract captions from HTML pages

```
Input: HTML <img> $node$;
Output: $imgCaption$;

$depthMaxNodes = getDepthMaxchild(node)$
if $parent.depth() > depthMaxNodes$ then
    return $getParentNodeText(node)$;
else
    $leftNode = node.leftSibling()$;
    while $leftNode.get\_text()$ is empty do
        $leftNode = leftNode.leftSibling()$;
    $rightNode = node.rightSibling()$;
    while $rightNode.get\_text()$ is empty do
        $rightNode = rightNode.rightSibling()$;
    return $leftNode.get\_text() +$
     $rightNode.get\_text()$ ;
```

A. De-duplicating images across time and space

The duplication problem in web archives is manifested in the dimensions of time and space. The same images are archived repeatedly over time. The same images are repeated across different web pages (web space). We initially planned to **ignore duplication**, index all images and address duplicates in run time when presenting search results to users. However, this over-simplistic approach was not applicable to a running service. Web users demand quick responses from search services typically below a few seconds. This requires that indexes are at least partially kept in memory, which is an expensive hardware resource. Indexing all images in our data set of 1.862 billion image records produced indexes too large which required unattainable additional hardware resources to support quick responses. Notice, that the solution of acquiring additional hardware is shortly applicable because web archives keep on collecting and preserving additional web data. Moreover, web archives are not the Internet Giants. Web archives are typically hosted by non-profit, educational or cultural heritage organizations with scarce resources. Therefore, research on how to minimise resources demand is crucial for them. Therefore, we decided to apply our efforts on studying how to de-duplicate web-archived images.

Metadata for images captured more than once is redundant for search purposes (i.e. same *imgAlt*, *imgCaption*, ...) and the obtained results showed that 70% of the web-archived images were duplicated (archived more than once). Our objective was to de-duplicate image information while keeping the most relevant metadata for all images so that we do not compromise search quality. For this purpose, we indexed the oldest page metadata based on the observation that web-archive users prefer oldest documents over the newest [23], while keeping all the image metadata captured over time.

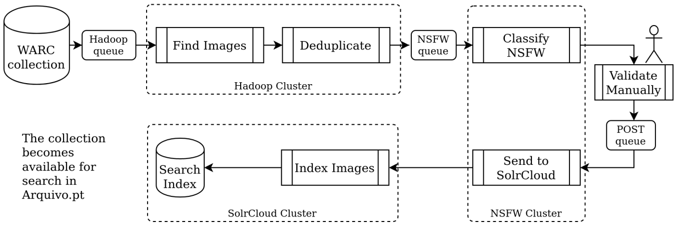


Fig. 4. Arquivo.pt image indexing workflow.

B. Assign Not Safe For Work ratings (RQ3)

Arquivo.pt automatically performs broad crawls of web pages hosted under the .pt domain. Thus, some of the images captured may contain pornographic content that users do not want to be displayed by default, for instance while using Arquivo.pt in a classroom. By default, Arquivo.pt hides pornographic images from the search results which were automatically classified as Not Safe for Work (NSFW). This filter can be disabled by the user through the image search interface or the API.

A previous version of the NSFW classifier is documented in [24]. Currently, Arquivo.pt applies an NSFW image classifier based on GantMan's model [25], a Deep Neural Network (DNN) solution. It is based on a TensorFlow [26] using a Inception v3 [27] network, trained with over 60 GB of images scrapped from the web. Instead of identifying images as safe or not safe, it returns the probability of an image belonging to one of five categories: *drawing* (SFW drawings), *neutral* (SFW photographic images), *hentai* (including explicit drawings), *porn* (explicit photographic images), *sexy* (potentially explicit images that are not pornographic such as woman in bikini). The *Nsfw* score is the sum of *hentai* and *porn* and it was computed for all web-archived images. Images are filtered from the search results if *nsfw* > 0.5.

V. Image indexing workflow (RQ3)

This section presents an overview of the indexing workflow of web-archived images that is executed on three separate processing clusters. The Hadoop cluster extracts and deduplicates metadata, the NSFW cluster classifies and assigns a NSFW rating to images and the SolrCloud cluster indexes and supports online search. These clusters are connected using a set of *Redis* queues which enable data to move sequentially across processing stages. The amount of data to process is considerable: 530TB of WARCs spread across 115 collections. To deal with this scale and simplify our workflow management, the processing of images is performed per collection (i.e. WARC files from the same collection are processed in bulk).

A. Extraction of images and metadata

The algorithm described in section III-A was transformed into Hadoop Map-Reduce jobs: Finding images and metadata and Content-based de-duplication. The task of Finding images and metadata is performed as a Map-Reduce process, which takes WARC files as input and outputs image and page metadata entries to HDFS, described in section III-A. The map process takes a set of WARC files, extracts page and image metadata for all images and page records and stores them in the HDFS entry matching their SURTs. On the reduce stage, for each page and image metadata in the *metadatas* set, create a new de-duplicated record (SURT-based de-duplication).

TABLE II
De-duplication statistics

	Count	% of no de-dup.	Diff. vs. prev.
Web-archived files	6,325,224,457	-	-
Web-archived images	2,443,485,866	-	-
Image Metadatas (ignore duplication)	1,962,799,850	100%	-
Matching SURT (Section V-A)	1,170,071,334	60%	-40%
Matching Digest (Section V-B)	983,373,297	50%	-14%
Solr collection dedup.	595,737,525	30%	-39%

B. Content-based de-duplication

The Content-based de-duplication task aims to simultaneously eliminate duplicates across time and space. This task is performed as a Map-Reduce process, that plugs directly into the output of the previous step: The Map stage parses a set of (W)ARC files, finds images and page <*img*> and passes the extracted data to the Reduce process, grouping image records and metadata <*img*> by SURT; On the **Reduce** stage, for each set of image and metadata results in a SURT, merges metadata and generates a combined JSON with the image and matching metadata. NSFW classification is performed as the last stage of the information extraction process on two servers with Nvidia P40 GPUs. The total metadata extraction time for the full 530TB of WARCs was 1346 hours.

C. Indexing textual metadata for web-archived images

The obtained textual metadata is stored using the Apache Solr Portuguese language analyser. We used the image digest as the identifier to enable deduplication of images across collections. Table II shows the impact of de-duplication at the different pipeline stages. The first column contains the stage in the processing pipeline. The second column shows the percentage of data compared to ignore duplication. The third column shows the reduction in data to index compared to the previous stage. The effect of de-duplication is apparent at all the stages of the pipeline. The *metadatas* stage would be equivalent to the ignore duplication stage described previously. Merging by SURT has the largest impact on the number of images to index. This was expected because most of the duplicates result from crawling multiple times the same pages over time (duplication in time) or to related web pages that reuse the same images such as logos or banner ads (duplication in space). The impact of cross-collection de-duplication was interesting, the obtained results showed that 39% of the web-archived images were duplicates that have already been archived in previous crawls.

TABLE III
METADATA STATISTICS

	Count	% of total
All images	595,737,525	100%
imgAlt or *imgTitle*	326,175,700	55%
imgCaption (generated from web page text)	526,081,214	88%
One of *imgAlt/Title/Caption*	541,375,820	91%

Table III shows how many images have the different types of metadata extracted from HTML. This table shows that only 55% of images have textual attributes directly assigned in the original HTML code of their hosting web pages (TITLE or ALT attributes). The remaining 45% of the images were found through the textual metadata obtained from their URLs or web page metadata. Our proposed algorithm to extract caption information related to images from the web pages is able to assign textual metadata to 88% of the images. Combining all approaches, the assignment of textual metadata to web-archived images was improved from 55% to 91%. This way, most of the web-archived images in Arquivo.pt could be indexed and became searchable by its users.

VI. SEARCHING WEB-ARCHIVED IMAGES (RQ3)

The indexing process comprised 595,737,525 images selected to support Arquivo.pt image search. This service searches images based on the user textual queries and retrieves the top results, i.e. the images most relevant to match the user inputted text.

A. Ranking features and algorithm

Arquivo.pt rankis the image search results based on the following fields, described in detail in section V-A: *imgTitle*, *imgAlt*, *imgCaption*, *imgUrlTokens*, *pageTitle* and *pageUrlTokens*. The Arquivo.pt image search system uses BM25 [28] ranking function for each field. Multiplicative boosts are then given to each field according to their importance. The image search ranking for a single term query is calculated according to Equation (1).

$$\begin{aligned} originalScore = {} & 4 \times imgTitleBM25 + \\ & 3 \times imgAltBM25 + \\ & 3 \times imgCaptionBM25 + \\ & 2 \times imgUrlTokensBM25 + \\ & 1 \times pageTitleBM25 + \\ & 1 \times pageUrlTokensBM25 \end{aligned} \tag{1}$$

Values correspond to the BM25 score of the query term for the ranking fields. Term weighting was performed empirically, by examining the content of the fields and how we expect it to be relevant to a particular image. Fields potentially more descriptive, such as *imgTitle* or *imgAlt* are weight heavier than less descriptive fields such as *imgUrlTokens*. Additional ranking scores are given to images that match query terms as phrase queries, pf1, pf2 and pf3[8]. These boosts are applied exponentially: $1000\times$ boost for ps1, $100\times$ boost for ps2 and $10\times$ boost for ps3. The rationale for choosing this boosting structure is to ensure that images which have textual metadata containing all query terms closer to each other are ranked higher, while penalizing images which have the query terms far apart in their metadata. This process matches the user's expectation and feedback, as most queries consist of person or institution names [13] and showed good results on our empirical evaluations. In the case of score ties, the following criteria are used: image capture timestamp by presenting the oldest first (web-archive users prefer oldest documents [23]), if tied, imgSURT (alphabetical order).

B. Deploying web-archive image search

We decided to adopt Apache SolrCloud 8.8.2 because it is widely used in the web archiving community (e.g. SolrWayback at the Royal Danish Archive) and the context of the Solr project (e.g. open-source, non-profit and non-commercial) are coherent with Arquivo.pt preservation policy. The index size was 629 GB, divided into 32 shards. The index was split across 4 nodes, meaning there are 8 shards per node. Load balancing and redundancy is achieved by using hash-based session distribution across our two branches (A and B). We have eight nodes available for SolrCloud, divided into two twin branches, A and B (four nodes per branch). Regarding Solr sharding, Figure 5 gives an overview of how this setup organized. To avoid relying on the Operating System file memory pagination and caching, we manually place the directories where the indexes are in memory using *vmtouch*[9]. *vmtouch* can manage file system RAM cache, and force files to be placed in memory without the risk of eviction. In addition, to enable placing the full index into memory without relying on Solr to warm up the cache, it enables the index to stay in memory across SolrCloud restarts. SolrCloud distribution enables querying any node in the cluster and receiving search results from all the nodes. However, nodes may have different amounts of RAM and threads available because it is hard to keep an uniform park of servers in a production environment along time. For instance, *Server 1* had double the amount of RAM (512 GB) than the remaining nodes. We set up a SolrCloud instance without shards to devote more RAM to query cache when larger amounts of hardware resources are available. We also set a heap size of 31 GB of RAM on all nodes, to benefit from Java's 32 bit pointer compression.

Table IV presents the obtained results for the response times of the API. These experiments were performed using a set of 1000 "two word" queries, extracted from the query logs. The experiments ran for 5 minutes, meaning that some queries may have been repeated over the experiment, which adequately models users search behavior. Experiments ran on JMeter and queried the API directly. The obtained results showed that the system can respond up to 50 concurrent users, while keeping

[8] https://solr.apache.org/guide/8_7/the-extended-dismax-query-parser.html

[9] https://github.com/hoytech/vmtouch

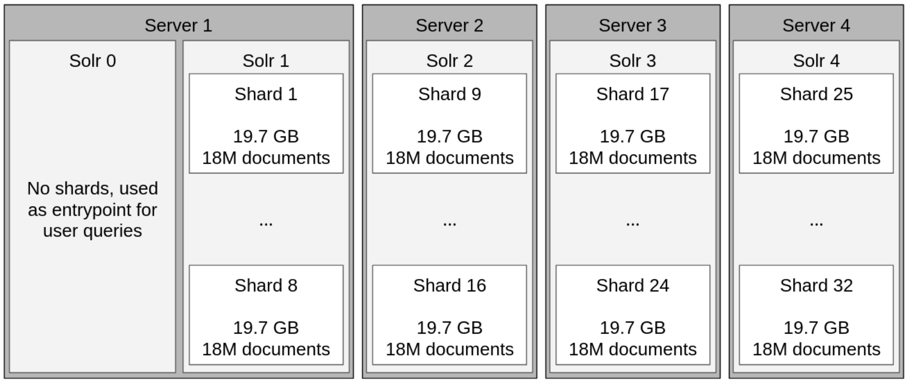


Fig. 5. SolrCloud shard distribution across servers

TABLE IV
API PERFORMANCE. Q/SEC REPRESENTS THE NUMBER OF QUERIES PER SECOND THAT THE SYSTEM CAN ANSWER

# requests	Avg.	Med.	$P_{95\%}$	$P_{99\%}$	Throughput
1	115 ms	74 ms	235 ms	769 ms	8 q/sec
3	120 ms	76 ms	259 ms	872 ms	24 q/sec
5	136 ms	85 ms	304 ms	1059 ms	36 q/sec
10	211 ms	128 ms	501 ms	1718 ms	46 q/sec
25	489 ms	266 ms	1297 ms	4334 ms	50 q/sec
50	970 ms	593 ms	2694 ms	6699 ms	50 q/sec

an average response time below one second. A free and open image search API was released, so that third-party software developers can integrate the Arquivo.pt image search results in their applications [10].

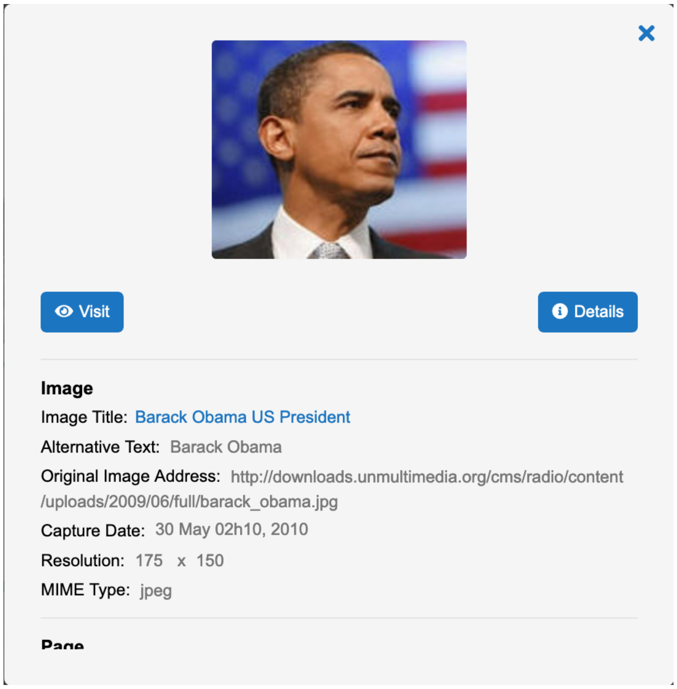
C. Web User Interface

The main goal of Arquivo.pt is to archive web data and make it accessible for everyone. Thus, image search must be made available in a user friendly manner, which can be browsed by web users of different levels of expertise. The user inputs a textual query and a set of image results are displayed in a grid layout to facilitate a quick choice of relevant images. When clicking on an image result, an image viewer is displayed, showing a larger image together with details about the image and web page where it was found, Figure 6. The UI also provides links to an archived Web page that embedded the image and to the API response fields.

Arquivo.pt's advanced search for images also provides multiple search result filters included in the API such as sentence or website search, date range, image format or size. You can find more information about search in the Arquivo.pt's image search FAQ [29, 30].

D. Image Search API

The Image Search API is documented on GitHub at https://arquivo.pt/api/imagesearch and the endpoint is located at https://arquivo.pt/imagesearch. It supplies the Arquivo.pt front-end. Providing an open and free API which returns image information and the original HTML from where the image was extracted from, follows one of Google's original goals [2] of building a large-scale search engine that can support novel research and it is available to the research community.



Image

Image Title: Barack Obama US President

Alternative Text: Barack Obama

Original Image Address: http://downloads.unmultimedia.org/cms/radio/content/uploads/2009/06/full/barack_obama.jpg

Capture Date: 30 May 02h10, 2010

Resolution: 175 x 150

MIME Type: jpeg

Page

Fig. 6. Image viewer for an image result

VII. Conclusions and Future Work

This article describes the research and development of the system that supports the Arquivo.pt image search service. It describes a set of algorithms which address the temporal features of historical web-data and increased the coverage of image search over a web archive. It discusses how to select textual metadata to describe web-archived images, how to de-duplicate images web-archived along time and how to deal with the large volume of web data archived along time. As an overall result of the application of this work, the number of

[10] https://arquivo.pt/api/imagesearch

searchable web-archived images increased from 22 to 1 862 million.

Considering the existing limited published literature, we believe that this article provides significant advancements in web-archive image search systems. Nonetheless, our work exposed plenty of opportunities for future work. Research possibilities to improve Web archive image search systems include extracting categories from images using scene recognition classifiers [31], generating image captions [32], retrieving similar images [33] or retrieving images by dominant colors [34]. Another important future goal would be to systematically evaluate and improve the quality of the ranking function that orders the image search results by relevance. We are in the process of creating an annotated image gold collection to fine-tune our ranking function and apply Learning-to-Rank models.

Acknowledgements We would like to thank Toke Eskildsen and Thomas Egense from the Royal Danish Web Archive for their inspirational work and valuable discussions regarding the indexing process and SolrCloud.

Arquivo.pt web archive preserved the content of the web links cited in this article so that these references may prevail valid over time. Thus, any broken link you may find cited in this article can be recovered from www.arquivo.pt. Arquivo.pt CitationSaver enables any user to archive web references automatically[11].

References

[1] D. Gomes, "Web archives as research infrastructure for digital societies: the case study of Arquivo.pt," *Archeion*, vol. 123, pp. 46–85, Nov. 2022. [Online]. Available: https://www.ejournals.eu/Archeion/2022/123/art/22601/

[2] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks*, vol. 30, pp. 107–117, 1998. [Online]. Available: http://www-db.stanford.edu/backrub/google.html

[3] L. A. Barroso, J. Dean, and U. Holzle, "Web search for a planet: The Google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, Mar. 2003.

[4] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A Distributed Storage System for Structured Data," *ACM Transactions on Computer Systems*, vol. 26, no. 2, pp. 4:1–4:26, Jun. 2008. [Online]. Available: https://doi.org/10.1145/1365815.1365816

[5] J. Dean, "Challenges in building large-scale information retrieval systems: invited talk," in *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2009, pp. 1–1. [Online]. Available: http://doi.acm.org/10.1145/1498759.1498761

[6] V. Riili, "18 years after Google Images, the Versace jungle print dress is back," Sep. 2019. [Online]. Available: https://blog.google/products/search/18-years-after-google-images-versace-jungle-print-dress-back/

[7] Fishkin, Rand, "The Evolution of Search Marketing, Rand Fishkin," 2018. [Online]. Available: https://theinbounder.com/videos/search-marketing-s-evolution-2018-and-beyond.html

[8] J. Dean, "Designs, lessons and advice from building large distributed systems," *Keynote from LADIS*, vol. 1, 2009. [Online]. Available: https://research.cs.cornell.edu/ladis2009/talks/dean-keynote-ladis2009.pdf

[9] A. Ntoulas, J. Cho, and C. Olston, "What's new on the web? the evolution of the web from a search engine perspective," in *Proceedings of the 13th international conference on World Wide Web*, ser. WWW '04. New York, NY, USA: Association for Computing Machinery, May 2004, pp. 1–12. [Online]. Available: https://doi.org/10.1145/988672.988674

[10] D. Gomes and M. J. Silva, "Modelling information persistence on the web," in *Proceedings of the 6th international conference on Web engineering*, ser. ICWE '06. New York, NY, USA: Association for Computing Machinery, Jul. 2006, pp. 193–200. [Online]. Available: https://doi.org/10.1145/1145581.1145623

[11] T. Eskildsen, "70TB, 16b docs, 4 machines, 1 SolrCloud," Nov. 2016. [Online]. Available: https://sbdevel.wordpress.com/2016/11/30/70tb-16b-docs-4-machines-1-solrcloud/

[12] J. Lauridsen, "SolrWayback 4.0 release! What's it all about?" Feb. 2021. [Online]. Available: https://sbdevel.wordpress.com/2021/02/12/solrwayback-4-0/

[13] M. Costa, "Information search in web archives," Ph.D. dissertation, Universidade de Lisboa, 2014. [Online]. Available: https://repositorio.ul.pt/handle/10451/16020?mode=full

[14] J. Kiesel, L. Meyer, F. Kneist, B. Stein, and M. Potthast, "An Empirical Comparison of Web Page Segmentation Algorithms," in *Advances in Information Retrieval. 43rd European Conference on IR Research (ECIR 2021)*, ser. Lecture Notes in Computer Science. Berlin Heidelberg New York: Springer, Mar. 2021.

[15] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Extracting Content Structure for Web Pages Based on Visual Representation," in *Web Technologies and Applications*, ser. Lecture Notes in Computer Science, X. Zhou, M. E. Orlowska, and Y. Zhang, Eds. Berlin, Heidelberg: Springer, 2003, pp. 406–417.

[16] ——, "VIPS: A VIsion based Page Segmentation Algorithm," Microsoft, Technical Report MSR-TR-2003-79, 2003. [Online]. Available: https://www.microsoft.com/en-us/research/publication/vips-a-vision-based-page-segmentation-algorithm/

[17] S. Alcic and S. Conrad, "A Clustering-based Approach to Web Image Context Extraction," in *The Third International Conferences on Advances in Multimedia*, Budapest, Hungary, 2011, p. 7.

[18] D. Gomes, E. Demidova, J. Winters, and Thomas Risse, Eds., *The Past Web: Exploring Web Archives*. Springer

[11] https://arquivo.pt/services/citationsaver?l=en

International Publishing, 2021. [Online]. Available: https://www.springer.com/gp/book/9783030632908

[19] E. Müller-Budack, K. Pustu-Iren, S. Diering, M. Springstein, and R. Ewerth, "Image Analytics in Web Archives," in *The Past Web: Exploring Web Archives*, D. Gomes, E. Demidova, J. Winters, and T. Risse, Eds. Cham: Springer International Publishing, 2021, pp. 141–151. [Online]. Available: https://doi.org/10.1007/978-3-030-63291-5_11

[20] A. Mourão and F. Melo, "Technical Report - Portuguese Web Archive Image Search," Arquivo.pt, Tech. Rep., 2021.

[21] I. International Organization for Standardization, "ISO 28500:2017 Information and documentation — WARC file format," International Organization for Standardization, Geneva, CH, Standard, Aug. 2017. [Online]. Available: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/80/68004.html

[22] P. Joshi and S. Liu, "Web document text and images extraction using DOM analysis and natural language processing," in *Proceedings of the 9th ACM symposium on Document engineering*, ser. DocEng '09. New York, NY, USA: Association for Computing Machinery, Sep. 2009, pp. 218–221. [Online]. Available: https://doi.org/10.1145/1600193.1600241

[23] M. Costa and M. J. Silva, "Understanding the Information Needs of Web Archive Users," in *Proc. of the 10th International Web Archiving Workshop*, Vienna, Austria, Sep. 2010, pp. 9–16. [Online]. Available: https://sobre.arquivo.pt/wp-content/uploads/understanding-the-information-needs-of-web-archive.pdf

[24] D. Bicho, "Automatic Identification of Not Suitable For Work images," Master's thesis, IInstituto Superior de Engenharia de Lisboa, Lisboa, 2019.

[25] G. Laborde, "Deep NN for NSFW Detection," 2019, publication Title: GitHub. [Online]. Available: https://github.com/GantMan/nsfw_model

[26] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and others, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *CoRR*, vol. abs/1512.00567, 2015, _eprint: 1512.00567. [Online]. Available: http://arxiv.org/abs/1512.00567

[28] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, Apr. 2009. [Online]. Available: https://doi.org/10.1561/1500000019

[29] Arquivo.pt, "Image search – sobre.arquivo.pt," 2022. [Online]. Available: https://sobre.arquivo.pt/en/help/help-about-image-search/

[30] ——, "Image advanced search – sobre.arquivo.pt," 2022. [Online]. Available: https://sobre.arquivo.pt/en/help/advanced-image-search/

[31] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[32] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 3156–3164. [Online]. Available: http://ieeexplore.ieee.org/document/7298935/

[33] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning Fine-Grained Image Similarity with Deep Ranking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1386–1393, iSSN: 1063-6919.

[34] J. Wang and X.-S. Hua, "Interactive Image Search by Color Map," *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 1, pp. 12:1–12:23, Oct. 2011. [Online]. Available: https://doi.org/10.1145/2036264.2036276