DejaVu: A System for Journalists to Collaboratively Address Visual Misinformation

Hana Matatov[1], Adina Bechhofer[2], Lora Aroyo[3], Ofra Amir[1], Mor Naaman[4]
[1] Technion [2]Columbia University [3]Vrije Universiteit [4]Cornell Tech
{hanama888,adinabec,l.m.aroyo}@gmail.com,oamir@technion.ac.il,mor.naaman@cornell.edu

ABSTRACT

Journalistic work increasingly depends on information from web sources and social media. Visual misinformation, for example images that have been manipulated or taken out of context, pose a significant issue for journalists using these sources. Based on informal interviews with working journalists, we developed DejaVu, a system that supports journalists in the task of detecting visual misinformation. DejaVu streamlines the task of looking for near-identical image matches using reverse image search, and extends it by crawling and indexing rogue social media sites such as 4chan. More importantly, DejaVu supports collaboration between journalists by allowing them to flag images, which are then indexed such that the image and its near-duplicates are highlighted for other journalists regardless of where on the Web they find them. A preliminary evaluation of DejaVu's visual indexing shows that it can support such collaboration even when flagged images are re-posted after being further manipulated.

KEYWORDS

Visual Misinformation, Journalism, Computer Vision, Social Media, Image Similarity, Image Indexing

ACM Reference format:
Hana Matatov[1], Adina Bechhofer[2], Lora Aroyo[3], Ofra Amir[1], Mor Naaman[4]. 2019. DejaVu: A System for Journalists to Collaboratively Address Visual Misinformation. In *Proceedings of Computation+Journalism Symposium, Miami, FL, USA, February 2019 (C+J'19),* 5 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Visual misinformation poses a significant challenge for journalists trying to verify the information they publish [9], especially in the context of breaking news events. First Draft News had proposed seven distinct types of problematic content on the Web[1], such as misleading content, manipulated or fabricated content, and sharing of genuine content out of context. These types of issues are also manifested in visual misinformation which includes, for example, adding incorrect captions (mislabeled content), editing the visuals, cropping the image to remove specific information, manufacturing synthetic deep fakes, or reusing images in misleading context[2].

Existing tools[3] were developed to help journalists (and others) find information about an image, e.g. whether earlier instances of the same image appear on the Web or on websites dedicated to addressing misinformation (for example Snopes[4]). Google Reverse Image Search (RIS)[5], for instance, shows similar and near-duplicate images given a query image [1]. However, RIS does not address all journalistic needs. For example, it is hard to quickly identify the *earliest* appearance of near duplicates. Another limitation is lack in coverage, in particular of social media sources like 4chan or Reddit. These sources are particularly important because they are often the source of visual misinformation.

Furthermore, even the most efficient tools require a significant manual effort from each journalist interested in verifying the authenticity of specific image. Once a journalist finds that an image is questionable, it is hard to share that information even within the same organization, as images could appear in different locations on the Web, can be named differently and can be edited or manipulated in various ways.

In this work, we present DejaVu: a system designed to support the process of identifying visual misinformation. First, DejaVu supports collaboration between trusted journalists by building a visually-indexed shared database of flagged images. This database enables sharing of information about suspicious images in a manner that easily fits journalists' workflow: as a browser-based annotation over the image wherever it appears on the Web. Second, the system supports expanded reverse-image-search coverage with an interface that shows duplicate and near-duplicate images from Web and social media (e.g. Reddit or 4Chan) at the same time, even if those images were previously deleted. In addition, DejaVu streamlines RIS results for the journalistic task by augmenting the Google RIS API, filtering the results to show only the near-duplicates, and using temporal sorting that prioritizes older images.

[1]https://firstdraftnews.org/fake-news-complicated/

[2]https://www.nytimes.com/2018/10/24/world/americas/migrant-caravan-fake-images-news.html

[3]e.g., https://www.getsurfsafe.com/

[4]https://www.snopes.com/

[5]https://images.google.com/

DejaVu was developed based on needs we elicited from journalists and misinformation experts through informal interviews with leading practitioners (e.g., at ABC News and The New York Times).

To examine the potential effectiveness of the system for journalists we performed an initial evaluation looking at the effectiveness of the system's visual indexing. The evaluation seeks to understand whether the performance can match the scenarios that journalists might face when working and collaborating around visual content verification. Our results show that DejaVu's performance with near-duplicates, manipulated in different manners, can indeed support these needs.

2 REQUIREMENTS AND IMPLEMENTATION

In a series of face-to-face interviews with NY-based journalists and information workers (including at ABC News and The New York Times) and through an exploration of online resources for content verification, we have assembled a list of requirements for journalistic needs and the associated system functionalities, including:

Streamline the reverse image search process. We found that journalists typically search for identical or similar images through the Google and Yandex[6] RIS services. The journalist would query the examined image using each service separately, and inspect the individual results separately, often clicking through many images to find the earliest possible instance of the image, which can provide evidence of its use in different context. DejaVu provides an integrated view of the near-duplicate results for each image, and sorts these results to highlight earlier instances of these images.

Index and search social media sources. Currently, most RIS tools focus on images posted on the web and offer limited coverage of images posted on social media. Sources like Reddit and 4chan are not included in Google's (consumer-facing) RIS result, but the journalists we interviewed expressed the need to be able to search these sources. DejaVu thus regularly collects and indexes images from key social media sources. As some of the social media sources like 4chan are ephemeral (i.e. URLs to images posted there do not persist) we also store the images in our own datastore.

Support collaborative image annotation. We found that journalists, even within the same organization, have very limited ability to find or share information about images they investigate. As a result, they suspect that work is often duplicated as multiple journalists attend and spend time on each image. Current tools that journalists use to overcome this challenge are problematic. For example, some journalists had taken to using a Slack channel to share images they detected as suspicious. However, such methods do not integrate into the journalists' workflow (e.g. requiring them to post, or search a specific Slack channel) and are not easily query-able. With DejaVu, a journalist can flag an image, which then gets added to a shared dataset, indexed by its visual features along with the image metadata and the flagging explanation provided by the journalists when reported. The image and annotation would then show up whenever another journalist searches for an duplicate or near-duplicate of the flagged image.

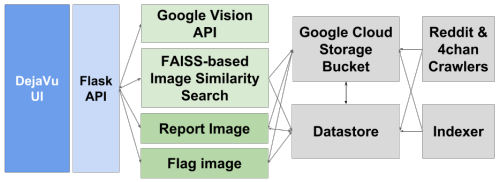


Figure 1: DejaVu System Architecture

The main flow of DejaVu is based on image search: a journalist uses a query image to retrieve duplicates and near duplicates organized in three groups: previously flagged images (and all the metadata associated with the flag), images from social media sources, and web-based images retrieved using reverse image search. At any point, the journalist can further inspect the query image or any other images in the results. Instead of search, DejaVu can make information available using a browser extension, where an image on any webpage can be clicked to trigger a search, or automatically be highlighted if a similar image was previously flagged.

Note that DejaVu is *not* a tool for identifying whether an image was manipulated or photoshopped. Other tools exist for this purpose, for example some concerned with *image archeology* [7], *image phylogeny* [2] or *image provenance analysis* [10], or that help a journalist closely examine the visual properties of a query images[7]. Instead, the system can help journalists not repeat work that is involved with such tasks.

DejaVu System Description

The DejaVu system has four main components: (1) *Crawlers* that scrape images from selected boards on 4Chan and Reddit several times a day; (2) A scalable *Image Index* that computes and stores image features to allow for retrieval of similar images; (3) A *retrieval engine for similar images*, and (4) a *Web Interface* that provides the multiple-source image-based search, and flag images or report new images by uploading their URL. An overview of the technical implementation DejaVu is provided in Figure 1, and more details about the key technical components is provided next.

[6]https://yandex.com/images/

[7]https://www.invid-project.eu/tools-and-services/invid-verification-plugin/

Social Media Crawler. In order to compliment existing efforts to monitor Twitter as a potential source of manipulated images, and to fulfill the requirement indicated in the face-to-face interviews, we first focused on Reddit[8] and 4chan[9] as a source of images.

- Reddit Image Crawler: We crawl Reddit using the Pushshift[10] API. We currently crawl two subreddits that are known as sources of misinformation: "r/the_donald" and "r/conspiracy". Every five hours we crawl the last 1000 posts from each subreddit, and save the posted images along with their metadata.
- 4Chan Image Crawler: Using 4chan's read-only API, we check every four hours for new images that were uploaded after the previous crawl, and add them with their metadata to the index. We are focusing on images from posts that appear in the "/pol/ - Politically Incorrect" board on 4chan.

The two social media crawlers store the crawled image files in a Google Cloud storage bucket, and use a Google Datastore for the image metadata.

We have crawled the image content from the listed subreddits and from 4chan starting from July 2018. At the time of submission, the dataset included 391,052 images that are used in the evaluation below.

Visual Features and Image Indexing. To support retrieval of near-duplicates, we index images by computing visual features for each image in the dataset, similar to the Provenance Image Filtering task [10]. There are several ways to represent images in lower-dimension to enable retrieval [6, 8, 10]. We chose to represent images using ORB descriptors [11], as they are fast to compute, open-source, and are based on geometric features likely to remain in images even after some manipulation. Each ORB descriptor is a 256 bit vector describing unique selected points in the image. Our implementation extracts a maximum of 200 descriptors ("features") for each image.

We used Principal Component Analysis (PCA) [4] to reduce each feature from 256 into a vector of 128 bits. The PCA model was bootstrapped using 10,000 of the images. Subsequent images go through dimensionality reduction using the trained PCA model.

To allow for efficient image search, we construct an index using the FAISS (Facebook AI Similarity Search) package, a library for efficient similarity search and clustering of dense vectors [5]. New images are added to the index in batches as they are crawled.

[8]https://www.reddit.com/
[9]http://www.4chan.org/
[10]https://pushshift.io/

Image Searching and Retrieval. When search is invoked for a query image, a similar process occurs: ORB descriptors are computed for the image and their dimensions are reduced by using the trained PCA model.

Then, the FAISS search function is used to compare the query image descriptors to the descriptors of all images in the dataset and obtain similarity scores. The query is using a k-nearest-neighbor search (KNN) [3] with k=100, i.e. for each feature of the query image, we find its 100 nearest neighbors in the dataset. For each image in the index, the similarity score to the query image is calculated as the number of its features matched by the KNN search to the query image features. DejaVu then returns the top N most visually similar images to the query image, i.e., those with the highest scores.

API and User Interface. The interaction as described above is currently realized as a web interface implemented in `Flask`. The interface allows journalists to perform search using a query image and report suspicious images as described above. The system is modular such that an API is available, returning the set of image matches based on a query image. We plan to make this API available with the system's official launch.

3 EVALUATION

We present an initial evaluation of DejaVu to test the capabilities and limits of the similarity-based indexing. Our goal was to assess how well our visual indexing finds matches in the image index and where it might fail, exposing its strengths and weaknesses.

Evaluation Procedure. To evaluate the search performance, we manually generated manipulated images of 9 original images which appeared in recent political news. We picked different types of images, e.g. images that depict people, buildings, landscapes, etc. Using manually manipulated images allowed us to calculate recall (i.e., how many of these images were retrieved) as part of the system assessment. This is necessary, as there was no ground truth with respect to the total number of matching images in our dataset.

For each image we manually created several manipulations, inspired by a common set of edits that people may perform. These manipulations include adding or removing text, adding or removing other visual markup, recoloring, cropping, flipping, changing resolution and adding parts of other images. We also added some more challenging types of manipulations, where the manipulated images present a different scene or view from the same situation as the query image.

In total we created 11 manipulated images for each of the 9 original images. We then added these manipulated images and the original images to our index. Figure 2 shows

an example of one of our query images and a subset of its manipulated versions.

To asses the ability of DejaVu to retrieve accurately manipulated images, we queried DejaVu with each of the 9 *original* images and analyzed the results.

To simplify the analysis below, both for recall and (for fairness) precision metrics calculation, when examining the retrieved images, we considered only our manually created manipulations as matches, and ignored correct matches to the query image that were existing images in the index (e.g. matching images crawled from Reddit or 4chan).

Evaluation Metrics. The goal of DejaVu is to find all similar images (relevant images) to a given query image, sorted according to similarity score. Thus, we used the standard retrieval metrics of precision and recall.
Specifically, we computed:

- *Recall at top K:* Calculated as the proportion of relevant images that were retrieved in top K results, where K=10,15,20.
- *Precision at top K:* Calculated as the proportion of the relevant images in top K results, where K=10,15,20.

Note that we have 12 instances (11 manipulation and one original) of each query image in the index. Also note that the K values offer a trade-off between precision and recall. That is, using K=10 limits the potential recall (as there are 12 images to retrieve), while using K=20 limits the potential precision for the same reason.

4 RESULTS

A summary of the evaluation results is shown in table 1. Overall, DejaVu successfully retrieved manipulated images, with average precision of 0.86 and average recall of 0.71 for top 10 results (K=10).

The recall at top K was identical for K=15 and K=20 for each query. Meaning that after the 15th results, no more new matches where found for any of the images. The best-performing images in terms of recall (image 2 and image 6) had retrieved all but one matching photo (11 of 12) in the first 15 results (K=15), but for worse-performing image (image 3) the system only retrieved seven of the matches (7 of 12), even when 20 images were retrieved (K=20).

Since the recall scores show that when matching images were retrieved, they were usually within the first 10 results, the precision scores accordingly decrease after $K = 10$. For K=10, precision values ranged from 0.7 and 1, meaning that most of the matching images in the first ten results were relevant.

We did not observe any significant differences in the retrieval performance on different types of images. For example, in our limited set of query images, an image of a human face and an image of a building resulted in identical precision and recall values.

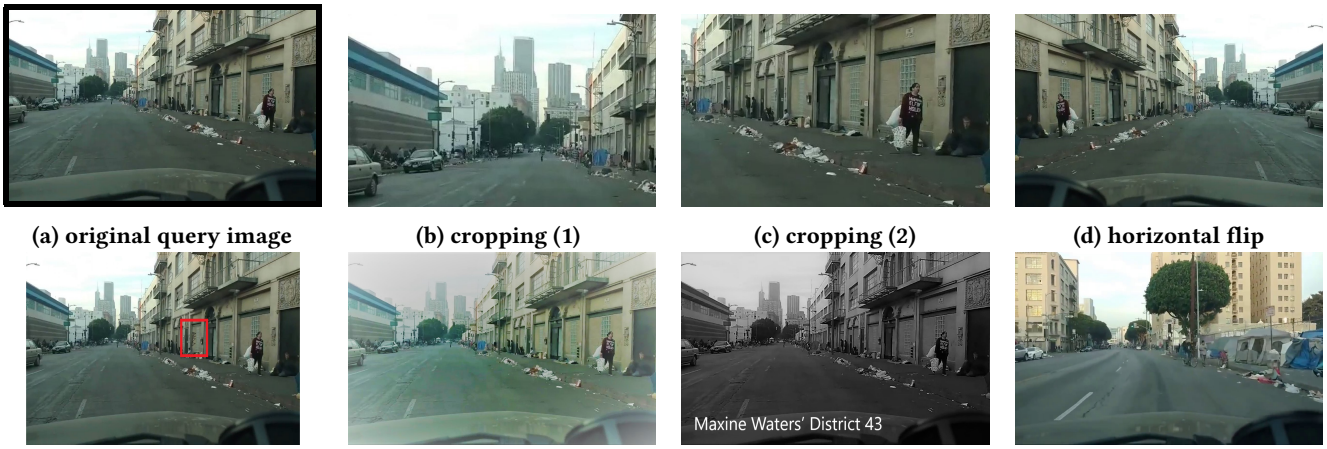
We examined which manipulation types DejaVu handles well and which ones it fails to identify. We found that DejaVu typically successfully identified color changing (including black& white), resolution manipulations, cropping and adding or removing text or other visual markups. However, in some cases combinations of more than one manipulation, extreme changes of colors or add-ons that appear on most of the image area resulted in lower scores. Only rarely DejaVu managed to determine a flipped version of the query image as a similar result. DejaVu also rarely retrieved manipulated images which present a different scene or view from the same situation as the query image, e.g. if the image is part of a video. This is not surprising, since changing the scene is an extreme manipulation. Some of these issues would be easy to address (e.g., by querying with a flipped version in addition to the original), though others will require more sophisticated solutions.

5 CONCLUSION AND FUTURE WORK

In this paper we presented DejaVu, a collaborative system for identifying visual misinformation which aims to support journalists in this task. DejaVu streamlines the reverse image search task, for example by adding results from social media sources like Reddit, and organizing results to highlight the earliest instances of such images on the web. The system also allows journalists to flag images as suspicious or report new images by uploading them to the shared dataset. Journalists who are examining the same (or visually similar) images on the Web, can then be alerted of the flag, regardless of where on the Web the image they examine appears.

We performed a preliminary evaluation to assess DejaVu's ability to detect and retrieve manipulated near-duplicate images. We found that the DejaVu system is capable of gathering and maintaining a large repository of images, index them and successfully retrieve similar images resulting from a range of manipulation types.

In future work we plan to extend DejaVu in several ways. First, we will explore additional computer vision methods to improve the retrieval of manipulated images, in particular those manipulations that DejaVu failed retrieving, and expand the range of manipulations types that the system supports. Second, we will add features requested by journalists, such as extracting the publishing dates of images (when available), and adding capabilities for identifying manipulated videos. Finally, we plan to deploy the system in newsrooms and study its use by journalists to better understand their workflows and improve the system to match their needs.



(a) original query image (b) cropping (1) (c) cropping (2) (d) horizontal flip

(e) addition of visual markup (in red) (f) change of color (g) b&w filter & added text (h) different scene

Figure 2: Example from the Ground Truth Collection. (a) shows the original images, (b)–(h) show example manipulations.

Metric	1	2	3	4	5	6	7	8	9	Average
Precision@10	0.8	1	0.7	0.9	0.8	1	0.8	0.8	0.9	0.856
Precision@15	0.533	0.733	0.466	0.6	0.533	0.733	0.533	0.533	0.6	0.585
Precision20	0.4	0.55	0.35	0.45	0.4	0.55	0.4	0.4	0.45	0.438
Recall@10	0.666	0.833	0.583	0.75	0.666	0.833	0.666	0.666	0.75	0.713
Recall@15	0.666	0.916	0.583	0.75	0.666	0.916	0.666	0.666	0.75	0.731
Recall@20	0.666	0.916	0.583	0.75	0.666	0.916	0.666	0.666	0.75	0.731

Table 1: Evaluation Results, showing precision and recall for different top K values (K=10,15,20). The columns 1-9 represent 9 different query images.

ACKNOWLEDGMENTS

The work was supported in part by Oath and Yahoo! Research through the Connected Experiences Lab at Cornell Tech. Hana Matatov was supported in part by the *Ministry of Science and Technology* of Israel.

REFERENCES

[1] Luiz Andre Barroso, Jeffrey Dean, and Urs Hölzle. 2003. Web Search for a Planet: The Google Cluster Architecture. (2003). https://ai.google/research/pubs/pub49

[2] Zanoni Dias, Siome Goldenstein, and Anderson Rocha. *Large-Scale Image Phylogeny: Tracing Image Ancestral Relationships.* Technical Report. http://www.ic.unicamp.br/~siome/papers/Dias-IMM-2013.pdf

[3] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2013. Optimized Product Quantization for Approximate Nearest Neighbor Search. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2946fi?!2953. DOI : http://dx.doi.org/10.1109/CVPR.2013.379

[4] I.T. Jolliffe. 2011. Principal Component Analysis. In *Principal Component Analysis, Second Edition*, Lovric M. (eds) International Encyclopedia of Statistical Science (Ed.). Springer, Berlin, Heidelberg. http://cda.psych.uiuc.edu/statistical_learning_course/JolliffeI.PrincipalComponentAnalysis(2ed.,Springer,2002)(518s)_MVsa_.pdf

[5] Jeff Johnson, Matthijs Douze, and Herv Jégou. 2017. Billion-scale similarity search with GPUs. (2 2017). http://arxiv.org/abs/1702.08734

[6] Yan Ke, Rahul Sukthankar, and Larry Huston. 2004. *Efficient Near-duplicate Detection and Sub-image Retrieval.* Technical Report. http://www.cs.cmu.edu/~rahuls/pub/mm2004-pcasift-rahuls.pdf

[7] Lyndon Kennedy and Shih-Fu Chang. 2008. *Internet Image Archaeology: Automatically Tracing the Manipulation History of Photographs on the Web.* http://www.ee.columbia.edu/~lyndon/pubs/acmmm2008-manipulation.pdf

[8] David G Lowe. 2004. *Accepted for publication in the.* Technical Report. https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[9] Alice Marwick and Rebecca Lewis. 2017. Media Manipulation and Disinformation Online. (2017). https://datasociety.net/pubs/oh/DataAndSociety_MediaManipulationAndDisinformationOnline.pdf

[10] Daniel Moreira, Aparna Bharati, Joel Brogan, Allan Pinto, Michael Parowski, Kevin W Bowyer, Patrick J Flynn, Anderson Rocha, and Walter J Scheirer. 2018. Image Provenance Analysis at Scale. (2018). https://arxiv.org/pdf/1801.06510.pdf

[11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: an efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*. IEEE, 2564–2571. http://www.willowgarage.com/sites/default/files/orb_final.pdf