



LUND UNIVERSITY

FBAadLibrarian and Pykognition: open science tools for the collection and emotion detection of images in Facebook political ads with computer vision

Schmøkel, Rasmus; Bossetta, Michael

Published in:
Journal of Information Technology & Politics

DOI:
[10.1080/19331681.2021.1928579](https://doi.org/10.1080/19331681.2021.1928579)

2022

Document Version:
Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):
Schmøkel, R., & Bossetta, M. (2022). FBAadLibrarian and Pykognition: open science tools for the collection and emotion detection of images in Facebook political ads with computer vision. *Journal of Information Technology & Politics*, 19(1), 118-128. <https://doi.org/10.1080/19331681.2021.1928579>

Total number of authors:
2

Creative Commons License:
CC BY-NC

General rights

Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00



FBAdLibrarian and Pykognition: open science tools for the collection and emotion detection of images in Facebook political ads with computer vision

Rasmus Schmøkel & Michael Bossetta

To cite this article: Rasmus Schmøkel & Michael Bossetta (2021): FBAdLibrarian and Pykognition: open science tools for the collection and emotion detection of images in Facebook political ads with computer vision, Journal of Information Technology & Politics, DOI: [10.1080/19331681.2021.1928579](https://doi.org/10.1080/19331681.2021.1928579)

To link to this article: <https://doi.org/10.1080/19331681.2021.1928579>



© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.



[View supplementary material](#)



Published online: 31 May 2021.



[Submit your article to this journal](#)





[View related articles](#)



[View Crossmark data](#)

FBAAdLibrarian and Pykognition: open science tools for the collection and emotion detection of images in Facebook political ads with computer vision

Rasmus Schmøkel  and Michael Bossetta 

ABSTRACT

We present a methodological workflow using two open science tools that we developed. The first, FBAAdLibrarian, collects images from the Facebook Ad Library. The second, Pykognition, simplifies facial and emotion detection in images using computer vision. We provide a methodological workflow for using these tools and apply them to a case study of the 2020 US primary elections. We find that unique images of campaigning candidates are only a fraction (<.1%) of overall ads. Furthermore, we find that candidates most often display happiness and calm in their facial expressions, and they rarely attack opponents in image-based ads from their official Facebook pages. When candidates do attack, opponents are portrayed with emotions such as anger, sadness, and fear.

KEYWORDS

Computer vision; emotions; visual political communication; computational social science; digital methods; social media; political campaigning

Introduction

Despite strong public interest in the electoral implications of social media, little remains known about political advertising on these platforms. While we know that campaigns advertise on social media for persuasion, fundraising, or collecting data (Bossetta, 2018; Kreiss, Lawrence, & McGregor, 2018), scholars lack systematic, empirical evidence about the content and emotional valence of these advertisements. Therefore, we developed two open science tools to assist researchers in the collection and analysis of images from the Facebook Ad Library.

First, FBAAdLibrarian is a data collection tool that assists researchers in retrieving images from the Ad Library. Second, Pykognition aids in providing emotion classifications from facial displays using industry-grade computer vision. In this article, we outline our theoretical motivations for building FBAAdLibrarian and Pykognition. Then we describe their functionalities, provide a suggested workflow for analyzing images in Facebook ads, and apply this workflow to an example case that compares candidates' emotional self-presentation during the 2020 US primary elections.

Theoretical motivations


Outside of experimental designs, assessing the role of visuals in digital political communication has proven

difficult. While some platforms offer publicly available data through their APIs, this data is typically text-based. As a result, scholars have honed methods for analyzing social media *texts*, but the study of social media *images* remains relatively nascent.

Scholars are therefore turning to computer vision techniques to analyze political images at scale (Haim & Jungblut, 2020). Such methodological innovations are crucial to our understanding of contemporary elections and democracy. Political campaigns increasingly incorporate social media in their electioneering, and most platforms support embedding images and videos into organic posts and advertisements. Visual media are persuasive in a political context because they can convey non-verbal behavior such as facial expressions, which humans prioritize when processing information (Grabe & Bucy, 2009). Thus, we set out to develop software that assists researchers in retrieving and classifying emotions in images from the Facebook Ad Library API.

Facebook launched the Ad Library to provide more transparency into paid advertising around elections. The Ad Library provides a searchable archive of political advertisements across several of Facebook's products, such as Facebook, Instagram, and Messenger. While the Ad Library opens up a data stream for analyzing political ads, several scholars have noted shortcomings that directly

CONTACT Michael Bossetta  michael.bossetta@kom.lu.se  Department of Communication and Media, Lund University, Lund, Sweden

 Supplemental data for this article can be accessed [here](#).

© 2021 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

impact research. Leerssen, Ausloos, Zarouali, Helberger, and de Vreese (2018) critique the Ad Library along the vagaries of defining a “political” ad, its verification processes, and a lack of detail regarding targeting practices. Bossetta (2020) further highlights how the Ad Library’s search requirement limits scholars’ understanding of political advertising at an aggregate level.

In addition to these critiques, we add that the Ad Library currently lacks a feature to systematically export the visual components of political ads. We therefore built FBAdLibrarian to assist researchers in collecting and archiving images from the Facebook Ad Library.

FBAdLibrarian

FBAdLibrarian¹ is a command line tool that collects images from hyperlinks offered by Facebook’s Ad Library API. Facebook currently permits downloading images associated with individual ads for research purposes (Facebook, n.d.). The Librarian assists researchers who have verified their identity with Facebook in this process. First, the Librarian takes the output of the API and prepares a hyperlink for each ad using the researcher’s access tokens. Then, the Librarian looks up each ad individually. If the ad includes an image, the Librarian saves the image to an output folder and names the image according to the ad’s unique identification number. If the ad includes a video, the Librarian will pass over the ad, but it will document it as a video ad in an output “metadata.txt” file. This allows researchers to see the proportion of images to videos in their data, as well as retrieve videos manually if desired.²

Depending on one’s research inquiry, image collection may be all that a researcher needs to conduct a qualitative or manual coding analysis. However, we also designed Pykognition to lower the barriers in applying computer vision to detect emotions in political images.

Pykognition

Pykognition³ is a Python wrapper for Amazon Web Service’s (AWS) Rekognition API, which provides industry-grade face and emotion detection. For facial detection, the algorithm provides a confidence score for the predicted probability that the image includes

a face (or multiple faces). Each face is categorized in a FaceDetail object, which includes metadata such as predicted age, gender, and the emotion predicted to be displayed by each face. It should be noted that scientific evidence is weak that *internal* emotional states can be detected from facial expressions (Barrett, Adolphs, Marsella, Martinez, & Pollack, 2019). However, we consider these labels valuable for classifying *external* facial configurations, especially for public actors who engage in strategic messaging. For our purposes here, we use the term emotion to keep with industry terminology, without engaging in academic debates on the nuanced intricacies of affect and emotion. The emotion classifications provided by the algorithm are: Happy, Sad, Angry, Confused, Disgusted, Surprised, Calm, Fear, and Unknown. Each emotion classification is accompanied by a confidence score ranging up to 99.99%.

We chose to build our software around the Rekognition API, since its emotion classifications align well with existing approaches to affective intelligence in political psychology. Affective Intelligence Theory (AIT) posits that emotional responses are generated by two information processing systems – the disposition system and surveillance system – that exhibit differential effects on voters (Marcus, Neuman, & MacKuen, 2000). The disposition system, which mobilizes participation and reifies partisan preferences, comprises emotions relating to *enthusiasm* (Happy, Sad) and *aversion* (Anger, Disgust). The surveillance system, which is characterized by emotions relating to *anxiety* (Fear, Calm, Confused, Surprised), is argued to demobilize participation but encourage information-seeking (see Brader & Marcus, 2013).

Pykognition simplifies the process of classifying emotions in images with the Rekognition API. Once the researcher establishes an AWS account, they only need to insert their access tokens and an input path where the images are stored. The ImageFaceAnalysis class sends images for classification to the Rekognition API and returns the emotion classifications as a spreadsheet and rendered onto output images.

Example case: the 2020 US primary elections

To demonstrate the utility of FBAdLibrarian and Pykognition, we apply them to a case study of

candidates' emotional self-presentation in Facebook ads during the 2020 US primary elections. Our reporting here is primarily methodological. Based on our experiences in building and testing these tools, we provide a workflow for harvesting, deduplicating, and classifying images from the Facebook Ad Library. We also aim to identify the number of "core images" used in these ads, defined as the underlying image of a politician upon which text or graphics can be overlaid.

The purpose of our workflow is to deduplicate images, classify emotions in core images, and then hydrate those annotations back into the broader dataset of duplicates returned by the Ad Library API. [Figure 1](#) graphically depicts our workflow, and we detail each step in the following sections.

Step 1: collect ads data from ad library API

After verifying our identity with Facebook, we collected the ads issued by the public Facebook pages of eight primary candidates campaigning for Super Tuesday (March 3rd, 2020). These candidates include the Republic incumbent (Donald Trump) and the seven Democratic challengers who qualified for the party's debate preceding Super Tuesday (Joe Biden, Bernie Sanders, Elizabeth Warren, Amy

Klobuchar, Pete Buttigieg, Mike Bloomberg, and Tom Steyer). To query the Ad Library API, we used the Radlibrary package for R (Fraser & Shank, 2020).

We aimed to collect ads issued by these pages one month before Super Tuesday and across several Facebook products (Facebook, Instagram, Messenger, and WhatsApp). At the time of data collection, the Radlibrary package did not allow users to specify exact time ranges, so we queried the API for all ads "30 days back" from the day of data collection (March 6th, 2020).⁴ This resulted in a total of 221,136 ads.

However, we later discovered that the API collected ads going much further back, with the first starting on October 3rd, 2019. We mention this caveat because the raw numbers we report include a broader range than our intended timeframe (February 5th–March 3rd, 2020). In reporting our main results, we remove ads falling outside of these dates. We simply wish to alert readers that the raw numbers reported in our data collection and pre-processing steps include images from before February 5th ($n = 8,179$) and after March 3rd ($n = 7,336$). Based on our experience, we encourage researchers to pay special attention to the dates of ads collected from the Ad Library API, and filter

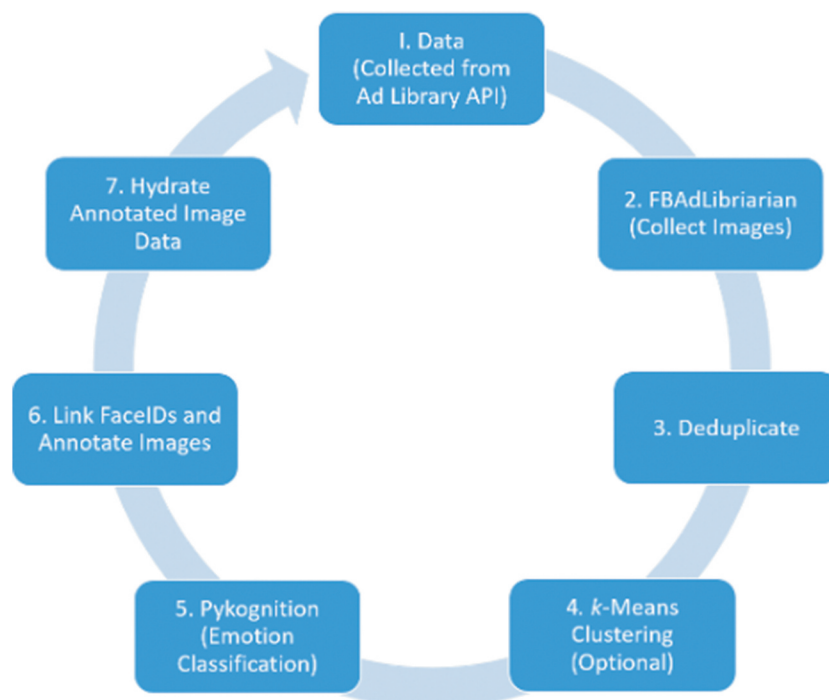


Figure 1. Workflow for collecting, classifying, and hydrating emotions data in Facebook image ads.

out those that fall outside the requested time period before using FBAdLibrarian.

Step 2: collect images using FBAdLibrarian

We then uploaded the collected 221,136 ads (as an .xlsx file including all retrievable metadata from the API) into FBAdLibrarian. After multiple iterations of testing, we reached a solution that delivered satisfactory harvesting results across Windows and Mac operating systems. As a final test run, both authors scraped the first half of the ads ($n = 110,568$) and achieved the same results for downloading images (46,192), as well as labeling the number of videos (64,373) and unknown content types (3). One author then scraped the entire dataset, resulting in the following breakdown of content types: images (90,432), videos (130,697), and unknown (7).

Users can check which ads are labeled unknown in the Librarian’s output “metadata.txt” file, which provides a hyperlink to the ad, its unique id, and its content type (image, video, or unknown). Apart from three ads that were text-only, we do not know exactly why the Librarian categorized the other four ads as unknown. Still, we consider 7 out of 221,136 ads (or $<.01\%$) an acceptable amount of data loss – especially before deduplication.

Step 3: deduplication

Upon examining the 90,432 images collected by the Librarian, we noticed the vast majority were duplicates. We therefore recommend deduplicating

images before classifying them with Pykognition, since the Rekognition API costs \$1 per 1,000 images processed.

For deduplication, we suggest using the Python script provided by Williams, Casas, and Wilkerson (2020)⁵. The script deduplicates images based on pixels, generates an output folder of unique images, and provides a spreadsheet that matches each unique image id to the ids of all of its duplicates. Using this deduplication method, we identified 1,279 unique image groups. These image groups, however, often contained the same core image but varied only slightly by text, such as changing the location of an event or the call-to-action button (e.g., from “Donate” to “Chip In”). Figure 2 shows an example of two images from Bloomberg’s page that were considered unique by the pixel-based method due to text differences. However, since the core image associated with these images is the same, the emotions displayed by the candidate do not change.

Step 4: k-means clustering

We therefore tested the performance of k -means clustering to further deduplicate the pixel-based image groups into unique core images. Similar to topic modeling, k -means clustering is an unsupervised clustering approach that attempts to group data points into a user-specified number of k -clusters. To provide input data for the clustering algorithm, we first ran each image through the convolutional neural network VGG19 (Simoyan & Zisserman, 2015). In order to use an unsupervised



Figure 2. Examples of text variants in pixel-based deduplication.

approach, we removed VGG19's classification layer and only used it to extract features as input data for the k -means algorithm.

The biggest challenge using k -means is estimating the optimal value for k . Therefore, after extracting features with VGG19, we iterated through a range of k values in an attempt to estimate the optimal number of k -clusters. We set the initial k at 650 and calculated the silhouette score (a distance measure of cluster consistency) for every k from 650 to 1,279. This method estimated the optimal number of k -clusters to be 883, which we used as the value of k in our final model.

In most cases, the algorithm clustered the same core image despite significant variations in text. As an extreme example, the following three images from the Steyer campaign were grouped into the same cluster. While the examples in Figure 3 constitute three distinct images, the core image of the candidate remains the same. Thus, we found the k -means output helpful in identifying core images, despite significant variations across the images within each cluster.

The output was not perfect, though. The largest cluster of 70 image groups was an "other" cluster, where the majority of images were infographics (41) but also images of politicians that should have been grouped with other clusters. We also found that some images were not clustered narrowly enough; two clusters of image groups could still have the same core image. Nevertheless, clustering provided a helpful labeling step to assist in the manual grouping of core images later. We prepended the cluster numbers to the names of each image in the cluster, which is useful for identifying similar core images by ordering them alphabetically in a folder.

At this stage, we sorted the deduplicated images into folders corresponding to the politician's Facebook page that issued them. During sorting, we also removed images that did not include faces. Non-facial images fell largely into four categories: infographics, merchandise, images that did not feature people, and images that featured people but not faces. In total, non-facial images constituted 220 of our original 1,279 deduplicated images. In hindsight, we recommend removing non-facial images before clustering to reduce processing time.

With the facial images sorted into folders by Facebook page, we then manually grouped similar core images into sub-folders for each page. The clustering aided in this process, since ordering the files alphabetically by cluster number revealed several coherent clusters of core images. However, since the clustering was not perfect, we still needed to finalize the sorting into core images manually.

Step 5: emotion classification with Pykognition

After sorting the deduplicated images into folders (by politician) and sub-folders (by core image), we ran all core images through the Rekognition API with Pykognition. The tool outputs a spreadsheet containing: the image name, the predicted number of faces (FaceIDs), the predicted emotion expressed by each face, and other metadata from the API that users can choose to include (e.g., age and gender).

While researchers can use this metadata to approximate the size and gender distribution of crowds, we found no immediate way to link the generated FaceIDs to specific politicians. Therefore, users can configure Pykognition to output images and automatically draw a green box around each face that labels its FaceID, emotion classification,



Figure 3. K-means clustering example.

and emotion confidence score. These boxes are helpful in linking the FaceIDs from rows in Pykognition’s output spreadsheet to specific persons in output images. Figure 4 depicts an example of Pykognition’s output images in cases when a politician is depicted alone and in a crowd.

The image on the left is an attack ad issued by the Bloomberg campaign. Trump is labeled with FaceID 1 (“FID” 1) and classified as Angry with 91% confidence. On the right, Warren poses with a crowd of supporters. Her FaceID is 4 and classified as Happy with 93% confidence. As the Warren example shows, face boxes can be difficult to interpret in images depicting crowds, since the boxes may overlap. Users can therefore adjust the emotion confidence threshold to reduce (or increase) the number of boxes drawn. Adjustments to the confidence threshold do not affect the underlying classifications; rather, they only affect how many boxes are drawn onto images.

In these examples, we drew boxes at 80% emotion confidence, which generally performed well in drawing boxes around politicians (who are often featured prominently in images) but not small, blurry faces in large crowds. Notice that no box is drawn around the supporter directly left of Warren. While this supporter is likely expressing happiness, the algorithm predicted Angry at 49% (thus falling below the 80% threshold). This classification is solely based on facial attributes (not the supporter’s clenched fist). We encourage users to experiment with different confidence thresholds and carefully compare images with the output spreadsheet, in

order to identify cases of false positives or false negatives.

Step 6: manual annotation

After classifying the deduplicated images with Pykognition, we removed all FaceIDs in the output spreadsheet not corresponding to campaigning candidates. Since this process requires going back-and-forth between spreadsheet and output images, we encourage researchers to also manually annotate images in the spreadsheet during this step.

For this case study, we applied a binary coding scheme consisting of three categories: Candidate, Opponent, and Agreement. All coding was done independently by the second author. “Candidate” was labeled if the campaigning candidate was included in the image, and “Opponent” was labeled if an opponent was in the image. These categories allow us to replicate the three categories of advertisements that Fowler, Franz, Martin, Peskowitz, and Ridout (2019) use to approximate ad tone. Using images, we can distinguish between ads featuring the candidate but not an opponent (Promote), images contrasting the candidate and opponent (Contrast), and images featuring only the opponent (Attack).

The “Agreement” category was labeled based on whether the coder agreed or disagreed with the emotion classifications from the Rekognition API. That is, emotions were not coded independently of the API’s classifications. “Agree” was labeled if the coder accepted the algorithm’s classification as

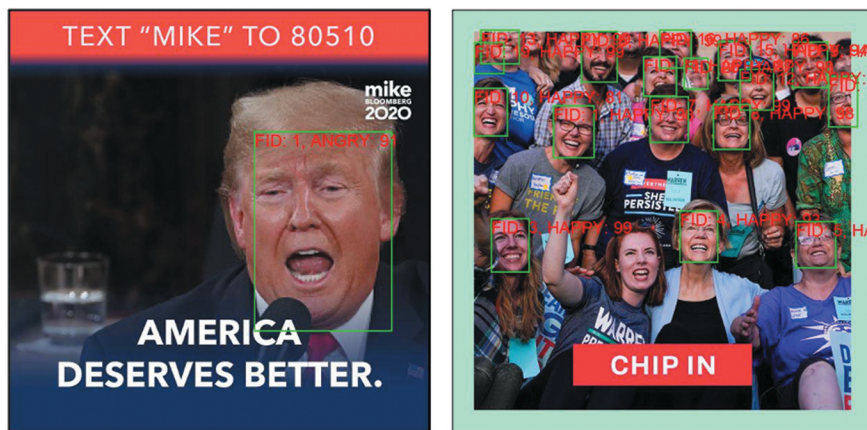


Figure 4. Pykognition image output with face boxes.

reasonable, and “Disagree” was labeled if the coder considered the classification misleading. While unorthodox, we chose this approach since analyzing each image directly alongside its classification (and confidence score) during coding allows one to scrutinize the algorithm’s performance and try to uncover patterns in its classifications. We report these observations in Appendix A (Section I), which provides examples of agreement and disagreement for each emotion classification and political candidate.

A more traditional approach would have been to code images independently of the algorithm, either using facial action coding or assigning our own interpretative labels. The former is problematic since the algorithm is likely trained on facial action coding to some extent, and therefore comparing results may boil down to how well facial action codes align between human and algorithmic judgment. Assigning interpretative classifications, meanwhile, is subjective and leaves little guidance for handling edge cases consistently.

Therefore here, we only provide an initial gauge of whether the algorithm’s classifications seemed reasonable (agree) or misleading (disagree). We did not perform strict coder reliability tests, and we acknowledge that robust, systematic verification of each emotion classification is a task for future research on a case-by-case basis.

Step 7: hydrate ad library API data with image annotations

The final step of our workflow is hydration. Once the deduplicated images have been classified, the researcher can hydrate their Ad Library API data with classifications from the Rekognition API and any researcher-added annotations. First, the metadata associated with each deduplicated image (identified by the pixel-based method) is copied to its duplicates. Then, this data can be merged with the original Ad Library API file by matching the images’ ids. In Appendix B, we provide scripts to conduct this hydration in both Python and R.

In sum, our workflow collects images from the Ad Library, deduplicates images automatically (and if desired, further into core images manually), classifies these images by emotions expressed in faces

and researcher-added annotations, and reintegrates those classifications/annotations back into the original Ad Library API corpus. With the cycle completed, we now report our results.

Results

After filtering ads returned from the Ad Library API by time period (February 5th–March 3rd, 2020), we collected 205,621 Facebook ads from the eight candidate pages. Of these ads, 84,308 (or 41%) contained images. When we deduplicated these images using the pixel-based method, we obtained 1,135 image groups, of which 910 contained faces. However, these image groups were often differentiated only by small differences in the text of the ads, so we sought to deduplicate further to the “core images” depicting politicians. When we further deduplicated into core images manually, assisted by labels generated through *k*-means clustering, we identified only 514 core images depicting people. Of these, only 434 core images depicted politicians.

Therefore, we find that relative to the overall number of image ads, the number of unique core images that depict politicians is extremely small (only .5% of all image ads). [Table 1](#) presents our results by candidate and segmented by the categories Promote (depicting the candidate) and Attack (depicting opponents). Contrast images depicting both candidates and opponents were rare ($n = 18$): only 4% of core images and .01% of overall image ads.

Our results show that for most candidates, the majority of Facebook image ads promote the candidate, aligning with Fowler et al.’s (2019) findings from the 2018 midterms elections. The Bloomberg campaign, however, was an outlier both in its large number of ads and the small proportion featuring the billionaire himself (3%). Moreover, a high proportion of Bloomberg’s image ads were attacks (39%), and the rest did not depict politicians. If we remove Bloomberg’s ads from the dataset, the percentage of Promote ads increases from 39% to 73%. Similarly, his removal from the Attack category drops the average of attack ads from 21% to 4%.

Therefore, despite Bloomberg’s large ad spends

Table 1. Ads per candidate page in promote and attack categories.

Candidate	Total		Promote Category		Attack Category			
	All Ads	Image Ads	Promote Image Ads	Percent of Image Ads	Core Images	Attack Image Ads	Percent of Image Ads	Core Images
Biden	2034	699	612	88%	28	-	-	-
Bloomberg	125339	40998	1358	3%	57	16060	39%	44
Buttigieg	15024	7844	7608	97%	59	-	-	-
Klobuchar	394	299	259	87%	8	4	1%	1
Sanders	8421	2785	2696	97%	9	-	-	-
Steyer	970	600	296	49%	102	30	5%	6
Trump	49626	29502	19335	66%	53	1632	6%	5
Warren	3813	1581	882	56%	43	12	1%	1
<i>Total</i>	<i>205,621</i>	<i>84,308</i>	<i>33,046</i>	<i>39%</i>	<i>359</i>	<i>17,738</i>	<i>21%</i>	<i>57</i>

attacking Trump, we find that attack advertising on Facebook before Super Tuesday was quite rare for image ads. Only five candidates issued attack ads with images during our timeframe and, apart from Bloomberg, their proportion was low relative to promote ads. Yet, we caution that the numbers we report do not factor in spending or impressions data. Thus, although attack ads appear low in number, they may be underrepresented in terms of how many impressions they received.

Next, Table 2 presents our agreement with the Rekognition API’s emotion classifications for core images featuring candidates ($n = 434$). Twelve images were classified as “NA” for not including facial expressions, such as when a politician was depicted from behind.

For the 422 images that could be classified, the coder agreed with the algorithm in 306 cases (73%) and disagreed in 116 cases (27%). In cases of agreement, the median emotion confidence score was 94% and the mean was 85%. For disagreement, the median was 62% and the mean was 63%. This suggests that our agreement with the algorithm typically rose with its predicted confidence score.

We offer three initial reflections on the Rekognition API’s performance. First, the algorithm performed best for Happy, which could relate

to a bias in the algorithm’s commercial applications (where companies may prioritize detecting happiness when consumers engage with their products). Second, Calm is the algorithm’s reference category, which partly explains its high presence. In the absence of any strong emotion, the algorithm may default to Calm, which may not always align with an intended projection of calmness. Finally, we noticed that certain emotions (e.g., Surprise, Confusion, Angry, and Fear) could be triggered by the candidate having an open mouth while giving a speech, leading to high disagreement rates (see Appendix A).

Finally, we report the emotion classifications for both the Promote and Attack categories across all image ads that depicted candidates (i.e., using the hydrated dataset). For each category, we present the 73% of emotions classifications where we agreed with the algorithm, and label the 27% where we disagreed as “Uncoded.” We chose not to recode cases of disagreement with human-labeled emotions, as this would introduce human classifications alongside algorithmic ones. We also find this presentation of the data helpful given our paper’s methodological focus, since it depicts the proportion of data we considered unreliable using only the algorithm’s predictions.

Figure 5 depicts the emotion classifications for the Promote category with our hydrated dataset. For the Promote category, we find Happy to be the dominant emotion followed by Calm. Surprised, Confused, and Angry appear in smaller proportions and are often artifacts of the facial expressions that candidates display in action shots while speaking.

Figure 6 shows the equivalent hydrated data for the Attack category. The size of the uncoded category is somewhat misleading. The Bloomberg

Table 2. Coder agreement with emotion classification.

Emotion	Total	Agree	Disagree	Percent Agree	Percent Disagree
Angry	40	21	19	53%	47%
Calm	146	109	37	75%	25%
Confused	10	2	8	20%	80%
Disgusted	1	0	1	0%	100%
Fear	8	3	5	44%	56%
Happy	170	151	19	89%	11%
Sad	25	8	17	32%	68%
Surprised	22	12	10	60%	40%
NA	12	-	-	-	-
<i>Total</i>	<i>422*</i>	<i>306</i>	<i>116</i>	<i>73%</i>	<i>27%</i>

*Total does not include 12 images labeled “NA”.

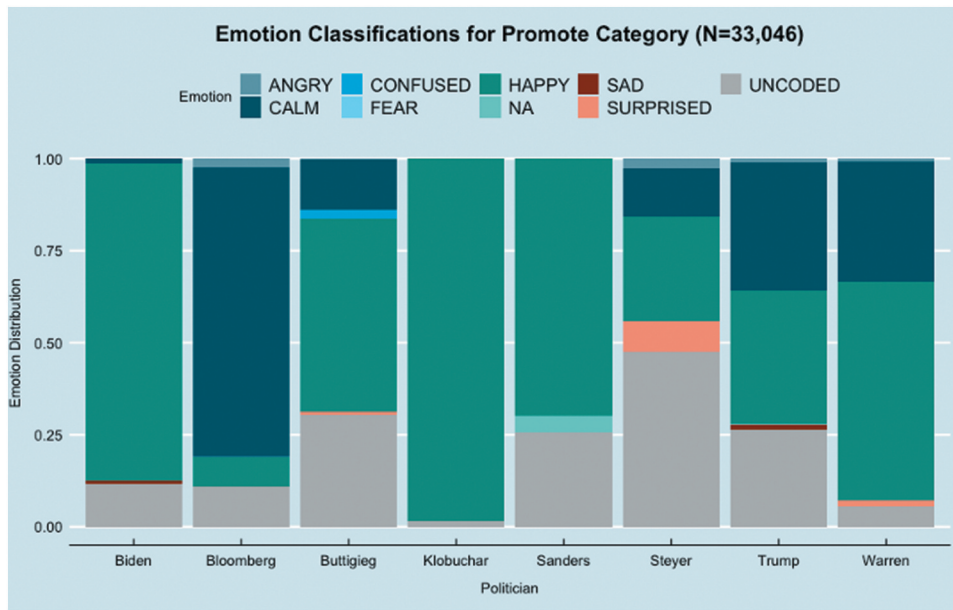


Figure 5. Emotions for the promote category.

campaign issued 14,017 ads using the same core image, which included a profile of Trump classified as Calm. We disagreed and would have labeled this image as Sad. Klobuchar only issued 4 image attack ads with the same core image, an image of Trump yelling that was classified as Calm. Steyer’s uncoded ads stem from six core images, and Trump’s only one. Had we recoded these nine core images, the overall number of Uncoded images would be drastically reduced. We include these examples in Appendix A (Section II).

Compared to the Promote category, Attacks show higher proportions of Anger, especially from Warren’s depictions of Trump. Trump’s attacks on Democratic rivals exhibit Fear and Surprise. Steyer presents Trump as Angry, Sad, and also showing Surprise. While Figure 6 highlights the limits of using off-the-shelf API classifications without human correction, it also shows promise for the tool, which identified only a small number of images that communicated happiness in attack ads.

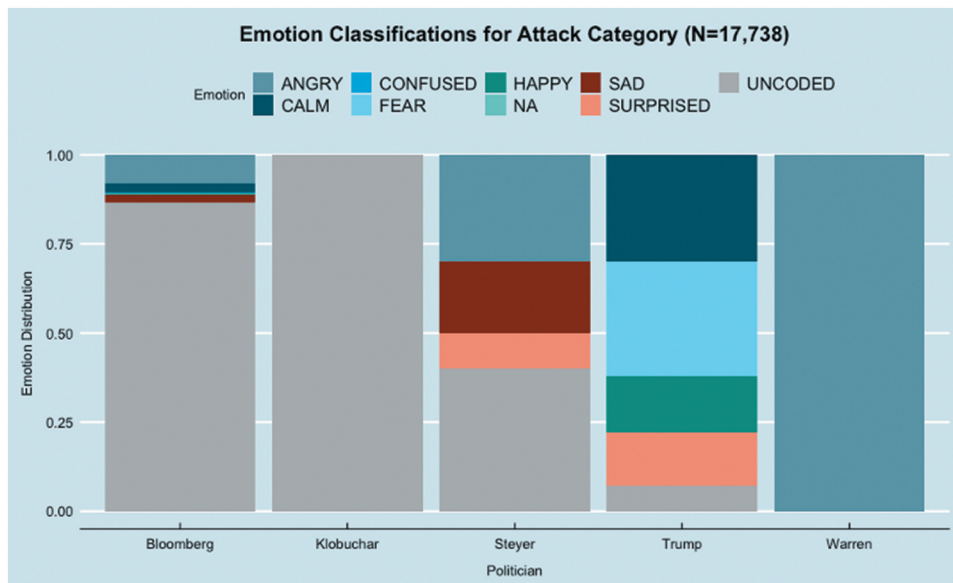


Figure 6. Emotions for the attack category.

Conclusion

This article introduces readers to the FBAdLibrarian and Pykognition, which we developed to aid in the collection and analysis of images from the Facebook Ad Library. We illustrate how the tools can be integrated into a methodological workflow, alert readers of pitfalls to avoid when using them, and hopefully provide inspiration for future research designs.

While our primary focus here has been methodological, our case study of the 2020 US primaries yields three interesting results. First, we find that despite the large number of ads that American campaigns issue on Facebook, the repertoire of core images that portray politicians is surprisingly small. Out of the over 80,000 image ads we collected, we identified only 434 core images of politicians across eight highly resourced campaigns. Second, we found that apart from the outsider Bloomberg campaign, most campaigns used Facebook image ads to promote candidates, rather than use Facebook micro-targeting to incite negative campaigning against their opponents. Third, our findings suggest that candidates communicated high proportions of happiness and calm, which is perhaps unexpected given current media narratives that cast Facebook as a medium for voter manipulation and suppression.

That said, our analysis is limited by several factors. We focused on candidate communication from their official Facebook pages. Ads from other sources like super PACs or news outlets may portray more negative emotions, and images in general election campaigns may differ from primary electioneering (Bossetta & Schmøkel, 2020). Still, our analysis provides an early test of applying industry-grade computer vision to political ad images. We consider our experience, despite its limitations, to show promise for bringing these tools into research designs and methodologies. We hope scholars will find our tools useful, apply them responsibly, and use them to push forward research on visual politics.

Notes

1. Source code: <https://github.com/schmokol/FBAdLibrarian>
2. We have not yet incorporated video downloading into the FBAdLibrarian since it slows data collection, but this feature will be added in the future.

3. Source code: <https://github.com/schmokol/Pykognition>
4. Searching by time range has been supported by the Ad Library API since version 2.4. We have now added this feature to the FBAdLibrarian. However, our tests suggest that the API yields incomplete data when querying by time range.
5. https://github.com/norawebbwilliams/images_as_data/blob/master/code/03-image-deduplication.py

Disclosure statement

The authors have no conflicts of interest to report.

Funding

This work was supported by Sweden's Innovation Agency (Vinnova) under Grant [2019-02151] (Self-FX: Self-effects on Social Media and Political Polarization).

Notes on contributors

Rasmus Schmøkel holds an M.Sc. in Political Science from University of Copenhagen, Denmark. His research interests include visual political communication, computational methods, and the role of emotions in algorithmic bias. You can follow him on Twitter @Rasmusschmokol

Michael Bossetta is Assistant Professor in the Department of Communication and Media at Lund University, Sweden. His research interests revolve around the intersection of social media and politics, including political campaigning, platform design, and cybersecurity. He produces the podcast Social Media and Politics, available on any podcast app. You can follow him on Twitter @MichaelBossetta and the podcast @SMandPPodcast.

ORCID

Rasmus Schmøkel  <http://orcid.org/0000-0001-9558-4341>
Michael Bossetta  <http://orcid.org/0000-0002-8611-5487>

References

- Barrett, L. F., Adolphs, R., Marsella, S., Martinez, A. M., & Pollak, S. D. (2019). Emotional expressions reconsidered: Challenges to inferring emotion from human facial movements. *Psychological Science in the Public Interest*, 20(1), 1–68.
- Bossetta, M. (2018). The digital architectures of social media: Comparing political campaigning on Facebook, Twitter, Instagram, and Snapchat in the 2016 US election. *Journalism & Mass Communication Quarterly*, 95(2), 471–496. doi:10.1177/1077699018763307

- Bossetta, M., & Schmøkel, R. (2020). Detecting emotions in Facebook political ads with computer vision. In D. Jackson, D. S. Coombs, F. Trevisan, D. Lilleker, & E. Thorson (Eds.), (pp. 98-99). *US election analysis 2020: Media, voters, and the campaign*. Poole: Bournemouth University. Retrieved from: <https://www.electionanalysis.ws/us/president2020/section-5-social-media/detecting-emotions-in-facebook-political-ads-with-computer-vision/>
- Bossetta, M. (2020). Scandalous design: How social media platforms' responses to scandal impacts campaigns and elections. *Social Media + Society*, 6(2), 205630512092477. doi:10.1177/2056305120924777
- Brader, T., & Marcus, G. E. (2013). Emotion and political psychology. In L. Huddy, D. Sears., & J. Levy (Eds.), *The Oxford handbook of political psychology* (pp. 165–204). Oxford: Oxford University Press.
- Facebook. (n.d.) *Facebook Ad Library API*. Retrieved from <https://www.facebook.com/ads/library/api/>
- Fowler, E. F., Franz, M. M., Martin, G. J., Peskowitz, Z., & Ridout, T. N. (2019). Political advertising online and offline. *American Political Science Review*. doi:10.1017/S0003055420000696
- Fraser, C., & Shank, K. (2020). *Radlibrary*. R package version 0.2.5. Retrieved from <https://github.com/facebookresearch/Radlibrary>
- Grabe, M. E., & Bucy, E. P. (2009). *Image bite politics: News and the visual framing of elections*. Oxford: Oxford University Press.
- Haim, M., & Jungblut, M. (2020). Politicians' self-depiction and their news portrayal: Evidence from 28 countries using visual computational analysis. *Political Communication*. doi:10.1080/10584609.2020.1753869
- Kreiss, D., Lawrence, R. G., & McGregor, S. C. (2018). In their own words: Political practitioner accounts of candidates, audiences, affordances, genres, and timing in strategic social media use. *Political Communication*, 35(1), 8–31. doi:10.1080/10584609.2017.1334727
- Leerssen, P., Ausloos, J., Zarouali, B., Helberger, N., & de Vreese, C. H. (2018). Platform ad archives: Promises and pitfalls. *Internet Policy Review*, 8(4). doi:10.14763/2019.4.1421
- Marcus, G. E., Neuman, W. R., & MacKuen, M. (2000). *Affective intelligence and political judgment*. Chicago: University of Chicago Press.
- Simoyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. Retrieved from <https://arxiv.org/abs/1409.1556>
- Williams, N. W., Casas, A., & Wilkerson, J. D. (2020). *Images as data for social science research: An introduction to convolutional neural nets for image classification*. Cambridge: Cambridge University Press.