PixelMod: Improving Soft Moderation of Visual Misleading Information on Twitter

Pujan Paudel and Chen Ling, *Boston University;* Jeremy Blackburn, *Binghamton University;* Gianluca Stringhini, *Boston University*

https://www.usenix.org/conference/usenixsecurity24/presentation/paudel-pixelmod

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

PixelMod: Improving Soft Moderation of Visual Misleading Information on Twitter

Pujan Paudel[◇]*, Chen Ling*[◇]*, Jeremy Blackburn*[♣]*, and Gianluca Stringhini*[◇]
[◇]*Boston University,* [♣]*Binghamton University*
{ppaudel,ccling,gian}@bu.edu, jblackbu@binghamton.edu

Abstract

Images are a powerful and immediate vehicle to carry misleading or outright false messages, yet identifying image-based misinformation at scale poses unique challenges. In this paper, we present PixelMod, a system that leverages perceptual hashes, vector databases, and optical character recognition (OCR) to efficiently identify images that are candidates to receive soft moderation labels on Twitter. We show that PixelMod outperforms existing image similarity approaches when applied to soft moderation, with negligible performance overhead. We then test PixelMod on a dataset of tweets surrounding the 2020 US Presidential Election, and find that it is able to identify visually misleading images that are candidates for soft moderation with 0.99% false detection and 2.06% false negatives.

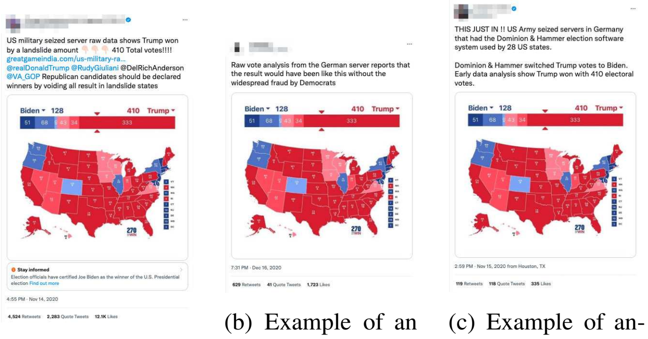
1 Introduction

Online users are constantly bombarded with false and misleading information, whether it is inaccurate and shared without malice (i.e., *misinformation*), or deliberately crafted to achieve a malicious goal, for example by state actors (i.e., *disinformation*) [71, 76]. To help their users better distinguish between accurate and misleading or false information, online platforms like Twitter have introduced *soft moderation* measures, where they add labels to posts that include inaccurate information, with the goal of providing better context to these claims. Unfortunately, little is known about how these moderation decisions are made by the platforms, and recent work found glaring issues with Twitter's soft moderation approach [15, 91].

While previous work on automated soft moderation [61] builds a solid foundation for the soft moderation of textual content, false information is not only spread through text, but images are commonly used to convey false and misleading narratives [28, 54, 88, 93]. While Twitter applies soft moderation decisions to posts containing images, the way in which this is done is not publicly known, and a preliminary analysis that we conducted shows that these labels are not applied uniformly and pervasively. For example, in Figure 1, we show three tweets that contain identical images discussing the same debunked electoral map claiming Trump's landslide victory with 410 votes after elections servers were seized in Germany by the U.S military [16].The first tweet (Figure 1a) received a warning label, while the other two tweets containing the same image (Figures 1b and 1c) did not receive any intervention, despite discussing the same misleading narrative.

These observations highlight the need for effective automated approaches to identify image content that should receive soft moderation. Moderating images, however, poses unique challenges. First, the sheer amount of data published on social media makes this a particularly challenging task, especially on platforms like Twitter, where 50 million tweets are posted every day [72]. Previous work has dealt with this problem by adapting *perceptual hashing* algorithms [28, 89, 93]. These algorithms identify visually similar images by leveraging syntactic embeddings, which are lightweight to calculate. Even if the calculation of hashes is relatively cheap, comparing an image against a large number of potential candidates is inefficient and does not scale. In addition to the computational costs of comparing images, previous research has shown that image-based false information is highly contextual and that similar images can be misinformation, satire, or even completely benign based on the context in which they are used [88, 89]. Existing approaches using perceptual hashing lack the nuance to determine this context, making them prone to false positives and unfit for the misinformation moderation use case.

Technical Roadmap. In this paper, we aim to address the limitations of previous approaches and develop a scalable, performant, and effective system able to analyze images posted on social media and identify candidates for soft moderation. We present PixelMod, an image search system designed to assist platform moderators by flagging visual content that is similar to the content they have previously applied moderation labels. PixelMod takes as seed input a list of images initially moderated by Twitter and encodes the images using percep-



[redacted]

US military seized server raw data shows Trump won by a landslide amount 👇👇👇 410 Total votes!!!! greatgameindia.com/us-military-ra... @realDonaldTrump @RudyGiuliani @DelRichAnderson @VA_GOP Republican candidates should be declared winners by voiding all result in landslide states

Stay informed

Election officials have certified Joe Biden as the winner of the U.S. Presidential election Find out more

4:55 PM · Nov 14, 2020

4,524 Retweets 2,283 Quote Tweets 12.1K Likes

[redacted]

Raw vote analysis from the German server reports that the result would have been like this without the widespread fraud by Democrats

7:31 PM · Dec 16, 2020

629 Retweets 41 Quote Tweets 1,723 Likes

[redacted]

THIS JUST IN !! US Army seized servers in Germany that had the Dominion & Hammer election software system used by 28 US states.

Dominion & Hammer switched Trump votes to Biden. Early data analysis show Trump won with 410 electoral votes.

2:59 PM · Nov 15, 2020 from Houston, TX

119 Retweets 118 Quote Tweets 335 Likes

(a) Example of a moderated tweet containing an image

(b) Example of an unmoderated tweet containing the same image

(c) Example of another unmoderated tweet containing the same image

Figure 1: Three tweets discussing a false electoral vote map showing Trump landslide victory based on false reports of seized election servers in Germany. Twitter added a warning label only to the first tweet.

tual hashing [24, 57]. To allow efficient hash comparisons, PixelMod leverages Milvus [85], a vector database that is optimized for searching among millions of hash records. After finding visually similar images, PixelMod leverages OCR to identify the context in which an image is used, allowing it to rule out false positives.

PixelMod successfully identifies both tweets in Figures 1b and 1c when queried with the image in Figure 1a. Additionally, PixelMod not only identifies images identical to the seed images but also visual variants of the images such as memes and fauxtography, which are commonly used for spreading misleading information [29, 89, 92].

Main Contributions & Findings. PixelMod addresses the limitations of previous image-based moderation approaches proposed in the academic literature, is platform-independent, and only requires a single misleading image already intervened by human moderators. With the ability to identify thousands of posts spreading misleading images, we show that an image search system like PixelMod can be used as a foundation for a large-scale soft moderation intervention system for images. While a large amount of research has been conducted on perceptual hashing and OCR, the novelty of our work lies in combining and tuning the two into a working end-to-end soft moderation system which achieves a much larger coverage than perceptual hashing alone. Besides content moderation, the takeaways offered by our work can be helpful for researchers and practitioners in building image similarity search systems for related security problems like phishing [47] and CSAM detection [22, 33].

We compare PixelMod with alternative approaches to assess visual similarity, showing that PixelMod outperforms them in terms of F1-score. In particular, the use of OCR to determine the context of false images allows PixelMod to operate using a higher similarity threshold than existing approaches only based on perceptual hashing, increasing recall while keeping precision high as well, suffering only a negligible runtime performance hit in return.

We then test PixelMod on Twitter data discussing the 2020 US Presidential Election. Starting from a seed set of 959 tweets containing misleading images, we use PixelMod to retrieve 40,244 tweets in the wild that are candidates for moderation. Performing a thorough manual analysis, we find PixelMod has a false negative rate of 2.06% and a false detection rate of 0.99%. These findings are a positive step towards building automated systems that complement Twitter's existing systems to moderate misleading images and improve the state of content moderation.

To better understand how soft moderation was adopted by Twitter during that period, we perform a qualitative measurement study of the content flagged by PixelMod, comparing it to tweets that received soft moderation labels from Twitter. We categorize the images identified by PixelMod based on Twitter's Platform Policy, finding that although different types of platform policies were moderated at different rates by Twitter, none of the policy violations exceeded a moderation rate of 5.96%.

2 Overview of PixelMod

PixelMod takes a *query image* that a platform wants to investigate for further moderation actions and retrieves images that are both visually and contextually similar. This allows a platform to quickly identify a large, high-certainty candidate list of posts for moderation. PixelMod has four stages, as illustrated by Figure 2: i) *generating* image embeddings, ii) *indexing* image embeddings, iii) *retrieving* visually similar images through approximate nearest neighbor search, and iv) *refining* contextually matching images via Optical Character Recognition (OCR).

We demonstrate PixelMod in the context of Twitter, identifying visually misleading information related to the 2020 US Presidential Election. However, PixelMod's architecture is designed to be portable to any social media platform.

2.1 Background

Identifying visually similar images given a query image is a reverse image search problem, which falls under the broader research area of Content-Based Image Retrieval systems [35]. The goal is to efficiently collect, index, and search for visually similar images over an index of millions of images, and there are several publicly available solutions (e.g., TinEye [81], Google Vision AI [32]). We conducted a preliminary exploration of these systems and identified two major limitations: 1) they require a paid subscription for programmatic access, and 2) they perform poorly for returning images from Twitter, Facebook, and other social media platforms because of platform restrictions on their crawlers. To address these

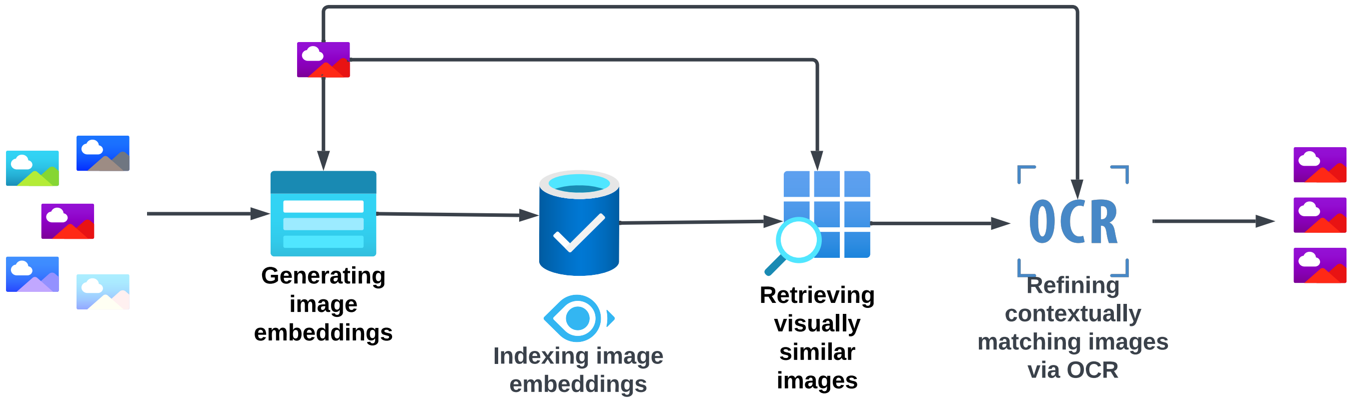
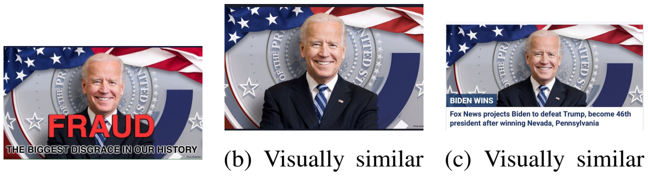

Generating image embeddings
Indexing image embeddings
Retrieving visually similar images
OCR
Refining contextually matching images via OCR

Figure 2: Overview of our image analysis pipeline.

limitations, we built a custom image search system focused exclusively on social media that collects images, generates syntactic embeddings (pHash and PDQHash), and indexes them (using the Milvus [85] vector database).

Motivating the need for multi-modality in visual similarity search. Prior works have discussed that incorporating contextual information within images is important when identifying when they have been manipulated or miscaptioned [89]. Zanettou et al. [92] discuss similar existing shortcomings of image similarity systems and the need to incorporate semantic components like OCR in image detection pipelines to effectively identify image memes. Another work studying misinformation images on WhatsApp in India [29] used semantic features like OCR from Google Cloud Vision to identify differences between different types of misinformation images (images taken out of context, photoshopped images, and memes) after retrieving images visually similar to fact-checked images. However, no existing work uses contextual information as a part of the detection system or pipeline itself.

Existing detection systems can identify images that are very similar to misleading images upon querying, but are unable to verify if the identified images are used in the same misleading context as the query image. We illustrate this case with an example in Figure 3. When we query existing image similarity systems with Figure 3a, they will typically return Figure 3b, and Figure 3c as matching results with strong confidence. However, Figure 3b, and Figure 3c are not related to the query image under question, which contains the text of “Fraud. The Biggest Disgrace In Our History”, despite being visually similar to the query. Figure 3b is an image used in a different context, with no connotation to the narrative of “Election Fraud” during the 2020 Elections, whereas Figure 3c is used in another context: Fox News projecting a win for Joe Biden. Thus, the results returned are used in entirely different context than the query image, lacking the connotation of “Election Fraud” which makes the query image misleading. A similar problem occurs when dealing with *fauxtography*, where images are manipulated with the inclusion of visual changes, overlay text, or are used out of context to mislead the user [89]. This highlights the need for incorporating context when using image similarity systems to improve misinformation detection systems.


FRAUD
THE BIGGEST DISGRACE IN OUR HISTORY
BIDEN WINS
Fox News projects Biden to defeat Trump, become 46th president after winning Nevada, Pennsylvania

(a) *Query* image

(b) Visually similar match #1

(c) Visually similar match #2

Figure 3: Example query image and results retrieved.

To address this issue, PixelMod analyzes the overlay text included in images by using optical character recognition (OCR) to better capture the context in which an image is used and then incorporate this context as a part of our image similarity pipeline. Including this context also helps overcome existing limitations of prior work using perceptual hashes that have to rely on lower similarity thresholds for matching [29, 89]. Incorporating context allows us to increase image similarity thresholds while at the same time achieving better recall without hurting precision.

2.2 Generating image embeddings

There are two types of embeddings we can extract from images for retrieval purposes. The first, *syntactic embeddings*, are fingerprints of an image’s visual features, computed such that two visually similar images will have identical, or very similar fingerprints. The second, *semantic embeddings*, capture features within the images (for example, if a given image is a chart or a portrait), and match images that are visually dissimilar, but still have similar meanings.

For our purposes, syntactic embeddings are more useful for two reasons. First, images identified through them are minor variations of the query image, making them easy to interpret and take action on. Second, semantic embeddings use complex deep learning architectures [37, 73], while syntactic embeddings are relatively fast and cheap to train and use. This means we can scale to millions of images with relatively few resources (e.g., PIXELMOD does not require a GPU.)

We generate our syntactic embeddings via *perceptual hashing*. We make use of two different *perceptual hashing* embeddings: i) pHash [57], and ii) PDQHash [24]. pHash encodes images as 64-bit vectors by extracting geometry-preserving feature points using Discrete Cosine Transform (DCT). The resulting vectors are invariant to simple image transformations and minor coloring differences. PDQHash is an improvement over pHash, encoding images as 256-bit vectors. Prior work studying the spread of misleading images in social media has used both pHash and PDQHash [41, 44, 66, 89, 93]. We evaluate pHash and PDQHash with respect to precision, and recall, as well as the time taken to generate an embedding, index it, and eventually query the index.

2.3 Indexing image embeddings

The most similar image to a query is the one whose embedding is the shortest distance from the query image's embedding. While we could perform a pairwise comparison with the query image and all candidate images, this is intractable at the scale at which social media platforms operate. To address this, there are specialized vector databases that can be used to perform queries over embeddings. For PIXELMOD, we chose to use Milvus [85] a purpose-built vector management system. Milvus abstracts away lower-level details (e.g., index creation and item insertion) via a higher-level API and Facebook's Faiss [43] library for efficient dense vector similarity search.

Milvus has two types of indexes: i) `BIN_FLAT` and ii) `BIN_IVF_FLAT`. `BIN_FLAT` guarantees a 100% recall rate by exhaustively searching for matches to a query embedding. `BIN_IVF_FLAT` is quantization-based, dividing embeddings into multiple cluster units and comparing the query embedding to the center of each cluster. `BIN_IVF_FLAT` is faster than `BIN_FLAT` but requires tuning hyperparameters to find an optimal recall rate. Milvus recommends using `BIN_FLAT` for datasets in the order of a few million vectors, which fits the datasets we use to evaluate PIXELMOD

2.4 Retrieving visually similar images

Vector databases like Milvus scale by leveraging approximate nearest neighbor search across their indices. Milvus determines the nearest neighbors by scoring candidates with a similarity metric. We use Hamming distance, leaving us with the next challenge of choosing an appropriate threshold to treat as "visually similar." We select Hamming distance over cosine similarity as prior works on perceptual hashing have predominantly used this metric to assess visual similarity on both PHash and PDQHash (e.g. baseline systems [89] and [92]). Additionally, we index the perceptual hashes on Milvus as binary hashes, and using Hamming distance as the similarity metric allows us to efficiently compare the binary representations of images. Previous works using pHash to study misinformation on a variety of social media platforms have recommended a Hamming distance of six [89], and eight [92]. The developers of PDQhash suggest a distance of 32, which has been used to study misinformation on WhatsApp [44, 66]. Other work using PDQHash has used a threshold of 40 to study the spread of COVID-19-related media in Pakistan over WhatsApp [41]. Ultimately, choosing a threshold depends on the dataset used and the particulars of how "visual similarity" is defined. In Section 4, we experimentally evaluate the choice of the visual similarity threshold (θ_{visual}) by optimizing over both precision and recall.

2.5 Refining matched images using OCR

The final component of PIXELMOD involves processing the results retrieved as "visually similar" to the query image by comparing text extracted via an Optical Character Recognition (OCR) engine. For the rest of the paper, we refer to the output text from an image extracted from an OCR engine as OCR label. The advantages of applying this OCR-driven post-processing to the matched results are two-fold: i) it eliminates the limitations of underlying perceptual hashing algorithms by filtering out their false positives and ii) it allows us to explore relatively larger distance thresholds for visual similarity matching, thus allowing us to minimize false negatives. Prior applications of perceptual hashing algorithms for identifying visually similar images have used conservative thresholds [41, 44, 89] to minimize false positives. In contrast, we aim to exploit larger thresholds to increase our recall as well as have a more tuneable mechanism with respect to precision trade-offs. In Section 4, we first validate the choice of Google Cloud Vision API as the underlying OCR engine [32], and experimentally evaluate the calibration of the similarity metric for the extracted OCR labels as our OCR engine) and corresponding threshold for similarity ($\theta_{textual}$).

3 Datasets

We use three different datasets to evaluate PIXELMOD (summarized in Table 1). Our first dataset, $\mathbf{D}_1$, contains 7.6M tweets collected via the Twitter Streaming API using a set of 2020 US Election hashtags (e.g., #ballotfraud, #voterfraud, #electionfraud, #stopthesteal) curated by Abilov et al. [4]. To comply with Twitter's terms of service, this dataset only provides tweet IDs, therefore, we need to retrieve the full tweet content from the Twitter API through a process called *hydration*. Out of the 4,017,259 tweets we were able to rehydrate, about 218K have at least one "media" entity: either an image or a video (tweets with videos also contain a thumbnail image, which we include in our dataset).

Dataset	# Tweets with Images
VoterFraud Dataset (**D**$_1$)	217,868
1% Twitter Stream (**D**$_2$)	12,560,319
Twitter Context Annotations (**D**$_3$)	6,952,300

Table 1: Overview of our dataset.

Our second dataset, **D**$_2$, consists of 13M tweets with media collected via the 1% Twitter stream from November 1 to December 31, 2020. Unlike **D**$_1$, we do not filter any keywords, therefore **D**$_2$ is a uniform 1% sample of Twitter.

Finally, we collect **D**$_3$ via Twitter's Academic Research full-archive search endpoint in late 2022, by querying *context annotations* [62] associated with the 2020 US Elections. Twitter annotates tweets with context annotations by semantically analyzing their content and metadata, categorizing them into a nested structure of domains and entity labels. The more than 80 domains cover things like politics and TV shows, while the nearly 145K entities cover details of the domains ranging from elections to festivals, to politics and media personalities.

We identify two context annotations related to the 2020 election by retrieving the context annotations of all tweets from **D**$_1$: 1) "2020 US Election Day" and 2) "2020 US Presidential Election," both in the *Events* domain. We also found other context annotations from the tweets, but they were associated with "Political figures" and "Politics" in general, not the 2020 elections specifically. To minimize the noise that can be introduced when using such a broad context, we do not include those annotations when building **D**$_3$. In the end, **D**$_3$ consists of the 6.9M tweets that we retrieve using the "2020 US Election Day" and "2020 US Presidential Election" context annotations and limiting our search to November 1st, 2020 to December 31st, 2020.

Index building. We build two different Milvus indices for pHash and PDQHash embeddings, using all the images in our datasets. The index size for both types of embeddings is 19.7M, after combining all the tweets from **D**$_1$, **D**$_2$, and **D**$_3$.

4 System Validation

In this section, we present the validation strategy for PixelMod. First, we discuss how we build a ground truth dataset. Then, we describe our experimental setting to determine the best hashing mechanism, and the best string similarity algorithm for the OCR labels, and the corresponding θ_{visual} and $\theta_{textual}$ to use with the hashing algorithms and string similarity algorithms. The goal is to experimentally determine the best operating values for PixelMod by optimizing both precision and recall for the overall system.

Building ground truth. First, we randomly sample 50 images from **D**$_1$. For each of these 50 images, we query both the pHash and PDQHash indexes for similar images, limiting the results to the maximum threshold for each algorithm (10 and 90). We select 90 as the maximum threshold for PDQHash as its developers experimentally verified it as the upper bound for images that are known to be different. For pHash, we select 10 as the maximum threshold because previous work has found that higher thresholds produce results that are too noisy [92]. This results in 11,825 unique images retrieved as similar across both indices for the same set of 50 *query images*. Then we manually annotate the results using a pairwise image annotation tool developed by [26]. Note that the goal of this annotation was to verify that images were *visually* and *contextually* similar to the original one (i.e., their overlay text contained similar words). For this reason, we did not need to build a codebook and have multiple annotators agree on the results as we did for more subjective experiments later in the paper. In the end, we find 9,785 images that are similar to our 50 source images. We refer to these 9,785 images as $\mathbf{GT}_{viz}$.

Determining θ_{visual}. Next, we use the images in $\mathbf{GT}_{viz}$ to determine the accuracy of the results returned by the two hashing algorithms when using different thresholds for θ_{visual}. Our Milvus indexes return the closest set of embeddings for a query image, scored by the Hamming distance to the candidate image. For pHash, we experiment with a range of Hamming distance thresholds from *4* to *10*, which is in line with previous works that found an optimal threshold between *6* [89] and *8* [92]. For PDQhash, on the other hand, it is recommended to use a range of pairwise distances for identifying similar images [24]. Therefore, we test multiple threshold ranges, some used by previous work using PDQHash [41, 44, 66] (*32*, *48*) as well as three additional ranges to take us up to the maximum possible threshold (*64*, *80*, and *90*).

Validating Google Cloud Vision API for OCR. To validate the ability of Google Cloud Vision API to correctly extract the OCR labels in a query image, we sample 50 images from $\mathbf{GT}_{viz}$ and create a ground truth of the text contained in the images. These are examples of images occurring "in the wild," which contain text in specialized fonts, small text in lower quality images, or text in artistic fonts, and are therefore a good test for the OCR component of PixelMod. On this dataset, the median Jaccard similarity of the ground truth text and the one extracted by Cloud Vision OCR is 1.0, and the mean is 0.95. This validates that Google Cloud Vision API can be successfully used as the underlying engine for identifying the contextual text contained in the images of PixelMod. Upon doing some error analysis of the OCR text, we find that the output of the Cloud Vision API is not missing any text present in the images, but is sometimes parsed in a different order than the ground truth. This is an artifact of how the OCR engine works on different regions of the image. We argue that this would not pose a problem when using the system for PixelMod as the method will work consistently across all the source images and the potential matches retrieved through perceptual hashing.

Determining $\theta_{textual}$. After validating that Cloud Vision API can be successfully used to extract the contextual text contained in images, we aim to determine the appropriate text similarity metric and corresponding similarity threshold to compare the OCR labels of the retrieved visual matches with the *query* images. We first check if the *query* image contains any OCR label, which we call $label_{query}$, using Google Cloud Vision API. If $label_{query}$ is not empty, then, for each visually matching image, we compute the corresponding OCR labels ($label_{match}$) using the same API endpoint. We experiment with multiple text similarity metrics, and multiple similarity thresholds to assess if $label_{match}$ is similar to $label_{query}$. The similarity metrics we experiment with are: i) Normalized Levenshtein similarity, ii) Jaro-Winkler similarity, iii) similarity metric based on Longest Common Subsequence (Metric LCS), and iv) Jaccard-index similarity. For each similarity metric, we measure F1 scores on the similarity threshold range of $0, 0.05, 0.1, 0.15, \ldots, 0.75, 0.8$ Note that, for Jaccard index, we experiment with values of n-grams ranging from 1 to 5 as this algorithm converts strings into a set of n-grams when computing the similarity.

Grid search setup for θ_{visual} and $\theta_{textual}$ to determine the best operating values for PixelMod. To identify the best set of components for PixelMod, we experimentally determine the best hashing method, the corresponding distance threshold for that hash, the similarity metric for the OCR label, and accordingly the similarity threshold for comparing $label_{query}$ and $label_{match}$. We perform a grid search over these four different components of PixelMod, scoring the combinations of the components by their corresponding F1-score, evaluated on **GT**$_{viz}$. We present the results of the grid-search experiment in Figure 4. For space reasons, we only present the best-performing text similarity method for each θ_{visual} and the two hashing methods. We analyze how the F1 score of the embeddings with θ_{visual} changes as we increase the $\theta_{textual}$.

We can see that the combination of PDQHash with θ_{visual} of 90, and the OCR component using Jaccard similarity ($ngram$ = 4) with $\theta_{textual}$ of 0.05 produces the best F1 score of 0.980. This setting yields a precision of 0.990 and a recall score of 0.979. The next best performing metrics are PDQHash with θ_{visual} of 80 and pHash with θ_{visual} of 10 using normalized Levenshtein similarities. We observe the immediate returns of expanding the θ_{visual} to the maximum bound to retrieve as many relevant results as possible, without compromising on the false positives. We will further evaluate this configuration with previous state-of-the-art image detection methods in Section 5.1. We also note that we can improve the performance of pHash embedding by using a wider θ_{visual} of 10, compared to the thresholds of 6 and 8 used in the prior works.

In the rest of the paper, we set PixelMod to use an image embedding of **PDQHash** with a θ_{visual} of 90 and an OCR post-processing component of Jaccard similarity ($ngram$ = 4) with $\theta_{textual}$ of 0.05.

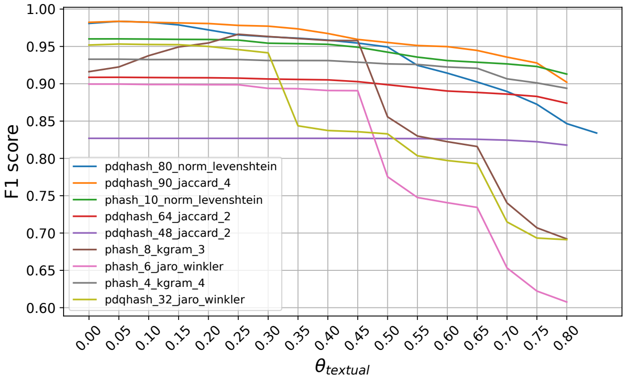


Figure 4: F1 score of different grid search components.

5 Evaluation

In this section, we first compare PixelMod with various other image similarity systems as baselines, showing that PixelMod is the best-performing approach for content moderation. Next, we use PixelMod to identify misleading images during the 2020 US Presidential Election in the wild and compare its effectiveness with the existing moderation system of Twitter. Finally, we perform a qualitative characterization of the misleading images by analyzing the images through the lens of Twitter's platform policies in detail.

5.1 PixelMod vs. baselines

We compare the performance of PixelMod against various image similarity systems as a baseline for evaluation. We use four different families of image similarity systems for comparison. First, we compare PixelMod against semantic embeddings that use Deep Neural Networks (DNNs) based methods. These include features extracted from final layers of DNN architectures specializing in image classification such as Inception-v3 [79] and ResNet [37]. The features extracted from these methods have been shown to be useful for visual search and similarity tasks [7, 74]. The next family of methods that we compare PixelMod against is image descriptors, which can be considered as "deeper syntactic hashes" [24]. This set of methods is popular in computer vision tasks such as object recognition, scene recognition, and work by extracting a large number of *keypoints* in an image. Matatov et al. [53] use ORB descriptors for identifying the spread of visual misinformation to assist journalists. We use ORB [70], SIFT [51], and DAISY [82] as image descriptors. Another family of methods we compare PixelMod with is the list of prior works that have used perceptual hashing algorithms for identifying images spreading misinformation. These include Phash and PDQHash with multiple distance thresholds [41, 44, 89, 92].

Finally, we evaluate the performance of PixelMod with

Method	Prec.	Rec.	F1	Runtime
Inception v3 [79]	0.679	0.878	0.766	0.031s
ResNet-50 [37]	0.518	0.938	0.667	0.027s
ResNet-101 [37]	0.617	0.932	0.742	0.030s
ResNet-152 [37]	0.440	0.962	0.604	0.034s
ORB descriptors [70]	0.491	0.535	0.512	0.040s
SIFT descriptors [51]	0.935	0.0136	0.026	0.253s
DAISY descriptors [82]	0.484	0.431	0.456	0.257s
PDQHash (thr. 32) [44]	0.992	0.798	0.885	0.020s
PDQHash (thr. 40) [41]	0.975	0.838	0.901	0.020s
Phash (thr. 6) [89]	0.995	0.596	0.746	0.017s
Phash (thr. 8) [92]	0.991	0.707	0.826	0.017s
Inception v3 [79] + SentenceBERT [65]	0.711	0.972	0.820	0.076s
CLIP (tweet text) [64]	0.814	0.787	0.800	0.149s
CLIP (tweet text + OCR) [64]	0.862	0.810	0.835	0.149s
CLIP (OCR) [64]	0.883	0.828	0.855	0.149s
PixelMod	**0.990**	**0.979**	**0.980**	0.223s

Table 2: Comparison of PixelMod with baselines.

multi-modal embeddings. While there is a rich body of work and systems existing in the vision-language literature, the majority of these works are trained with an objective of explaining the visual concept contained in the image [50]. On the other hand, OCR texts occur as semantic context to the image itself and can be much longer, noisy, and nuanced. Thus it is not possible to directly adapt models such as Universal Image Text Representation Learning (UNITER) [18], or Visual-BERT [49] that have been trained on vision-language tasks for our problem, in the same way adapting Inception-v3 or ResNet is possible. However, to assess the potential of a multi-modal embedding that leverages text from an OCR engine, or text from the accompanying tweet with DNN-based features, we experiment with "concatenated" vision-language embeddings. First, we concatenate the embeddings obtained from Inception-V3 (a 2048-dimensional vector) and Sentence-BERT [65] (a 768-dimensional vector) to create a joint embedding from the two modalities.

We experimented with two different techniques for unifying the embeddings: i) concatenating, and ii) stacking, further reporting the best results. While much more sophisticated mechanisms of combining embeddings such as attention mechanisms exist, the dataset size of **GT**$_{viz}$ limits us in evaluating methods that combine embeddings or modalities. Fully leveraging multimodal embeddings requires an end-to-end deep learning training setup, and accordingly, large-scale datasets designed for the problem of soft moderation, which currently does not exist.

In the same way, we also combine textual and visual embeddings using the CLIP model from Open AI [64]. CLIP consists of a text and an image encoder which encodes textual and visual information into a multimodal embedding space by using contrastive learning. The model is trained on various text/image pairs from Web sources, and has demonstrated impressive zero-shot capabilities for classification purposes and has been used to detect hateful content on social media [31, 63]. Alongside the OCR text, we also experiment with using the tweet text as part of the multi-modal embedding to evaluate if using the tweet text as further context can improve model performance for image retrieval. To this end, we concatenate the visual information encoded through the image encoder with three different variants of textual information: i) tweet text, ii) tweet text + OCR text, and iii) OCR text. After extracting the joint embeddings using CLIP, we further normalize the multi-modal embeddings for evaluation.

For the neural network-based image embeddings (Inception-v3, ResNet-50, ResNet-101, and ResNet-152) we experiment with multiple resolutions of input (1x, 2x, etc.), reporting the best performing F1 score for each method. Similarly, for the neural network-based image embeddings and the multi-modal embeddings, we experiment with multiple similarity thresholds $(0, 0.05, 0.1, 0.15, \ldots, 0.75, 0.8)$ and report the best performing score for each method (except for the methods that use perceptual hashes with pre-determined similarity thresholds). For the image descriptors, we experiment with a different number of features (30, 60, 90, and 120) and report the best-performing F1 score for each method. We compare these methods on **GT**$_{viz}$.

The results of the comparison of PixelMod with the baselines are presented in Table 2. Image descriptor methods perform poorly when used for soft moderation. This is because these approaches are designed for higher-level tasks like object and scene recognition. Deep neural network approaches work well in identifying the subjects of moderated images (e.g., Donald Trump or Joe Biden) and therefore report a high recall. At the same time, however, their precision is low, because they flag any image containing the same subjects as similar. We find that leveraging multimodal embeddings slightly improves the performance over DNN-based methods, but still has a very low precision (0.711) for Inception-v3+SentenceBERT. In the same way, using concatenated embeddings from Open AI's CLIP significantly increases the precision over single-modality DNN methods, but has a very low recall (0.828). Among the three different variants of CLIP embeddings leveraging different channels of modalities, we find that encoding the simplest channel, i.e. CLIP (OCR) has the best performance. Surprisingly, encoding tweet text alongside the OCR text, i.e., both CLIP (tweet text) and CLIP (tweet text + OCR) does not improve the performance of the CLIP model. This can be attributed to how state-of-the-art multimodal embeddings like CLIP are designed to match image and caption pairs, and therefore fail to capture the nuances of text and media co-usage in social media. Perceptual hashing approaches, on the other hand, identify visually similar images and therefore report a high precision. At the same time, the need to exclude images that are visually similar but contain text that is contextually different forces these approaches to use low similarity thresholds, which limits their recall.

PixelMod overcomes the limitations of all three types of approaches. The use of perceptual hashing allows our approach to have better precision than neural network-based ones. At the same time, the use of OCR to determine the context of an image allows PIXELMOD to operate at a higher threshold than existing systems based on perceptual hashing, addressing the low recall reported by these methods. While PIXELMOD's precision is slightly lower than the best-performing perceptual hash method (0.990 vs. 0.995), its recall is the highest among all tested approaches. As a result of this, PIXELMOD reports the best F1-score among the tested approaches, balancing false negatives and false positives better than previous work.

We also compare the runtime of different systems in Table 2. For each method, we report the combined time of generating the image embeddings and indexing those embeddings on Milvus, averaging over 5 independent runs on identical system load. The time for retrieving visually similar images from Milvus is not considered as Milvus optimizes the retrieval time across all embeddings of different sizes (average of 0.27 seconds) for all of the systems. We can observe that the runtime of PIXELMOD is around 8 times slower than DNN-based methods, and about 10 times slower than perceptual hashing-based methods. Computing the OCR label of an image is the most time-consuming operation of PIXELMOD compared with other methods as it takes an average of 0.223 seconds per image. While this is a large overhead incurred by PIXELMOD, we need to keep in mind that its OCR component is only triggered when a similar image to a seed one is identified within the threshold of θ_{visual}, which only occurs for 0.973% of the images in $\mathbf{GT}_{viz}$. When there is no match, the time overhead of PIXELMOD is the same as PDQHash. Therefore, we argue that this slowdown of 0.2 seconds every 100 images on average is an acceptable trade-off, allowing PIXELMOD to improve its F1-score by 8% over the second best performing algorithm and allowing for more comprehensive soft moderation.

To further check the impact of the OCR component of PIXELMOD, we examine how the latency of OCR changes with increasing amount of text in the images. We characterize the amount of text by using the percentage of image area covered by it. Figure 5 shows a scatter plot of the OCR runtime against the percentage of image covered by text in $\mathbf{GT}_{viz}$. The Pearson's correlation coefficient (r) between the percentage of image covered by text and runtime is 0.401, suggesting a moderate positive correlation between runtime and the amount of text contained in an image. This moderate correlation indicates that the fraction of image covered by text plays a role in the OCR runtime.

5.2 Detecting Visual Misleading Information on Twitter using PIXELMOD

In this section, we evaluate PIXELMOD in the wild to further identify images that, while spreading false information, were not moderated by Twitter. This showcases the utility of our approach while also highlighting the serious need for scalable and automated techniques to identify visually misleading information. We first build a set of seed images spreading misleading information, using the 2020 US Presidential Election as a case study. We then apply PIXELMOD to find more images that are similar to the seeds that *should have been* moderated but were not. Finally, we perform a thorough qualitative analysis of the types of images detected, providing a characterization of image-based misinformation in the context of the 2020 US Presidential Election and how they align with Twitter's platform policies.

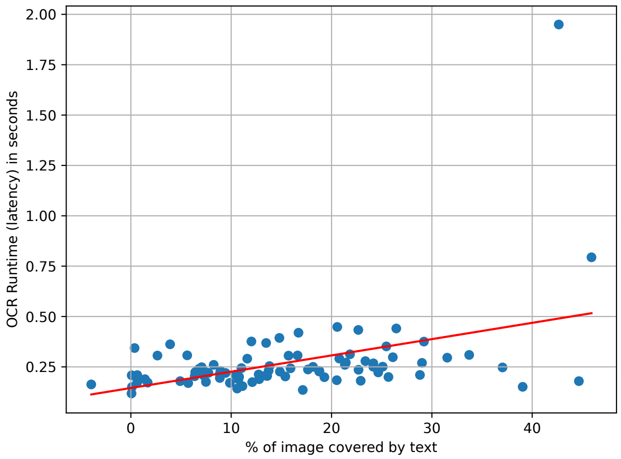
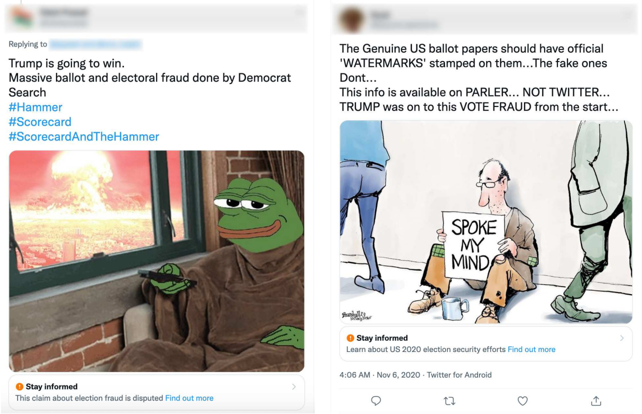


Figure 5: Latency of OCR by changing the percentage of text covering images in $\mathbf{GT}_{viz}$.

5.2.1 Identifying and Filtering 2020 US Presidential Election misleading images

In this section, we first describe how we select a dataset of images that received soft moderation from Twitter in the context of the 2020 US Presidential Election. We then present the results reported by PIXELMOD when looking for similar images in the wild and compare these results with the coverage of the moderation applied by Twitter.

Building a seed set of misleading images. To build a seed set of misleading images contained in tweets that were moderated by Twitter, we use datasets $\mathbf{D}_1$ and $\mathbf{D}_3$. Following the methodology of [91], we check the moderation status of all tweets in $\mathbf{D}_1$ during late 2022 and find 1,133 tweets that were moderated by Twitter. Unfortunately, $\mathbf{D}_3$ is too large (6.9M tweets) for us to check the moderation status of each tweet, given the limitations imposed by the Twitter API. Therefore, we only check for the tweets containing an image that occurred more than 5 times in $\mathbf{D}_3$, based on their PDQhash, resulting in 542,700 candidate tweets. From these, we identified a further 3,134 tweets with images that were moderated. Combining the moderated tweets from these two sources, we



[redacted]

Replying to [redacted]

Trump is going to win.
Massive ballot and electoral fraud done by Democrat
Search
#Hammer
#Scorecard
#ScorecardAndTheHammer

Stay informed
This claim about election fraud is disputed Find out more

[redacted]

The Genuine US ballot papers should have official
'WATERMARKS' stamped on them...The fake ones
Dont...
This info is available on PARLER... NOT TWITTER...
TRUMP was on to this VOTE FRAUD from the start...

Stay informed
Learn about US 2020 election security efforts Find out more

4:06 AM · Nov 6, 2020 · Twitter for Android

(a) Comfy Pepe meme used as media on a tweet.

(b) A cartoon illustration used as media on a tweet.

Figure 6: Moderated images irrelevant to the 2020 election.

end up with a final seed set of 4,267 images.

An issue that we face at this stage is that our dataset contains tweets that were moderated by Twitter, but we do not know if the moderation decision was made based on the image included in the tweet or based on the text. In fact, it is rather common for social media users to include generic images (e.g., memes or GIFs) in their posts that do not contain false information themselves. Two examples of these types of tweets are presented in Figure 6. Neither image conveys false information, but the text is clearly supporting false claims of election fraud. If we kept these images in our seed dataset, PIXELMOD would flag many unrelated tweets that just happen to include them, greatly reducing its effectiveness.

To avoid this issue, we manually annotate the 4,267 tweets with images that were moderated by Twitter, discarding those not directly related to false claims about the 2020 US Presidential Election. We first sample 200 images from the 4,267 seed images and have two annotators independently label them as being relevant or not. The annotators then discuss their results, finding that 186 out of 200 images were relevant with nearly perfect agreement (Cohen's Kappa score of $\kappa = 0.9177$ [55]). This indicates that an image's relevance to the 2020 US Election is clearly understood between the two annotators. We then split the remaining 4,067 seed images between them and wind up with a total of 959 relevant images.

Detecting misinformation images using PIXELMOD. Finally, we use PIXELMOD to retrieve more tweets from the index that contain similar images to the seed misleading images. To this end, we initially retrieve 50,439 images visually similar to the 959 seed misleading images, out of which 40,244 are further filtered to be contextually similar to the query images, and hence candidates for moderation.

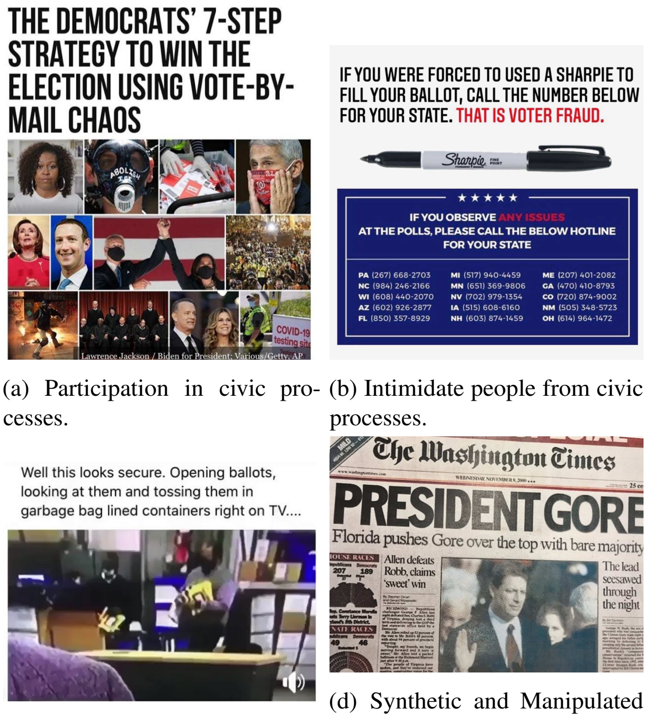
To check for the accuracy of these moderation candidates identified by PIXELMOD, we perform a false positive and false negative analysis. First, we randomly sample 50 *query images* from the 959 seed misleading images. Since the 50 *query images* are already manually verified to contain misleading images related to the 2020 US Presidential Election, we can assume that visually and contextually similar images to the images will be misleading images as well. We retrieve and annotate all the images from our index visually similar to the 50 *query images* following the similar methodology used to build $\mathbf{GT}_{viz}$ in Section 4, resulting in a reference set of 10,387 images. Upon querying PIXELMOD with the same 50 *query images*, it retrieves 10,172 images both visually and contextually similar to the *query images* as moderation candidates. We find that 96 of these detected images are both visually and contextually dissimilar to the corresponding *query images*, resulting in a false detection rate of 0.99%. This is due to the well-known limitations of perceptual hashing algorithms [92], for example when dealing with images where a solid color background dominates. On the other hand, we find that PIXELMOD has a false negative rate of 2.06% as it could only successfully detect 10,172 images out of the possible 10,387 images. Based on these results, PIXELMOD performs a lot better than Twitter's internal soft moderation approach, which moderated only 521 of those 10,387 images (94.98% false negatives).

Finally, we check if the candidate tweets containing the images identified by PIXELMOD were also moderated by Twitter. Following the methodology in [91], we extract the relevant metadata related to soft moderation interventions for the tweet. Out of the 40,244 tweets checked, we find that only 2,950 tweets received soft moderation interventions by Twitter. We were unable to check the intervention-related metadata for 2,479 tweets as they were inaccessible during the time we conducted the study. This analysis indicates that Twitter's soft moderation approach is inadequate, with a large proportion of tweets (92.66%) containing misleading images left without moderation, and it showcases the utility of automated approaches like PIXELMOD.

5.2.2 Characterizing misleading images about the 2020 Presidential Election on Twitter

In this section, we perform an in-depth analysis of the images identified by PIXELMOD. To aid human annotation, we first group visually similar images into *Image Stories*. We then code the extracted *Image Stories* according to Twitter's platform content policies [83, 84], with the goal of understanding how the detection performed by PIXELMOD aligns with the platform's existing moderation guidelines. Finally, we investigate whether misleading images violating different types of content policies get moderated differently by Twitter.

Aggregating misleading images into *Image Stories*. To help manually code the images detected by PIXELMOD, we aggregate them into *Image Stories*. We define an "Image story" as a set of similar images that convey the same misleading message in the same context. For example, the three images


THE DEMOCRATS' 7-STEP STRATEGY TO WIN THE ELECTION USING VOTE-BY-MAIL CHAOS
Lawrence Jackson / Biden for President; Various/Getty, AP
IF YOU WERE FORCED TO USED A SHARPIE TO FILL YOUR BALLOT, CALL THE NUMBER BELOW FOR YOUR STATE. THAT IS VOTER FRAUD.
Sharpie FINE POINT
IF YOU OBSERVE ANY ISSUES AT THE POLLS, PLEASE CALL THE BELOW HOTLINE FOR YOUR STATE
PA (267) 668-2703
NC (984) 246-2166
WI (608) 440-2070
AZ (602) 926-2877
FL (850) 357-8929
MI (517) 940-4459
MN (651) 369-9806
NV (702) 979-1354
IA (515) 608-6160
NH (603) 874-1459
ME (207) 401-2082
GA (470) 410-8793
CO (720) 874-9002
NM (505) 348-5723
OH (614) 964-1472
Well this looks secure. Opening ballots, looking at them and tossing them in garbage bag lined containers right on TV....
The Washington Times
PRESIDENT GORE
Florida pushes Gore over the top with bare majority
Allen defeats Robb, claims 'sweet' win
The lead seesawed through the night

(a) Participation in civic processes.

(b) Intimidate people from civic processes.

(c) Outcomes of civic processes.

(d) Synthetic and Manipulated Media.

Figure 7: Example images violating four major categories of Twitter's Civic Integrity Policy.

included in three different tweets in Figure 1 are visually similar and should be grouped together. To this end, we follow a grouping approach similar to the one adopted by Zannettou et al. when studying image memes [92].

First, we apply clustering to the PDQhash embeddings of the images detected by PIXELMOD to group visually similar images by using the DBSCAN clustering algorithm [27]. To use the DBSCAN clustering algorithm, we need to select parameters for two parts of the process: i) distance threshold for assessing similarity, and ii) minimum number of elements in a cluster. Through the experimental validation performed in Section 4, we have already established that using PDQhash as the syntactic embedding, with a θ_{visual} of *90* is best suited for identifying visually similar images. We use this empirically informed distance threshold as the Hamming distance difference threshold for the first clustering parameter and set the minimum number of elements in a cluster to be 1, to allow for misleading images that occur in isolation, without visual variants. Upon using the DBSCAN algorithm to cluster the 40,244 images detected by PIXELMOD, we obtain 258 clusters of images. These image clusters represent the misleading images aggregated into visually similar claims conveying the same misleading message.

Analyzing misleading images through the lens of Twitter's Platform Policies. We present our codebook, which guides our annotation process for characterizing *Image Stories* used during the 2020 Presidential Election. We develop the codebook such that it aligns with the platform policies of Twitter [83, 84]. We use two different policies outlined in Twitter's *Platform Integrity and Authenticity* resources: i) Synthetic and manipulated media policy [83] and ii) Civic Integrity Policy [84]. Each policy contains multiple categories of violations, and each category is broken down into multiple rules. The Civic Integrity Policy outlines how "Twitter should not be used for the purpose of manipulating or interfering in elections or other civic processes such as refereed, censuses." We include 3 different categories from the Civic Integrity Policy out of the four available, which we discuss in further detail. The fourth category, "False or misleading affiliation" is related to the use of fake and parody accounts, and thus not related to misleading images. Similarly, we include the "Synthetic and Manipulated Media Policy" as this policy covers the usage of media on Twitter, which is the major modality of content in our study. The details of the individual rules and categories are listed in Appendix A.

To begin the process, we first randomly sample 100 *Image Stories* out of the 258 extracted to guide us in developing the codebook. First, we take each of the 16 different rules of Twitter's Civic Integrity Policy [84], divided into four major categories as the draft of our initial codebook. Two annotators independently code the sampled images, considering additional contextual information learned from fact-checking articles associated with them. We discussed these codes, repeating the process three times until the final codebook reached a point where further iterations would not improve it. Note that we use the 16 different rules to help us identify the closest violation of Twitter's platform policy by the images while annotating the *Image stories* with the granularity corresponding to the four categories associated with the rules. We reach a nearly perfect agreement (Cohen's Kappa score of $\kappa = 0.9$ [55]).

Positioning the extracted *Image Stories* alongside the platform policies of Twitter helps us re-evaluate the content moderation decision based on Twitter implementations and policies. We present an example image violating each of the four categories discussed in Figure 7. In the rest of this section, we describe the four categories in our codebook in detail, along with their definitions and our evaluation for matching images with the corresponding category.

1. ***Misleading information about how to participate in civic processes***: This category refers to images that mislead people about "election participation procedures and requirements, cause confusion about officials, or discuss threats to voting locations." Note that we did not find any images related to threats at voting locations in the extracted *image stories*. For election participation procedures and requirements, misleading images aim to spread false claims about the irregularity of the criteria by which votes are being cast. For example, dead people are casting votes; votes are cast multiple times; votes are being cast after the closing of polls. Misleading images may also claim that mail-in ballots are insecure, illegal, and a

source of voter fraud. Figure 7a aims to intimidate voters from voting by mail, casting doubts over the security of mail-in ballots despite mail-in ballots proven to be safe to use [1].

2. ***Misleading information intended to intimidate people from civic processes***: This category contains images that mislead people about "how votes are being counted (or not counted), problems with ballot equipment, disruption at voting locations, and the closing of polls". Misleading images in this category aim to make claims about legitimately cast ballots getting invalidated, malfunction of voting machines, switching of votes between candidates, and deleting of votes on the machines. We find images spreading misleading claims about disruptions at voting locations like poll workers filling ballots, poll workers forcing voters to use Sharpies, and ballot observers not being allowed to observe the counting process. The example in Figure 7b is spreading false claims about ballots using Sharpies in Arizona being invalidated and poll workers forcing people to use Sharpies [75].

3. ***Misleading information about the outcomes of civic processes***: This category refers to the images that mislead people about "election rigging, ballot tampering, vote tallying, declaration of premature victory, casting doubt on the outcome of civic processes, calling for interference with the implementation of election results, and undermining public confidence in the methods and results of the election." Misleading images in this category aim to make claims about the illegal processing and handling of ballots, which include alteration, manipulation, destruction, forging, counterfeiting, stealing, and pre-filling ballots. We also find false claims about voter registration and turnout, and visual and statistical anomalies in patterns of vote counting. By casting doubt on the outcome of civic processes, misleading information aims to question the integrity of the election process, discredit different stages of election processes, and call for interference with the implementation of election results using slogans such as "Stop the Steal," "Fraud." The misleading image in Figure 7c casts doubt over bipartisan vote tallying process, suggesting that Trump ballots were discarded in garbage bags [1].

4. ***Synthetic and Manipulated Media***: This category refers to images that are "significantly and deceptively altered, manipulated, or fabricated, and images shared with deliberate intent to deceive people". For this category, included images are significantly and deceptively altered, manipulated, or fabricated. On the other hand, images shared with malicious intent include out-of-context tweets sharing the media. For example, adding quotations shared in the past that are taken out of context to spread election misinformation or images from the past that are re-used in the present context, like discarded ballots from the 2018 election being repurposed to say ballots are being tampered with in the 2020 election. Figure 7d is a doctored image in connection to the "Bush-Gore Florida recount" during the presidential race in 2000 [9], shared by a Trump campaign spokesman.The Washington Times later confirmed that they never published such a story [69].

We assign each *Image Story* to *one* category. There are some observations from our annotation process: We notice that the use of memes is common in spreading misleading information. They are designed for people to add text to them to make a variation. During the annotation process, we first assess the overlay text on the memes to see if it falls into any of the categories other than the fourth: Synthetic and Manipulated Media. If it does not, we classify it as an image altered to be shared with malicious intent.

We also find that a large portion of images are used out of context and could potentially be categorized into the fourth category: Synthetic and Manipulated Media. However, out-of-context information is widely used to push narratives related to the other three themes. Thus, we take precedence of annotating the images into the first three categories if it suits any. Otherwise, it is annotated as the fourth category. We annotate these images according to their primary content to expand and deepen the qualitative analysis of these moderated images. For example, a screenshot of a video clip in which ballot workers are transcribing ballots is used out of context and spread with the misleading content of "ballot stuffing." In this example, the medium of spreading the misleading image is taking it out of context. However, the primary content being spread is related to vote counting and ballot tampering, which falls under the Twitter Civic Integrity Policy. Thus, we annotate it as Category Three: Misleading information about the outcomes of civic processes.

Moderation of misleading images by violation category. After associating the misleading images with the corresponding platform policy violations, we want to understand whether Twitter is more likely to moderate images violating some of them. Table 3 reports the breakdown of the rate of moderation by Twitter of images belonging to the different categories of violations, among the tweets identified by PIXELMOD. A high-level overview of the breakdown by category suggests that Twitter moderated specific types of misleading *image stories* more aggressively, while the moderation rate on other categories appears to be laxer. Misleading *image stories* of the categories "Intimidation from Civic Processes" and "Participation in Civic Processes" are moderated the most, with 5.96%, and 4.16% of the images getting moderated. Surprisingly, the most popular category of content violation among *image Stories*, "Misleading information about outcomes of civic processes" has the lowest moderation rate of 1.77%. Finally, images violating the category "Synthetic and Manipulated Media" have a moderation rate of 2.76%. While we can make overall observations that different types of platform policy violations might have been disproportionately moderated from Twitter's end, we find concerning results of overall moderation rate across the categories being very low.

Category	# *Image stories*	Moderation %
Participation in Civic Processes	57	4.16%
Intimidation from Civic Processes	78	5.96%
Outcomes of Civic Processes	81	1.77%
Synthetic and Manipulated Media	42	2.76%

Table 3: Moderation rate of images breakdown by category.

6 Related Work

Soft Moderation Approaches by Social Media. Social media platforms namely Facebook and Twitter have increasingly shifted their approach towards warning labels as a tool for content moderation, and additionally, provide surrounding context to the users when interacting with potentially false information. These platforms have applied warning labels on false information shared ranging from Covid-19 pandemic [21, 23, 67, 68], 2020 US Presidential Elections [5, 14, 19, 34] to climate denial misinformation [17, 20]. Twitter has reported that approximately 74% of the tweet viewership happened after the warning labels were applied to the tweets and, warning labels were effective in decreasing users quoting the misleading tweets by an estimated 29%. Savvas Zannettou [91] performed empirical analysis on a sample of 2,244 tweets with warning labels related to the 2020 US Presidential Elections. Paudel et al. proposed LAMBRETTA, a system that aims to improve soft moderation of textual content on Twitter by leveraging Learning to Rank [61]. To the best of our knowledge, PIXELMOD is the first end-to-end approach to perform soft-moderation of images on social media that has been proposed by the research community.

Applications of Perceptual Hashing in Computer Security. Perceptual hashing techniques have been widely used in computer security, leveraging them to detect phishing websites [47], scam websites [46, 56], fraudulent services [60], and to identify impersonators on social media [30]. Perceptual hashes have also been used for content authentication [6, 86], and tamper detection on images [80, 94]. Alkhowaiter et al. [8] evaluated six types of perceptual hashes on their ability to detect image manipulation and transformation over two major social media platforms: Facebook and Twitter. Historically, Microsoft developed PhotoDNA [40], a type of syntactic embedding to identify and report the distribution of child exploitation material. Facebook currently uses PDQHash in the "ThreatExchange" platform[1] to share "signals" of harmful media on their platform.

Image-based misinformation in social networks. Several studies are focusing on developing automated detection methods to identify images containing misinformation, either manipulated images [2, 12, 95], or images that are taken out of context or misinterpreted on social media [3, 10, 25, 42, 96]. Most of the works on multi-modal misinformation tackle classifying or detecting a single image as misinformation, while very few works have focused on studying the spread, and diffusion of misleading images within and across social media. The closest work to PIXELMOD is a system called DejaVu [53], which is designed to assist journalists in collaboratively addressing the spread of visual misinformation by using ORB descriptors [70] (another type of syntactic embedding) to encode the images, and FAISS [43] to index the images. Additionally, other works study the spread of COVID-19 media through WhatsApp in Pakistan [41], and other types of visual misinformation in WhatsApp [44, 66]. On the other hand, works by [59, 93] study the usage of images in state-sponsored influence campaigns. Wang et al. [89] analyzed the spread of *Fauxography* images on social media, which are images that are presented in an incorrect or untruthful fashion, by using ground truth fact-checked images from fact-checked organization Snopes [2]. Similarly, Zannettou et al. used visual similarity (i.e., perceptual hashing) and images annotated by the website KnowYourMeme to study the evolution and diffusion of image memes posted on social media [92].

7 Discussion and Conclusion

In this paper, we presented PIXELMOD, a scalable system able to identify images that are candidates for soft moderation on Twitter. PIXELMOD overcomes the inability of perceptual hashing to discern the context of an image by incorporating OCR into the matching process. Our results show that PIXELMOD outperforms three types of image-matching systems based on perceptual hashing, image descriptors, and deep neural networks. With the highest F1-score among competitors, PIXELMOD places itself as the state of the art in automated image-based soft moderation.

We believe that PIXELMOD (which we make publicly available)[3] will be an inspiring foundation for researchers and online platforms aiming to improve content moderation on social media. In the rest of this section, we first discuss the ethical considerations of our work and design implications that online platforms should keep in mind when deploying PIXELMOD. We then discuss the limitations of our approach and some directions for future work.

Ethical Considerations. Our work only uses publicly available Twitter data that was collected from the official API while it was still open to academic researchers, and we do not interact with users. As such, this work is not considered human subject research by our institution. We also preserve the privacy of Twitter users as we do not analyze any personally identifiable information (e.g., location data, account names) and blur the regions of example tweets used in the paper containing identifiable (meta)data. Additionally, we take steps to blur the example images used in the paper unless i) they

[1] https://developers.facebook.com/docs/threat-exchange/

[2] https://www.snopes.com/fact-check/

[3] https://github.com/idramalab/pixelmod

are public figures, ii) they are an illustration, and iii) they are stock images. As with any content moderation system, PixelMod could be used for malicious purposes like censorship and surveillance. We advocate that the system should be used with ethical principles in mind, following the *respect for public interest* and *beneficence* principles from the Menlo report [45].

Design implications. Our experiments found that Twitter soft moderation misses most images that should be labeled. This means that leveraging PixelMod would help Twitter's moderators cover more false information on their platform. However, when deploying PixelMod, Twitter or other online platforms should take several aspects into consideration. First, while PixelMod's detection performance exceeds that of existing approaches, a false detection rate of 0.99% may still be considered too high by online platforms to consider its adoption as a fully automated soft-moderation system. We envision PixelMod to be used by platform moderators as a tool to identify a set of candidates for soft moderation with limited false positives, which then receive manual vetting. The ultimate decision on whether to apply the moderation labels, however, should remain with the human operator. This is not dissimilar from what online platforms are doing already, but our approach would allow them to obtain a more comprehensive view of misleading content on their network. Additionally, PixelMod could help address the main pain points of relying on human moderation, which is the latency in decision-making and having limited moderator resources [11, 78].

Second, PixelMod is an inherently reactive system: it requires a set of images already identified by the platform as misleading. This process could be streamlined by using example images that have been fact-checked by dedicated organizations [89]. When curating the set of seed images in Section 5.2.1, we found that querying images directly from a list of moderated tweets can be tricky, since all of the moderated images might not be of misleading nature. In such cases, moderators using PixelMod should take an additional step to ensure the query images are misleading in nature, rather than ambiguous images not related to the events being studied, to get the most relevant results as candidates for moderation.

Misleading images can spread across multiple social media platforms, propagating with different contexts and forms [39, 90]. Platforms can use PixelMod as a tool alongside an industry-shared database of known misleading image hashes for tracking the spread of misleading images on their service and across other online communities. Platforms already use shared databases for tracking Child Sexual Abuse Material (through the National Center for Misleading and Exploited Children) and terrorism content (through the Global Internet Forum to Counter Terrorism) [22, 33, 58]. A tool like PixelMod backed by a centralized database could easily be added to major social media platform's existing efforts to combat fraud and misinformation [77].

Runtime implications. As discussed in Section 5.1, the increased runtime overhead of PixelMod is due to the OCR component, which only gets triggered for 0.973% of the images in $\mathbf{GT}_{viz}$. This rate is 1.450% among the 19.7M images used in our 'in-the-wild' evaluation in Section 5.2.1. It is to note that factors like image resolution, background, and font complexity could also impact the runtime of the underlying OCR engine. The overhead reported is an upper bound on the runtime of PixelMod and the average runtime of our system is in the same order as other baselines. This occasional slowdown is a tradeoff we make for increased recall for PixelMod, which allows us to achieve a much higher coverage of soft-moderation candidates than the ones achieved by Twitter. However, PixelMod can be adapted according to the content moderation budget of the platform. The initial set of results retrieved by PixelMod (visual matches to query images) can be sorted or filtered through metadata such as the popularity of the account posting the images, or other metadata aligning with specific content moderation strategies of a platform, before passing through the contextual similarity component.

Applying PixelMod to other platforms and topics. We could not test PixelMod on other topics and platforms due to the lack of reliable datasets across platforms and topics. First, we are not able to test our system on other social media platforms due to the lack of access to a uniform sample of posts (like the 1% sample that forms our $\mathbf{D}_2$ dataset) While datasets containing misleading images exist for other platforms like WhatsApp and Telegram, these are not suited for our evaluation since they only contain a single instance of labeled misleading images, making the visual similarity research process that is at the center of PixelMod moot. Even though Twitter applied warning labels on COVID-19 misinformation, these were unreliable and inconsistent [48, 52], which we independently confirmed in our preliminary analysis. Despite the limitations on evaluation settings, we expect PixelMod to generalize well across multiple platforms and topics. The only requirement for a platform to apply PixelMod to a new campaign is a set of seed images that are known to be misleading, and the system should generalize well to other platforms and topics, as the underlying image embeddings are syntactic in nature, and do not incorporate any domain-specific metadata (e.g. number of retweets, number of likes available on tweets).

Limitations. Despite the promising performance of PixelMod in identifying visually similar images at scale, the embedding used by PixelMod (PDQHash) is vulnerable to adversarial manipulations [36, 38]. An adversary could modify images to have a PDQhash that is very far from the one of the corresponding seed image, generating a false detection by PixelMod. This is a serious risk, and future research should investigate defenses against these attacks. Some potential avenues of defense are utilizing an ensemble of hashes: combining results from both pHash and PDQHash [36], and adversarial training of embeddings [87]. Using an ensemble of hashes would force an attacker to jointly optimize the adversarial attacks against an ensemble of hashing methods as opposed to a single method, thus increasing the operation cost on the end of adversaries. Future works can also look at incorporating the contextual information (OCR text) contained in the images as part of generating the syntactic embeddings themselves, making it difficult for adversaries to modify the images without deviating from the contextual messaging of the images. At the same time, the threat model here assumes centralized coordination by an adversary. While this is within reach when dealing with state-sponsored disinformation actors [59, 71], it is not applicable to content spread autonomously and in good faith by regular users, for example in the wake of the uncertainty surrounding the COVID-19 pandemic. In the case of crisis scenarios, misleading content with significant risk factors might proliferate as many different variants of the original content as a consequence of the broad and diverse vector of sharing by people. This might render the initial set of seed hashes and similarity threshold being used ineffective in tracing the spread of the images. One such example of this was the spreading of videos of the Christchurch mosque shooting, where Facebook reported that their systems were defeated by variants of the original videos as bad actors started sharing it [58]. Platforms could use methods from online learning to update the seed set of hashes to flag, and dynamically adjust the distance threshold in specific crisis cases to better handle such events.

Apart from applying soft moderation warning labels, Twitter outlines "reduced visibility" as a possible consequence for tweets violating their Civic Integrity Policy [84]. This means it is possible Twitter could have identified all of the images subsequently identified by PIXELMOD, and chosen to reduce visibility of tweets including them instead of applying warning labels. However, we cannot analyze this phenomenon due to the lack of any public metrics regarding the visibility of tweets. Due to the nature of the evaluation datasets, images containing multi-lingual text could not be evaluated. Therefore, future applications of the system in multi-lingual settings might require further configuration of the underlying OCR engine (e.g. specifying *"languageHints"* parameters in Cloud Vision OCR engine with the intended language of evaluation).

Future work. In the future, we plan to apply PIXELMOD to other online platforms. The approach is platform-independent, and this could give interesting insights on how misinformation spreads across different online communities, and which communities are particularly influential in generating viral image misinformation. For example, we believe quite strongly that because of its low computational overhead and high performance, it makes a good fit for deployment by decentralized social networks (e.g., Mastodon), an environment where recent work has shown the potential to actually help *improve* PIXELMOD's performance via federated model sharing [13]. Since our approach does not require any platform-specific information, it is particularly interesting as a basis for further research, especially at a time where academic research is being seriously threatened by the discontinuation of the Twitter Academic API as we know it. Regardless of the specifics of future work, our presentation of PIXELMOD provides a roadmap for measuring, tuning, and benchmarking soft moderation systems; critical for any moderation tool's success. We strongly believe that the computer security community has a lot to say in this space, and hope that more researchers will get into this space.

Acknowledgments. We would like to thank the anonymous reviewers and the anonymous shepherd for their feedback, and Primah Muwanga for her help setting up the PIXELMOD artifact after the paper was accepted. This work was supported by the NSF under grants CNS-1942610, CNS-2114407, CNS-2114411, CNS-2247867, CNS-2247868, and IIS-2046590, and by a grant from the Media Ecosystems Analysis Group (MEAG).

References

[1] Trump's repeated false attacks on mail-in ballots. https://www.factcheck.org/2020/09/trumps-repeated-false-attacks-on-mail-in-ballots/, Sep 2020.

[2] S. Abdali, R. Gurav, S. Menon, D. Fonseca, N. Entezari, N. Shah, and E. E. Papalexakis. Identifying misinformation from website screenshots. *arXiv preprint arXiv:2102.07849*, 2021.

[3] S. Abdelnabi, R. Hasan, and M. Fritz. Open-domain, content-based, multi-modal fact-checking of out-of-context images via online resources. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14940–14949, 2022.

[4] A. Abilov, Y. Hua, H. Matatov, O. Amir, and M. Naaman. Voterfraud2020: a multi-modal dataset of election fraud claims on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, 2021.

[5] D. Abril. Facebook reveals that massive amounts of misinformation flooded its service during the election, Nov 2020.

[6] F. Ahmed, M. Y. Siyal, and V. U. Abbas. A secure and robust hash-based scheme for image authentication. *Signal Processing*, 90(5), 2010.

[7] K. T. Ahmed, S. Ummesafi, and A. Iqbal. Content based image retrieval using image features information fusion. *Information Fusion*, 51, 2019.

[8] M. Alkhowaiter, K. Almubarak, and C. Zou. Evaluating perceptual hashing algorithms in detecting image manipulation over social media platforms. In *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2022.

[9] E. Alvarado, D. A. Graham, C. Murphy, and A. Amy Weiss-Meyer. The bush-gore recount is an omen for 2020. https:

//www.theatlantic.com/politics/archive/2020/08/bush-gore-florida-recount-oral-history/614404/, Aug 2020.

[10] S. Aneja, C. Bregler, and M. Nießner. Cosmos: Catching out-of-context misinformation with self-supervised learning, 2021.

[11] A. Arsht and D. Etcovitch. The human cost of online content moderation. *Harvard Journal of Law and Technology*, 2018.

[12] B. Bayar and M. C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *ACM IH*, 2016.

[13] H. Bin Zia, A. Raman, I. Castro, I. Hassan Anaobi, E. De Cristofaro, N. Sastry, and G. Tyson. Toxicity in the decentralized web and the potential for model sharing. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(2), jun 2022.

[14] S. Bond. Twitter expands warning labels to slow spread of election misinformation, Oct 2020.

[15] S. Bradshaw and S. Grossman. Were Facebook and Twitter consistent in labeling misleading posts during the 2020 election? https://www.lawfareblog.com/were-facebook-and-twitter-consistent-labeling-misleading-posts-during-2020-election, 2022.

[16] C. Caldera. Fact check: Map showing trump landslide based on false report of seized election servers in germany. https://www.usatoday.com/story/news/factcheck/2020/11/18/fact-check-fake-map-shows-trump-with-410-electoral-votes/3767048001/, Nov 2020.

[17] J. Calma. Facebook will add a new label to some climate change posts in the uk, Feb 2021.

[18] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*. Springer, 2020.

[19] K. Conger. Twitter says it labeled 0.2disputed., Nov 2020.

[20] S. Connellan. Facebook to add labels to climate change posts, Oct 2021.

[21] E. Culliford. Twitter launches labels, warnings on misleading covid-19 information. https://www.reuters.com/article/us-health-coronavirus-twitter/twitter-launches-labels-warnings-on-misleading-covid-19-information-idUSKBN22N2E4, May 2020.

[22] E. Culliford. Exclusive zoom has joined tech industry counterterrorism group, Dec 2021.

[23] E. Culliford. Facebook to label all posts about covid-19 vaccines. https://www.reuters.com/article/us-health-coronavirus-facebook/facebook-to-label-all-posts-about-covid-19-vaccines-idUSKBN2B70NJ, Mar 2021.

[24] A. Davis and G. Rosen. Open-sourcing photo-and video-matching technology to make the internet safer. *Facebook Newsroom*, 2019.

[25] P. Dewan, A. Suri, V. Bharadhwaj, A. Mithal, and P. Kumaraguru. Towards Understanding Crisis Events On Online Social Networks Through Pictures. In *ASONAM*, 2017.

[26] A. Dutta, A. Gupta, and A. Zissermann. Vgg image annotator (via). *URL: http://www. robots. ox. ac. uk/vgg/software/via*, 2, 2016.

[27] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, 1996.

[28] K. Garimella and D. Eckles. Image based misinformation on whatsapp. In *International AAAI Conference on Web and Social Media (ICWSM)*, 2017.

[29] K. Garimella and D. Eckles. Images and misinformation in political groups: Evidence from whatsapp in india. *arXiv:2005.09784*, 2020.

[30] O. Goga, G. Venkatadri, and K. P. Gummadi. The doppelgänger bot attack: Exploring identity impersonation in online social networks. In *Proceedings of the 2015 internet measurement conference*, 2015.

[31] F. González-Pizarro and S. Zannettou. Understanding and detecting hateful content using contrastive learning. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, 2023.

[32] Google. Cloud vision api. https://cloud.google.com/vision/.

[33] Google. Google's efforts to combat online child sexual abuse material. https://transparencyreport.google.com/child-sexual-abuse-material/reporting.

[34] M. Graham and S. Rodriguez. Twitter and facebook race to label a slew of posts making false election claims before all votes counted, Nov 2020.

[35] V. N. Gudivada and V. V. Raghavan. Content based image retrieval systems. *Computer*, 28(9), 1995.

[36] Q. Hao, L. Luo, S. T. Jan, and G. Wang. It's not what it looks like: Manipulating perceptual hashing based applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[37] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[38] S. Hu, Z. Zhou, Y. Zhang, L. Y. Zhang, Y. Zheng, Y. He, and H. Jin. Badhash: Invisible backdoor attacks against deep hashing with clean label. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

[39] K. Hunt, B. Wang, and J. Zhuang. Misinformation debunking and cross-platform information sharing through twitter during hurricanes harvey and irma: a case study on shelters and id checks. *Natural Hazards*, 103(1), 2020.

[40] T. Ith. Microsoft's photodna: Protecting children and businesses in the cloud. *URL: https://news. microsoft. com/features/microsofts-photodna-protecting-children-andbusinesses-in-the-cloud/366 REFERENCES*, 2015.

[41] R. T. Javed, M. E. Shuja, M. Usama, J. Qadir, W. Iqbal, G. Tyson, I. Castro, and K. Garimella. A first look at covid-19 messages on whatsapp in pakistan. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2020.

[42] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian. Novel visual and statistical image features for microblogs news verification. *IEEE transactions on multimedia*, 19(3):598–608, 2016.

[43] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 2019.

[44] A. Kazemi, K. Garimella, G. K. Shahi, D. Gaffney, and S. A. Hale. Tiplines to combat misinformation on encrypted platforms: a case study of the 2019 indian election on whatsapp. *arXiv preprint arXiv:2106.04726*, 2021.

[45] E. Kenneally and D. Dittrich. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102*, 2012.

[46] A. Kharraz, W. Robertson, and E. Kirda. Surveylance: Automatically detecting online survey scams. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018.

[47] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[48] J. Lange. Twitter is now flagging the use of 'oxygen' and 'frequency' in the same tweet, prompting new meme. https://theweek.com/speedreads/922275/twitter-now-flagging-use-oxygen-frequency-same-tweet-prompting-new-meme, 2020.

[49] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014.

[51] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2. Ieee, 1999.

[52] K. Lyons. Twitter promises to fine-tune its 5G coronavirus labeling after unrelated tweets were flagged. https://www.theverge.com/2020/6/27/21305503/twitter-labels-5g-conspiracy-coronavirus, 2020.

[53] H. Matatov, A. Bechhofer, L. Aroyo, O. Amir, and M. Naaman. Dejavu: a system for journalists to collaboratively address visual misinformation. In *Computation+ Journalism Symposium. Miami*, 2018.

[54] H. Matatov, M. Naaman, and O. Amir. Stop the [image] steal: The role and dynamics of visual content in the 2020 us election misinformation campaign. *arXiv preprint arXiv:2209.02007*, 2022.

[55] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 2012.

[56] N. Miramirkhani, O. Starov, and N. Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. *arXiv preprint arXiv:1607.06891*, 2016.

[57] V. Monga and B. L. Evans. Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs. *IEEE Transactions on Image Processing*, 2006.

[58] F. Newsroom. Partnering to help curb spread of online terrorist content, Dec 2016.

[59] L. H. X. Ng, J. Moffitt, and K. M. Carley. Coordinated through aweb of images: Analysis of image-based influence operations from china, iran, russia, and venezuela. *arXiv preprint arXiv:2206.03576*, 2022.

[60] N. Nikiforakis, F. Maggi, G. Stringhini, M. Z. Rafique, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, and S. Zanero. Stranger danger: exploring the ecosystem of ad-based url shortening services. In *Proceedings of the 23rd international conference on World wide web*, 2014.

[61] P. Paudel, J. Blackburn, E. De Cristofaro, S. Zannettou, and G. Stringhini. Lambretta: Learning to rank for twitter soft moderation. In *IEEE Symposium on Security and Privacy*, 2023.

[62] T. D. Platform. Tweet annotations. https://developer.twitter.com/en/docs/twitter-api/annotations/overview.

[63] Y. Qu, X. He, S. Pierson, M. Backes, Y. Zhang, and S. Zannettou. On the evolution of (hateful) memes by means of multimodal contrastive learning. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023.

[64] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 2021.

[65] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[66] J. Reis, P. d. F. Melo, K. Garimella, and F. Benevenuto. Can whatsapp benefit from debunked fact-checked stories to reduce misinformation? *arXiv preprint arXiv:2006.02471*, 2020.

[67] V. Romo. Twitter to remove or place warning labels on covid vaccine conspiracy tweets. https://www.npr.org/sections/coronavirus-live-updates/2020/12/16/947355414/twitter-to-remove-or-place-warning-labels-on-covid-vaccine-conspiracy-tweets, Dec 2020.

[68] G. Rosen. An update on our work to keep people informed and limit misinformation about covid-19. https://about.fb.com/news/2020/04/covid-19-misinfo-update/, May 2021.

[69] C. O. Rourke. No, the washington times didn't run the 'president gore' headline that trump spokesman shared. https://www.politifact.com/factchecks/2020/nov/09/viral-image/no-washington-times-didnt-run-president-gore-cover/, Nov 2020.

[70] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*. Ieee, 2011.

[71] M. H. Saeed, S. Ali, J. Blackburn, E. De Cristofaro, S. Zannettou, and G. Stringhini. Trollmagnifier: Detecting state-sponsored troll accounts on reddit. In *IEEE Symposium on Security and Privacy (SP)*, 2022.

[72] D. Sayce. The number of tweets per day in 2022. https://www.dsayce.com/social-media/tweets-day/.

[73] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[74] J. E. Sklan, A. J. Plassard, D. Fabbri, and B. A. Landman. Toward content-based image retrieval with deep convolutional neural networks. In *Medical Imaging 2015: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 9417. SPIE, 2015.

[75] R. Staff. Fact check: Tabulation machines in arizona can read ballots marked with sharpie pens. https://www.reuters.com/article/uk-factcheck-sharpie-arizona/fact-check-tabulation-machines-in-arizona-can-read-ballots-marked-with-sharpie-pens-idUSKBN27L2R5, Nov 2020.

[76] K. Starbird, A. Arif, and T. Wilson. Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW), 2019.

[77] N. Statt. Major tech platforms say they're 'jointly combating fraud and misinformation' about covid-19, Mar 2020.

[78] M. Steiger, T. J. Bharucha, S. Venkatagiri, M. J. Riedl, and M. Lease. The psychological well-being of content moderators: the emotional labor of commercial moderation and avenues for improving support. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, 2021.

[79] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[80] Z. Tang, S. Wang, X. Zhang, W. Wei, and S. Su. Robust image hashing for tamper detection using non-negative matrix factorization. *Journal of ubiquitous convergence technology*, 2(1), 2008.

[81] TinEye. Tineye: Reverse image search. https://tineye.com/.

[82] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5), 2009.

[83] Twitter. Synthetic and manipulated media policy. https://help.twitter.com/en/rules-and-policies/manipulated-media, 2020.

[84] Twitter. Twitter's civic integrity policy | twitter help, 2020.

[85] J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu, K. Yu, Y. Yuan, Y. Zou, J. Long, Y. Cai, Z. Li, Z. Zhang, Y. Mo, J. Gu, R. Jiang, Y. Wei, and C. Xie. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, 2021.

[86] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue. A visual model-based perceptual image hash for content authentication. *IEEE Transactions on Information Forensics and Security*, 10(7), 2015.

[87] X. Wang, Z. Zhang, G. Lu, and Y. Xu. Targeted attack and defense for deep hashing. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[88] Y. Wang, C. Ling, and G. Stringhini. Understanding the use of images to spread covid-19 misinformation on twitter. *Proceedings of the ACM in Human Computer Interaction (CSCW)*, 2023.

[89] Y. Wang, F. Tamahsbi, J. Blackburn, B. Bradlyn, E. De Cristofaro, D. Magerman, S. Zannettou, and G. Stringhini. Understanding the use of fauxtography on social media. In *ICWSM*, 2021.

[90] T. Wilson and K. Starbird. Cross-platform disinformation campaigns: lessons learned and next steps. *Harvard Kennedy School Misinformation Review*, 1(1), 2020.

[91] S. Zannettou. I won the election: An empirical analysis of soft moderation interventions on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, 2021.

[92] S. Zannettou, T. Caulfield, J. Blackburn, E. De Cristofaro, M. Sirivianos, G. Stringhini, and G. Suarez-Tangil. On the Origins of Memes by Means of Fringe Web Communities. In *Proceedings of the Internet Measurement Conference 2018*, 2018.

[93] S. Zannettou, T. Caulfield, B. Bradlyn, E. De Cristofaro, G. Stringhini, and J. Blackburn. Characterizing the use of images in state-sponsored information warfare operations by russian trolls on twitter. In *Proceedings of the international AAAI conference on web and social media*, 2020.

[94] Y. Zhao and W. Wei. Perceptual image hash for tampering detection using zernike moments. In *2010 IEEE International Conference on Progress in Informatics and Computing*, volume 2. IEEE, 2010.

[95] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018.

[96] D. Zlatkova, P. Nakov, and I. Koychev. Fact-checking meets fauxtography: Verifying claims about images. In *EMNLP-IJCNLP*, 2019.

A Twitter's content policies and violations.

We list the four different categories of Twitter's platform policy violations and the corresponding rules for each category below.

noitemsep

- Misleading information about how to participate.
  - Images misleading people about participation procedures and requirements.
  - Images sowing confusion about officials and institutions.
  - Images discussing threats on voting locations.
- Misleading information intended to intimidate people from civic processes.
  - Images about votes not being counted.
  - Images about Equipment Problems.
  - Images about disruptions at voting locations.
  - Images about closing of polls.
- Misleading information about outcomes of civic processes.
  - Images undermining public confidence in methods and results of election.
  - Images with misleading claims about election rigging.
  - Images with misleading claims about ballot tampering.
  - Images with misleading claims about vote tallying.
  - Images with declaration of premature victory.
  - Images casting doubt on outcome of civic processes.
  - Images calling for interference with the implementation of election results.
- Synthetic and Manipulated Media.
  - Images that are significantly and deceptively altered, manipulated or fabricated.
  - Images shared with malicious intent , including out of context tweets sharing the media.