Image Authentication Using Self-Supervised Learning To Detect Manipulation Over Social Network Platforms

Mohammed Alkhowaiter,[1,2] Khalid Almubarak,[2] Mnassar Alyami,[1] Abdulmajeed Alghamdi,[1] and Cliff Zou[1]

[1]*College of Engineering and Computer Science, University of Central Florida, USA*
{mok11@knights. | mnassar.alyami@knights. | a.alghamdi@knights. | changchun.zou@}ucf.edu
[2]*College of Computer Engineering and Science, Prince Sattam University, Saudi Arabia*
{m.khowaiter@ | k.almubarak@}psau.edu.sa

***Abstract*—Social media nowadays has a direct impact on people's daily lives as many edge devices are available at our disposal and controlled by our fingertips. With such advancement in communication technology comes a rapid increase of disinformation in many kinds and shapes; faked images are one of the primary examples of misinformation media that can affect many users. Such activity can severely impact public behavior, attitude, and belief or sway the viewers' perception in any malicious or benign direction. Mitigating such disinformation over the Internet is becoming an issue with increasing interest from many aspects of our society, and effective authentication for detecting manipulated images has become extremely important. Perceptual hashing (pHash) is one of the effective techniques for detecting image manipulations. This paper develops a new and a robust pHash authentication approach to detect fake imagery on social media networks, choosing Facebook and Twitter as case studies. Our proposed pHash utilizes a self-supervised learning framework and contrastive loss. In addition, we develop a fake image sample generator in the pre-processing stage to cover the three most known image attacks (copy-move, splicing, and removal). The proposed authentication technique outperforms state-of-the-art pHash methods based on the SMPI dataset and other similar datasets that target one or more image attacks types.**

***Index Terms*—Perceptual hashing, Computer security, Social media, Fake news, Digital forensics**

I. Introduction

Nowadays, the spread of disinformation over the Internet is an increasingly important and impactful social and technical issue to be studied and analyzed. Social media platforms that connect all nations over the world into one place with the support of sharing all media content types (text, video, image, and sound) aggravate the false information spread. Study [1] shows that fake images on social media increase user engagement, irrespective of the depth of the manipulation. Fake news and information could cause harm to others if not detected early. Figure 1 shows a real-world example of fake image spreading over the media. The altered photograph, investigated by Snopes.com [2], shows a billboard to hire 'crypto bros' advertised by McDonald's. This faked image spread along with the crush of some cryptocurrencies leaving a cruel emotion to those investors who lost their money. Another manipulated image spread over the Internet shows George W. Bush at a book reading at school in Houston in 2002 holding the book upside down with a false caption [3]. Recently, many platforms activated fake detection features on their platforms to reduce or eliminate false information, e.g., COVID-19 misleading information [4]. But there are still many technical challenges to research and conquer in order to win the war against disinformation spreading.

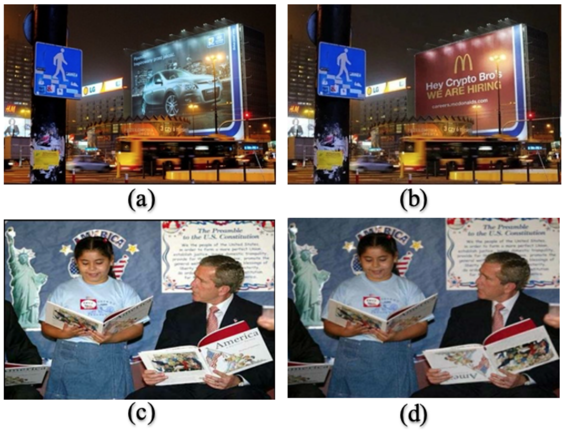


Fig. 1. Originals images (a and c) and their altered copies (b and d) that spread over the network.

Perceptual Hashing (pHash) is an alternative technique to replace cryptographic hashing for media authentication in many platforms, such as social media, where the multimedia content could be legitimately transformed by the platforms [5]. In such case, a single bit modification would invalidate the authentication function by traditional crypto-hashing techniques. Intensive work on pHash has been introduced in different applications such as image near-duplicates [6], search engines [7], image retrieval [8], and image authentication [9]–[15]. The research on these applications studies the images' similarities to detect content-preserving manipulations that do not change an image's content.

The development of previous works relied on different benchmarks, such as CASIA [16], USC-SIPI [17], and PS-Battles [18]. The majority of the datasets used in these previous papers were manually crafted or applied significant alterations in the images, making the developed models unverifiable by real-world applications nor effective for small content-alteration. To demonstrate this issue, authors of [19] developed a new benchmark that targets real-world applications. It shows a weakness on image authentication algorithms on real-world application using Facebook and Twitter platforms images.

In addition, we conducted a literature review on pHash and found that few works have been introduced for image authentication using machine learning, such as [11], which shows that Convolutional Neural Network (CNN) is effective in developing a pHash system with high accuracy. However, at the evaluation step in [11], they used the same JPEG compression with quality factors of {1, 5, 10, 30, 50, 70, 90, 95} at its training stage as content-preserving image operation. In real-world applications, the quality factor could be any number between 1:100, which might create an evaluation bias in their proposal.

The success of related work on image classification [20], [21] inspired us to conduct investigation of different CNN approaches for image authentication. Regardless of different CNN architecture design, such as layer length, channels, and kernel size, the output of each CNN model provides a best image feature representation for classification. We found these vector representations can be projected and exploited for image hashing with a smaller hash length to keep more space in the memory. In short, the projection vectors of the last layer were used and converted into buckets using random projection of Locality Sensitive Hashing (LSH) [22].

Inspired by [10], [11] that use machine learning for image authentication, we integrated a well known CNN network [23] to enhance image hashing generation. We propose a model to detect manipulations on User-generated content, and evaluate it using Facebook and Twitter as case studies.

Our contributions to this study are as follows:

- An alteration technique is designed for better detecting copy-move, splicing, and removal operations for the pre-processing phase.
- Self-supervised framework using a ResNet-18 [21] model is constructed and trained to obtain the image features.
- Locality Sensitive Hashing (LSH) is integrated with a deep CNN at the test phase to construct the final hash.
- Our method is compared with state-of-the-art methods using SMPI dataset [19], IMD2020 [24] and COVERAGE [25] and showed best performance among them all for detecting manipulated images.

The remainder of this paper is organized as follows: We review related work in Section II. Our proposed approach is described in Section III-B. We evaluate our technique and discuss our results in Section IV and Section V, respectively. Finally, we draw our conclusion and discuss future works in Section VI.

II. Related Work

Many image authentication works are shallow approaches that use traditional engineered algorithms. However, in recent years awareness of machine learning approaches increased with the success of deep learning models such as AlexNet in ImageNet classification challenge and promising results in other computer vision problems. Therefore, it is reasonable to say that most image authentication algorithms are built on top of shallow or machine learning paradigms. The following is an overview of most recognized works on images pHash under these two approaches.

Shallow approach– The followers of this model such as Discrete Cosine Transform (DCT) [26] provides an excellent work in representing images from different scales with small digits (e.g., 64 bits) to express the number of discrete data points. These data were evaluated in terms of the sum of cosine functions with different frequencies to convert it from the spatial domain to the frequency domain. Ring Partition and Invariant Vector Distance (RPIVD) is another shallow model introduced by [9]. They divided the image into rings and applied four statistical measures (mean, variance, skewness, and kurtosis) to each ring to extract the features. Both models are effective on image authentication based on CASIA [16], USC-SIPI [17] datasets but limited at SMPI dataset [19].

The authors in [12] provide a perceptual image hashing method by combining a statistical feature-based approach with visual perception using Watson's visual model theory. The statistical feature-based generated by extracting key-point-based features using the input image to scale-invariant feature transform (SIFT) algorithm. The visual perception is received using Watson's visual model to preserve sensitive features that are important for humans perceiving image content processing. The accuracy of this model overcame the [9], [26] on the same benchmark ground.

Machine Learning Approach– Learned algorithms on the other side are trending these years on image classification, retrieval, and authentication since this approach extracts better feature vectors. Reference [10] proposes a data-driven image fingerprinting algorithm based on a neural network approach with two training stages: pre-trained and fine-tuning. The first stage uses a Denoising Autoencoder (DAE) to restore a distorted image to its original state. There are 72 distorted images for each original image, including nine different operations, such as JPEG compression and Gaussian noise. Each function has different strength parameters that generated the 72 copies. The main goal of this network is to reduce the discrepancies between original and distorted images. The fine-tuning approach further reduces the overlap between the probability density curves of fingerprint distances calculated from perceptually identical and irrelevant image pairings. This method is akin to restricting fingerprints into a region similar to the original image for distorted images.

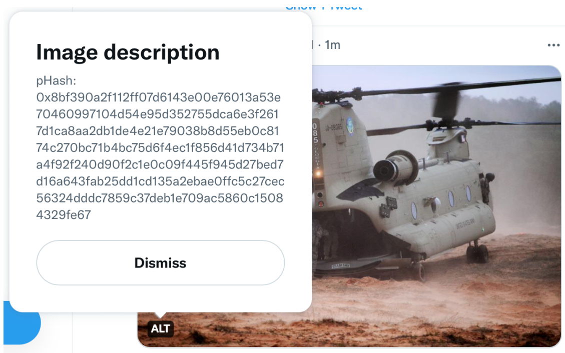


Image description

pHash:
0x8bf390a2f112ff07d6143e00e76013a53e
70460997104d54e95d352755dca6e3f261
7d1ca8aa2db1de4e21e79038b8d55eb0c81
74c270bc71b4bc75d6f4ec1f856d41d734b71
a4f92f240d90f2c1e0c09f445f945d27bed7
d16a643fab25dd1cd135a2ebae0ffc5c27cec
56324dddc7859c37deb1e709ac5860c1508
4329fe67

Fig. 2. Sample image posted on Twitter platform that has pHash on the Image description feature.

The authors of [11] introduce an image pHash scheme based on the CNN framework for feature extraction and a fully connected layer at the end of the network for final image hash sequence constructing. The CNN model contains five convolutional and five pooling layers, generating 256 feature vectors, and is reduced by the fully connected layer into 50 vectors. The proposed work added four constraints. The first two constraints are added at the feature map after processing convolutional layers, the ReLU layer, and max pooling by calculating the Mean Squared Error (MSE) of identical images with perceptually identical (distorted copies) and identical images with distinct pairs. The other two constraints went before final hash construction. All four constraints were added onto the total cost function with weight allocation. The 3,000 samples of the dataset for training are collected from COCO [27], where each image generates 64 distorted copies, and the distinct copies are paired with random images for a total of 405,000 images.

III. Methodology

In this section, we first discuss the assumptions to authenticate images over the social media platforms used in our research. Then, we explain the details of our proposed new system.

A. Application Scenario and Assumptions

To ensure security, the following assumptions are proposed. It is assumed that users must create accounts using their valid information to use our system. Naturally, Twitter accounts holders are verified by Twitter, Inc. The generated pHash of an image from our system can be added to the image through the description feature on Twitter's platform in hexadecimal representation (e.g., see Figure 2). End-users can download said image, copy its pHash from the description, and use our system for authentication. Moreover, users can re-publish the image with its pHash on their account. Adversaries here are forced to provide their credential information to our system in order to generate a new pHash. This assumption applies on other social media platforms, e.g., Facebook.

B. Proposed System

As shown in Figure 3 and Algorithm 1, the system design consists of four stages: pre-processing, feature extractor, contrastive loss, and pHash generator, respectively. We describe each stage below:

1) pre-processing: Before each image passes to the training phase, it goes through data augmentation, resizing, applying random color jitters, and random compression. Data augmentation improves performance when applied to deep learning models, as is shown in SimCLR [28]. SimCLR is a self-supervised learning model that uses data augmentation to generate two augmented images of each image in the batch and minimize the difference during the training task. Our model uses SimCLR architecture with crucial modification. We introduce a new step to the augmentation process to suit our target task. Instead of creating only two augmented images as it has been done in the original SimCLR, we, also, create a content-changing sample from the original image x called an *altered* image $\tilde{x}_{alt}$. The alteration is added to the image randomly selected from one of three image modification techniques *copy-move*, *splicing*, and *removal*. The copy-move ($cp-mv$) alteration is an operation of randomly copying a spot of an image with size of m×m×3, where m $\in$ {16:208} and randomly pasting it on a different location of the same image. The splicing (sp) is the same process as copy-move in randomization, but the spot is pasted on a different image. Finally, the removal (rm) alteration is where we follow the same technique of selecting a spot with a random size and a random location, but we apply kernel simple blurring filter 50 times on the same spot without moving it to a different location.

As in Figure 3, the augmentation process in this approach applied to original and altered images to cover multiple distorted versions. From these augmentations and alterations, each image is converted into two pairs after fixed resizing to 224×224×3: original x with random *augmented* $\tilde{x}$ and *altered* x_{alt} with random *augmented_altered* $\tilde{x}_{alt}$. The augmented refers to the version of the original image with content-preserving manipulation. On the other hand, altered represents the content-changing manipulation. Finally, augmented_altered is the copy of the altered image with content-preserving manipulation. The next stage in the training knows that each pair is authentic on its own and unauthentic in comparison to the other pair.

2) feature extractor: Each image is passed to a convolution neural network (ConvNet) as shown in Figure 3. ConvNet produces feature maps that capture image features. Next, these feature maps are flattened and mapped to a n-dimensional (n-dim) feature vector through a fully connected layer (FCL), (z, $\tilde{z}$, z_{alt}, and $\tilde{z}_{alt}$), where n-dim is a hyperparameter representing the number of nodes in the last layer chosen ahead of the training. The ConvNet used is ResNet-18 which consists of convolution layers, poling, ReLu, and skip connection.

3) Contrastive loss: At the training stage, the n-dim vector is passed to the loss. We use contrastive loss as used in [28] to maximize the agreement between the positive samples and minimize the agreement between the negative samples by minimizing the normalized temperature-scaled cross entropy loss (NT-Xent). We assign the temperature τ to 0.1

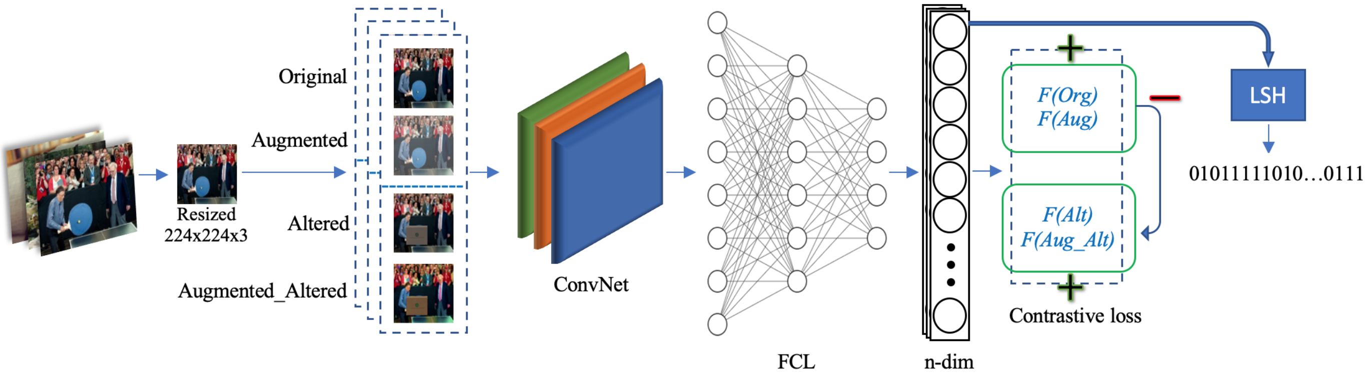


Fig. 3. Proposed approach for image authentication.

as suggested by [28]. SimCLR uses the augmented pair as positive samples. Negative samples are collected by pairing an image with another in the same batch that is not its augmented twin. However, our proposed approach takes the original image and its augmentation as a positive example, as well as the altered version with its augmentation. The negative sampling is the same as the original approach in SimCLR. Consequently, the original image will be paired with its altered version as a negative example to force the model to distinguish between images that shares high level features with content modification.

4) pHash generator: At the evaluation step, LSH is used in the hash generation to convert the long length of extracted features into small binary bits representations by mapping close feature vectors to buckets with similar hash values [29], [30]. Random projection is one type of LSH we used because it provides an independent secret key during random hyperplane generation that can be adaptive for security purposes. In practice, the feature vectors from FCL are flouting points with a length of 512 multiplied by a random hyperplane matrix of the size of 1024x512. This matrix multiplication is finally converted to a bit vector by applying a Heaviside step function to each element. The final generated hash length is 1024 bits.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

This section explores the experiment setup, the training configuration, and the main results based on F1-score metric.

A. Experimental Setup

To evaluate the robustness and effectiveness of the proposed scheme, we run a large number of experiments. Our implementation and training were done using NVIDIA GeForce RTX 3090 GPU. All other prior models were re-implemented and tested using the Colab platform based on best effort resources. We use PyTorch-lightning[1] open-source python library for our proposed system. Moreover, we re-implemented or used provided sources of other schemes and integrated a final hash generation using LSH for deep learning approaches.

Algorithm 1 Training Stage for the Proposed Model.

```
input: batch size $N \times 2$, constant $\tau$
network: ResNet-18 $(f)$ + FCL $(g)$
for randomly sample $x \in \{X\}$ do
    draw one attack $a \in \{cp-mv, spl, rm\}$
    $x_{alt} = a(x)$
    draw two augmentation functions $d \sim aug$, $\tilde{d} \sim aug$
    $\tilde{x} = d(x)$
    $\tilde{x}_{alt} = \tilde{d}(x_{alt})$
    # Forward Pass:
    $z = g(f(x))$
    $\tilde{z} = g(f(\tilde{x}))$
    $z_{alt} = g(f(x_{alt}))$
    $\tilde{z}_{alt} = g(f(\tilde{x}_{alt}))$
end for
Calculate contrastive Loss $L$
update the network parameters to minimize $L$
return network $f(.)$ and $g(.)$
```

B. Training Configuration

The configuration of the training goes through multiple processes. First, we collected 180,000 images from different resources, as shown in Table I. This diversity prevents bias to any image classes. Next, four modified samples were derived from each original image sample after resizing into 224×224×3 and paired into two groups. The first pair contains the original image and its augmented copy, and the second contains the altered and its augmented altered sample. Thus, the total number of training examples was increased to reach 720,000 images. We trained our model with contrastive loss to increase the agreement of similar images and decrease the agreement of dissimilar images. We used SMPI [19] as a benchmark for evaluating images that were collected from Facebook and Twitter platforms. In addition, we tested our model on the datasets IMD2020 [24] and COVERAGE [25] to compare it with other state-of-the-art models.

[1] https://www.pytorchlightning.ai/

TABLE I
THE STRUCTURE OF TRAINING AND VALIDATION SETS.

Stage	Dataset	no.
Training set	Flickr [31]	8,000
	Holopix50k [32]	41,000
	Tiny ImageNet [33]	100,000
	PS-Battles [18]	10,000
	ImageCLEF [34]	21,000
Validation set	SMPI [19]	19,458
	IMD2020 [24]	200
	COVERAGE [25]	200

C. Main Results

The similarity metric we used for our approach is Hamming Distance measurement, as used by [26], [28], [30]. The algorithm presented in [12] is the only one that used Euclidean distance. The pHash distance d between two images draws the line of the threshold θ that will be the indicator in our image authentication system. For similar images, the distance d should satisfy $0 \leq d \leq \theta$. For altered or dissimilar images d should be above the threshold θ, , i.e., $\theta < d$. Table II shows the best selected θ based on the best F1-score assessments.

Table II compares five schemes based on the SMPI dataset. The bold F1-scores in each column are the best-reported score, which shows the significant improvement using our proposed approach on both social media platforms. Overall, four models [12], [26], [28], [30] have close F1-score at Twitter with 0.87, 0.84, 0.88, and 0.88 respectively. Our proposed scheme reached the highest score by 0.99. In contrast, [12], [26], [28], [30] under-perform with Facebook with 0.70, 0.44, 0.82, and 0.82 respectively and a new high record achievement with the proposed model by 0.92.

We evaluated the performance of the models using the Receiver Operating Characteristic (ROC) curve as illustrates at Figure 4. The X-axis is the probability of False-Reject Rate (FRR), the ordinate is the probability of False-Accept Rate subtracted from one $(1 - FAR)$. An ROC curve that is closer to the top left corner means a better performance of content authentication. From ROC curves of the five schemes in Figure 4(a), we can observe that our scheme achieves the best ROC curve on the Facebook platform compared with the others, whereas model in [12] is the worst. Figure 4(b) represents the ROC curve on the Twitter scale shows small gaps in most models, where our proposed technique overpasses the others.

Moreover, we evaluated the proposed algorithm using other datasets IMD2020 [24] and COVERAGE [25] that are mainly founded for image forgery assessing on the scale of copy-move, splicing, and removal. We picked real-life manipulated images part from [24] that are collected from the Internet. Based on the same θ that we picked previously for our model, the Area Under Curve (AUC) performance comparisons is provided at Table III with other models [35]–[38]. Our model's results, which are in bold, are ahead of others by a high percentage.

TABLE II
F-SCORE RESULTS OF EACH MODEL USING SMPI DATASET AND THEIR THRESHOLD θ.

Model	threshold θ	Facebook	Twitter
DCT [26]	0.12	0.70	0.87
VisualModelBased [12]	6.46	0.44	0.84
SimCLR [28]	0.04	0.82	0.88
NuralHash [30]	0.02	0.82	0.88
Proposed	0.02	**0.92**	**0.99**

TABLE III
AREA UNDER CURVE (AUC) PERFORMANCE COMPARISON.

Model	IMD2020 [24]	COVERAGE [25]
CFA1 [35]	0.586	0.485
J-LSTM [36]	0.487	0.614
ManTra-Net [37]	0.748	0.819
TraFor-Self [38]	0.848	0.884
Proposed	**0.98**	**0.99**

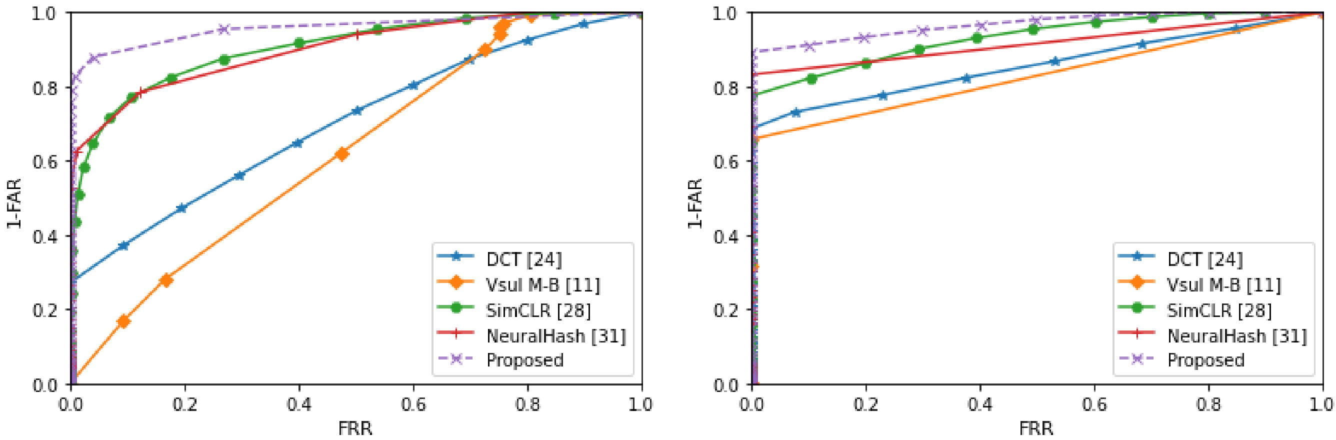
V. Discussion

Many magnificent works used deep learning for image classifications and were based on a large-scale dataset such as [39]. On the other hand, image authentication received less attraction due to multiple reasons. First, most image datasets are generated with big alterations; therefore, many developed systems accomplished high accuracy on those datasets. second, small alteration to the image is hard for the systems to detect, and the concept of pHash authentic distorted copies of the original image. Third, distorted copies of the original image have unlimited and unknown factors. For instance, the compression quality factor during the model design is fixed, i.e., quality factor $\in \{1, 5, 10, 30, 50, 70, 90, 95\}$ and the tested images are compressed with random and unknown values 1:100. This example is one case where nine other image operations can be implemented on an image and considered as an authentic copy, not counting that one image might receive multiple operations.

For instance, Facebook deals with each user differently during exploring the platform because Facebook needs to allow their users with weak network coverage to use their platform by applying different compression quality factors based on the network status. For example, we examined downloading a shared image from the same post by two PCs, we received different sizes of the same image. This alteration itself makes the task of authentication complex.

In addition, the length of pHash plays a significant part in image authentication. The larger is better to make the extracted feature vectors more sensitive. In contrast, the larger length would cause more overhead on the payload of the image in practice. Therefore, we remain our evaluation on 1024-bits to make it more applicable to various applications.

Finally. we looked into another direction to enhance our proposed model. We increased the number of altered and augmented version of each image to cover more samples in the batch. Therefore, the batch consists mostly of one image and its n-altered and n-augmented versions. The results showed degrading in the performance of the model. We concluded that



(a) Facebook.

(b) Twitter.

Fig. 4. Comparison of ROC curves for each authentication model using SMPI dataset.

increasing the samples of the original image in the same batch harm the model efficiency. Therefore, we limited the model to have only four samples from the original image. These samples are original, augmented version, altered and augmented of the altered version.

VI. Conclusion and Future Work

In this paper, we introduced a robust authentication system targeting image authentication on social media networks. We built an alteration generator that simulates the three real-world image alteration attacks (copy-move, splicing, and removal). Also, a new data augmentation technique was included during the training process to generate distorted copies of the original. The localization and modification applied on the images were small scaled as minimum as just 1%. Each image gathered with its augmentations in a batch as pairs for self-supervised learning using ResNet-18 network with a contrastive loss. The proposed system achieves robust authentication on different datasets SMPI [19], IMD2020 [24] and COVERAGE [25] with highest F1-score comparing with other works. Further development can be added on studying the robustness of the proposed model under various attacks including adversarial examples.

References

[1] Y. Wang, F. Tahmasbi, J. Blackburn, B. Bradlyn, E. De Cristofaro, D. Magerman, S. Zannettou, and G. Stringhini, “Understanding the use of fauxtography on social media,” 2020.

[2] D. Evon. (2022) Hey ‘crypto bro’s,’ that mcdonald’s billboard is fake. [Online]. Available: https://www.snopes.com/fact-check/crypto-mcdonalds-billboard/

[3] D. Mikkelson. (2002) Does this photograph show president bush reading a book upside-down? [Online]. Available: https://www.snopes.com/fact-check/bush-upside-book/

[4] Twitter, “How we address misinformation on twitter,” https://help.twitter.com/en/resources/addressing-misleading-info, May 2022.

[5] L. Atzori, S. Corona, and D. D. Giusto, “Error recovery in jpeg2000 image transmission,” vol. 3, pp. 1733–1736, 2001.

[6] N. I. Korsunov and D. A. Toropchin, “Recognition method of near-duplicate images based on the perceptual hash and image key points using,” vol. 1, pp. 261–264, 2015.

[7] T. website”. (2022) ”what is tineye”. [Online]. Available: ”https://tineye.com”

[8] F. Sabahi, M. O. Ahmad, and M. N. S. Swamy, “Content-based image retrieval using perceptual image hashing and hopfield neural network,” pp. 352–355, 2018.

[9] Z. Tang, X. Zhang, X. Li, and S. Zhang, “Robust image hashing with ring partition and invariant vector distance,” vol. 11, no. 1, pp. 200–214, 2016.

[10] Y. Li, D. Wang, and L. Tang, “Robust and secure image fingerprinting learned by neural network,” vol. 30, no. 2, pp. 362–375, 2020.

[11] C. Qin, E. Liu, G. Feng, and X. Zhang, “Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints,” vol. 31, no. 11, pp. 4523–4537, 2021.

[12] K. Wang, Xiaofeng *et al.*, “A visual model-based perceptual image hash for content authentication,” vol. 10, no. 7, pp. 1336–1349, 2015.

[13] W. Zhen-kun, Z. Wei-zong, Ouyang-Jie, L. Peng-fei, D. Yi-hua, Z. Meng, and G. Jin-hua, “A robust and discriminative image perceptual hash algorithm,” pp. 709–712, 2010.

[14] C.-P. Yan, C.-M. Pun, and X.-C. Yuan, “Quaternion-based image hashing for adaptive tampering localization,” vol. 11, no. 12, pp. 2664–2677, 2016.

[15] Y. Zhao, S. Wang, X. Zhang, and H. Yao, “Robust hashing for image authentication using zernike moments and local features,” vol. 8, no. 1, pp. 55–63, 2013.

[16] Y. Zheng”. (2021) ”casia dataset”. [Online]. Available: ”https://ieee-dataport.org/open-access/modified-casia#files”

[17] U. University”. (1977) ”usc-sipi image database”. [Online]. Available: ”https://sipi.usc.edu/database/”

[18] S. Heller, L. Rossetto, and H. Schuldt, “The PS-Battles Dataset – an Image Collection for Image Manipulation Detection,” vol. abs/1804.04866, 2018.

[19] 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)*, 2022, pp. 149–156.

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” 2013.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[22] A. Gionis, P. Indyk, R. Motwani *et al.*, “Similarity search in high dimensions via hashing,” in *Vldb*, vol. 99, no. 6, 1999, pp. 518–529.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” vol. 115, no. 3, pp. 211–252, 2015.

[24] B. Novozamsky, Adam *et al.*, “Imd2020: A large-scale annotated dataset tailored for detecting manipulated images,” pp. 71–80, March 2020.

[25] B. Wen, Y. Zhu, R. Subramanian, T.-T. Ng, X. Shen, and S. Winkler, “Coverage — a novel database for copy-move forgery detection,” pp. 161–165, 2016.

[26] L. Du, A. T. S. Ho, and R. Cong, “Perceptual hashing for image authentication: A survey,” vol. 81, 2020.

[27] 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,” pp. 740–755, 2014.

[28] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” pp. 1597–1607, 2020.

[29] A. Gionis, P. Indyk, R. Motwani *et al.*, “Similarity search in high dimensions via hashing,” vol. 99, no. 6, pp. 518–529, 1999.

[30] L. Struppek, D. Hintersdorf, D. Neider, and K. Kersting, “Learning to break deep perceptual hashing: The use case neuralhash,” 2021.

[31] M. Hodosh, P. Young, and J. Hockenmaier, “Framing image description as a ranking task: Data, models and evaluation metrics,” vol. 47, pp. 853–899, 2013.

[32] Y. Hua, P. Kohli, P. Uplavikar, A. Ravi, S. Gunaseelan, J. Orozco, and E. Li, “Holopix50k: A large-scale in-the-wild stereo image dataset,” *arXiv preprint arXiv:2003.11172*, 2020.

[33] Y. Le and X. Yang, “Tiny imagenet visual recognition challenge,” vol. 7, no. 7, p. 3, 2015.

[34] H. Müller, P. Clough, T. Deselaers, B. Caputo, and I. CLEF, “Experimental evaluation in visual information retrieval,” *The Information Retrieval Series*, vol. 32, pp. 1–554, 2010.

[35] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, “Image forgery localization via fine-grained analysis of cfa artifacts,” vol. 7, no. 5, pp. 1566–1577, 2012.

[36] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, “Exploiting spatial structure for localizing manipulated image regions,” pp. 4970–4979, 2017.

[37] Y. Wu, W. AbdAlmageed, and P. Natarajan, “Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features,” pp. 9543–9552, 2019.

[38] J. Hao, Z. Zhang, S. Yang, D. Xie, and S. Pu, “Transforensics: Image forgery localization with dense self-attention,” pp. 15 055–15 064, 2021.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” pp. 248–255, 2009.