DRAG: Dynamic Region-Aware GCN for Privacy-Leaking Image Detection

Guang Yang[1,2], Juan Cao[1,2], Qiang Sheng[1,2], Peng Qi[1,2], Xirong Li[3], Jintao Li[1]

[1] Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
{gyang, caojuan, shengqiang18z, qipeng, jtli}@ict.ac.cn, xirong@ruc.edu.cn

Abstract

The daily practice of sharing images on social media raises a severe issue about privacy leakage. To address the issue, privacy-leaking image detection is studied recently, with the goal to automatically identify images that may leak privacy. Recent advance on this task benefits from focusing on crucial objects via pretrained object detectors and modeling their correlation. However, these methods have two limitations: 1) they neglect other important elements like scenes, textures, and objects beyond the capacity of pretrained object detectors; 2) the correlation among objects is fixed, but a fixed correlation is not appropriate for all the images. To overcome the limitations, we propose the **D**ynamic **R**egion-**A**ware **G**raph Convolutional Network (**DRAG**) that dynamically finds out crucial regions including objects and other important elements, and models their correlation adaptively for each input image. To find out crucial regions, we cluster spatially-correlated feature channels into several region-aware feature maps. Further, we dynamically model the correlation with the self-attention mechanism and explore the interaction among the regions with a graph convolutional network. The DRAG achieved an accuracy of 87% on the largest dataset for privacy-leaking image detection, which is 10 percentage points higher than the state of the art. The further case study demonstrates that it found out crucial regions containing not only objects but other important elements like textures. The code and more details are in https://github.com/guang-yanng/DRAG.

Introduction

Social media like Facebook has been part of our daily life. People post a large number of images on social media to record and share their lives. However, the convenience of online image sharing brings about the risk of privacy leakage. The shared images contain rich information like personal relationships and physical disability (Orekondy, Schiele, and Fritz 2017). Malicious use of such information has been documented (Solsman 2020) causing dire consequences like fraud and cyber violence (Equifax 2020). As a severe issue that is close to our daily life, privacy-leaking image is attracting increasing concerns.

Social media platforms allow users to set privacy preferences like the visibility of their content to protect privacy, but

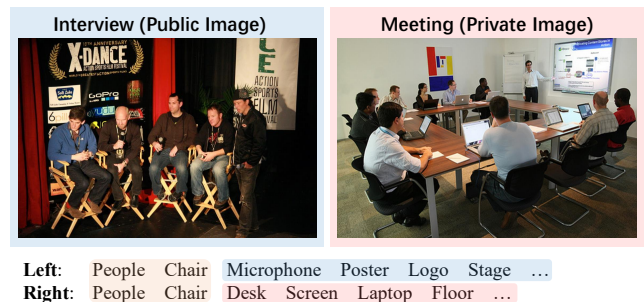| **Left**: | People | Chair | Microphone | Poster | Logo | Stage | … |
| **Right**: | People | Chair | Desk | Screen | Laptop | Floor | … |

Figure 1: Example of images that people share online in the Image Privacy dataset (Yang et al. 2020). (a) is a public image that is safe for sharing, while (b) is a private image that may leak sensitive information. Public and private images contain many common elements as well as specific ones. The co-occurrence and interaction of the elements provide semantic clues of scenes, activities, etc., which is crucial for privacy-leaking image detection. Therefore, methods for this task need to find out the elements and take their co-occurrence and interaction into consideration.

many users still unconsciously share images that may leak privacy. Although people have common expectations about the privacy setting of online images (Hoyle et al. 2020), they often lack the awareness of the privacy risk of the shared images (Tuunainen, Pitkänen, and Hovi 2009; Wang et al. 2011). Liu et al. (2011) proved that there is a gap between users' expectations and the reality of users' privacy settings of the shared images. The above phenomenon and the potential harms make it urgent to help users reduce the privacy risks during image sharing. Users may unintentionally share images that leak privacy and the spread of the images is almost uncontrollable. Therefore, a feasible method to reduce the privacy risks is to automatically identify images that may leak privacy and give warnings to users before sharing.

We use *private images* to refer to images that may leak privacy and *public images* to refer to images that are safe for sharing. Researchers mainly consider non-personalized consensus and build corresponding datasets, and several examples are presented in Fig. 1. Following Tran et al. (2016) and Yang et al. (2020), we formulate privacy-leaking image detection as a binary classification task (i.e., predict whether

a given image is *private* or *public*). Fig. 1 demonstrates that the interaction among the elements in the images provides clues and helps distinguish between private and public images. Yang et al. focuses on objects and their correlation to identify private images based on object detection. However, they neglect other important elements like scenes (Tonge and Caragea 2019), textures, and objects beyond the capacity of pre-trained object detectors. Furthermore, the correlation among objects is fixed, but the elements vary in different images, making the fixed correlation inappropriate.

To overcome the limitations, we propose **D**ynamic **R**egion-**A**ware **G**raph Convolutional Network (**DRAG**) to dynamically find out regions of the crucial elements, and model their correlation adaptively per image. The workflow of DRAG is presented in Fig. 2, which contains two main parts. In the first part (Fig. 2 (2)), the DRAG finds out N crucial regions from the feature map without the reliance on the object detectors. Specifically, based on the feature map obtained from the backbone, the DRAG clusters the spatially-correlated feature channels into N region-aware feature maps. In the second part (Fig. 2 (3)), DRAG adopts the graph convolutional network (GCN) to model the interaction among the N regions. The regions are obtained dynamically for each image, and thus the correlation among the regions should be adaptive rather than predefined and fixed. We dynamically model the correlation with the self-attention mechanism to initialize the correlation matrix for GCN. Then the interaction among the N crucial regions is explored by propagating corresponding features through GCN with the adaptive correlation matrix. Finally (Fig. 2 (4)), the propagated features are concatenated with a global representation of the image to identify private images. Compared with existing works, the dynamic nature of the DRAG enables it to find out more diverse elements (not only objects) and model their correlations adaptively.

Our main contributions are summarized as follows:

(1) We proposed a novel framework DRAG for privacy-leaking image detection. The DRAG dynamically finds out crucial regions without the limitation of pretrained object detectors and models the correlation among crucial regions adaptively for each image.

(2) To explore the interaction among the crucial regions, we proposed a region-aware method to initialize the graph for GCN based on spatially-correlated channels clustering and self-attention mechanism.

(3) The experimental results prove the effectiveness of the proposed framework for privacy-leaking image detection. The DRAG that only utilizes visual features outperformed existing methods, including visual-based and multi-modal ones.

Related Work
Privacy-Leaking Image in Online Image Sharing
Liu et al. (2020) concluded several privacy issues of online image sharing. In this paper, we focus on the unawareness of privacy during image sharing. There are two main types of methods to deal with the risk of unawareness of privacy.

The first type of method mainly adopts classification models to identify private images. Zerr, Siersdorfer, and Hare (2012) proposed a privacy-aware classifier based on visual features like face and color histograms. Buschek et al. (2015) proposed a multi-modal method that assigns privacy labels to the images based on visual features and metadata like location and publication time. Tonge, Caragea, and Squicciarini (2018) utilized another kind of metadata, tag, and Tonge and Caragea (2019) further derived features of the object, scene, and tags for privacy-leaking image detection. Yang et al. (2020) extracted a knowledge graph from the images and identified private images based on object detection and graph neural networks.

The second type of method focuses on sensitive regions in the images, including approaches like object detection and semantic segmentation. Some detected private attributes such as faces (Sun, Wu, and Hoi 2018), license plates (Zhou et al. 2012), and social relationship (Li et al. 2017a). Orekondy, Schiele, and Fritz (2017) defined a list of privacy attributes and detected them simultaneously. Some works attempted to protect privacy-leaking image based on blurring (Fan 2018), blocking (Li et al. 2017b), cartooning (Hasan et al. 2017), and perturbation (Oh, Fritz, and Schiele 2017). Shetty, Fritz, and Schiele (2018) removed private objects from the images based on a generative method. However, a person may be recognized even his face is not visible (Oh et al. 2016), and the redacted image may be recovered (Shen et al. 2019). As the usage of shared images is almost uncontrollable, it is better to prevent the risk from the beginning. Therefore, we follow the first type of method to solve the issue of privacy-leaking images by classification.

Graph-based Methods in Visual Tasks
Graph-based methods have shown great potential in many vision tasks in recent years, including visual question answering (Teney, Liu, and van den Hengel 2017), person re-identification (Wu et al. 2019), multi-label image recognition (Marino, Salakhutdinov, and Gupta 2017), and relationship recognition (Wang et al. 2018). Ye et al. (2020) utilized GCN (Kipf and Welling 2017) for multi-label classification. Yang et al. (2020) adopted graph neural networks for privacy-leaking image detection and showed that modeling the interaction among crucial elements is an effective way. However, their framework only focuses on the objects that the pretrained object detector can recognize. Inspired by Ye et al. (2020) and Yang et al. (2020), we proposed DRAG that can model the interaction among more crucial elements dynamically with GCN. Furthermore, instead of a fixed correlation matrix for all the images (Yang et al. 2020), we extract the correlation matrix for each input image adaptively based on the self-attention mechanism.

Approach
Overview of DRAG
The DRAG dynamically finds out crucial regions and models their correlation adaptively for each input image. Then the DRAG explores the interaction among the crucial regions with GCN and identifies the private images.
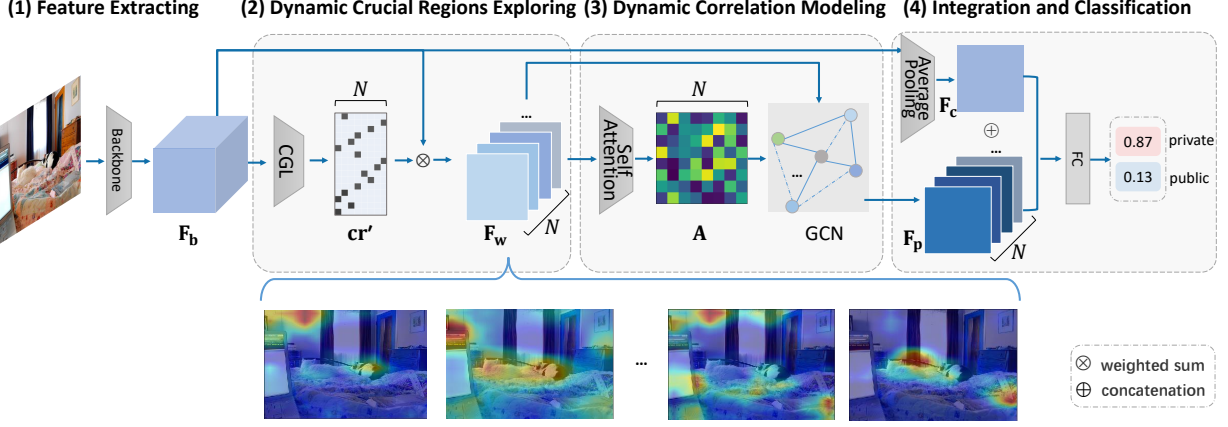
Figure 2: Workflow of the Dynamic Region-Aware GCN (DRAG) for privacy-leaking image detection. (1) DRAG first extracts the feature $\mathbf{F_b}$ of the input image. (2) The channels of $\mathbf{F_b}$ are then clustered into N groups with Channel Grouping Layer (CGL). According to the approximate clustering result $\mathbf{cr'}$, $\mathbf{F_b}$ are aggregated into N feature maps $\mathbf{F_w}$ to represent N differentiated regions (Examples are at the bottom). (3) DRAG formulates a graph with $\mathbf{F_w}$ as the N nodes and uses the self-attention mechanism to obtain the correlation matrix $\mathbf{A}$. Then a GCN is used for feature learning on this graph. (4) The learned feature $\mathbf{F_p}$ is concatenated with a global representation of the image $\mathbf{F_c}$ to identify private images.

Specifically, the DRAG contains two main parts (see Fig. 2). In the first part (Fig. 2 (2)), the DRAG extracts diverse and tiny clues and then clusters them to obtain region-aware feature maps as the representation of crucial regions, without the reliance on the object detectors. In the second part (Fig. 2 (3)), the DRAG models the correlation among the crucial regions based on the self-attention mechanism and initializes the correlation matrix for GCN. Then the features of crucial regions are propagated through GCN with the adaptive correlation matrix to explore the interaction among these regions. Finally (Fig. 2 (4)), the propagated features are concatenated with a global representation of the image to classify a given image as private or public.

Dynamic Crucial Regions Exploring

Tasks like object detection and fine-grained image recognition need to focus on objects for better performance. For example, the Region Proposal Network (Ren et al. 2015) is used to select regions that may contain objects, attention-based methods (Fu, Zheng, and Mei 2017) are used to focus on the details of objects for fine-grained image classification. However, the clues for privacy-leaking image detection are revealed not only by the objects but also by other elements such as scenes and textures . To focus on these crucial elements, we find out differentiated regions in an image based on the channel grouping mechanism (Zheng et al. 2017).

We first trained a backbone (here, ResNet (He et al. 2016)) and got the convolutional feature of the input image $\mathbf{F_b} \in \mathcal{R}^{C \times H \times W}$, where W, H, and C are the width, height, and channel number of the feature. According to Simon and Rodner (2015), the peak responses of the channels correspond to various visual patterns. Following Zheng et al. (2017), we clustered the channels by K-means (Mac-

Queen et al. 1967) according to spatial correlation among the corresponding peak responses and adopted the cluster results as the representation of crucial regions. To combine the clustering with neural networks, the clustering process was approximated by several fully-connected layers (FCs), and the details are as follows:

A group of channels whose peak responses appear in neighboring locations were clustered together. For each channel, we got the coordinates of the peak response $[t_x, t_y]$ of all training images and formulated them into a vector: $\left[t_x^1, t_y^1, t_x^2, t_y^2, \ldots, t_x^{\Omega}, t_y^{\Omega}\right]$, where t_x^i and t_y^i are the coordinates of peak response of the ith training image, and Ω refers to the size of the training set. This vector was used as the feature for clustering with K-means. The channels were clustered into N groups to represent N differentiated regions. The clustering results were formulated as a matrix $\mathbf{cr} \in \mathcal{R}^{N \times C}$ with $cr_{ij} = 1$ or 0, which indicates that if the jth channel belongs to the ith group (i.e., the ith region).

The pretrained backbone initially focuses on the objects and will be fine-tuned to adapt to privacy-leaking image detection. To let the clustering results obtained from the backbone be optimized together, we adopted FCs to approximate the clustering process, which is called the channel grouping layer (CGL). CGL takes the feature map $\mathbf{F_b}$ as input, then outputs the estimated result of clustering $\mathbf{cr'} \in \mathcal{R}^{N \times C}$:

$$\mathbf{cr'} = CGL(\mathbf{F_b}) = sigmoid(FCs(\mathbf{F_b})), \quad (1)$$

We used the $\mathbf{cr'}$ to get the feature of each region with an averaged weighted sum mechanism. For the ith region, its corresponding feature $\mathbf{F_{wi}} \in \mathcal{R}^{H \times W}$ was obtained by :

$$\mathbf{F_{wi}} = \frac{1}{C} \sum_c \mathbf{F_{bc}} * cr'_{ic}, \quad (2)$$

where C is the number of channels in $\mathbf{F_b}$, $\mathbf{F_{bc}}$ is the feature of the cth channel. $\mathbf{cr'_{ic}}$ is the estimated indicator that if the cth channel belongs to the ith region. By concatenating features of all regions, we finally obtained the $\mathbf{F_w} \in \mathcal{R}^{N \times H \times W}$.

Initialization To obtain a proper initialization, CGL was pretrained to let the $\mathbf{cr'}$ be as close to $\mathbf{cr}$ as possible, and the details are described in the Experiments section. During the joint learning, we adopted two losses, $Dis(\cdot)$ and $Div(\cdot)$, to force the CGL to learn differentiated regions as follows:

$$Dis(\mathbf{F_w}) = \sum_{i \in N} \sum_{(x,y) \in \mathbf{r_i}} \mathbf{F_{wi}}(x,y)^2 \left[\|x - t_{ix}\|^2 + \|y - t_{iy}\|^2 \right],$$

$$Div(\mathbf{F_w}) = \sum_{i \in N} \sum_{(x,y) \in \mathbf{r_i}} \mathbf{F_{wi}}(x,y)^2 \left[\max_{j \neq i} \mathbf{F_{wj}}(x,y) - mrg \right]^2, \quad (3)$$

where i refers to the ith region, t_{ix} and t_{iy} are the coordinates of peak response in the ith region. mrg is the mean of all the values in feature map $\mathbf{F_w}$, which represents a margin to make $Div(\cdot)$ less sensitive to noise. The $Dis(\cdot)$ encourages a compact distribution in the feature of a region (i.e., similar visual patterns from a specific part to be grouped together), while the $Div(\cdot)$ forces the model to learn diverse regions rather than similar ones. Such constraints make the CGL learn differentiated regions for image privacy detection.

Dynamic Correlation Modeling

We obtained the feature of several crucial regions $\mathbf{F_w}$ based on CGL. To explore the interaction among these crucial regions for privacy-leaking image detection, we formulated a graph with regions as nodes and correlation among the regions as the edges to take the advantage of GCN. The regions were obtained dynamically for each image, and thus the correlation should be adaptive rather than predefined and fixed. Inspired by Ye et al. (2020) and Vaswani et al. (2017), we proposed a dynamic way to get an adaptive correlation matrix for GCN.

To model the correlation among the N crucial regions, we adopted the self-attention mechanism (Vaswani et al. 2017) which is widely used in NLP tasks to learn the correlation among words. We first got three vectors query (q), key (k), and value (v) from $\mathbf{F_{wi}}$ for each region r_i with three fully-connected layers. For all the N regions, the matrices $\mathbf{Q} \in \mathcal{R}^{\mathbf{N} \times \mathbf{d_k}}$, $\mathbf{K} \in \mathcal{R}^{N \times d_k}$ and $\mathbf{V} \in \mathcal{R}^{N \times N}$ were calculated by:

$$\mathbf{Q} = \mathbf{W_q} \mathbf{F}_w + \mathbf{b_q}, \mathbf{K} = \mathbf{W_k} \mathbf{F}_w + \mathbf{b_k}, \mathbf{V} = \mathbf{W_v} \mathbf{F}_w + \mathbf{b_v}, \quad (4)$$

where d_k is the dimension of both q and k, and $\mathbf{W}$ and $\mathbf{b}$ refer to the weights and biases in fully-connected layers, respectively. The results of self-attention $\mathbf{A}$ was given by:

$$\mathbf{A} = Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right)\mathbf{V}. \quad (5)$$

Each value in the matrix $\mathbf{A} \in \mathcal{R}^{N \times N}$ is obtained by considering one region and all other ones. As a result, $\mathbf{A}$ is able to represent the correlation among the N regions.

Feature Integration and Classification

We adopted GCN with the activation function of $ReLU$ to explore the interaction among the crucial regions, which propagates features through the nodes as follows:

$$GCN(\mathbf{X}) = ReLU(\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{X} \Theta), \quad (6)$$

where $\mathbf{I}$ refers to an identical matrix, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ denotes the adjacency matrix with inserted self-loops, $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$ is the diagonal degree matrix, and Θ is the learned weights. To avoid over-smoothing of node features, we only adopted two GCN layers and finally got the propagated feature $\mathbf{F_p} \in \mathcal{R}^{N \times H \times W}$:

$$\mathbf{F_p} = GCN(GCN(\mathbf{F_w})). \quad (7)$$

To prevent that the learned regions neglect important global information in the image, we got a compressed feature $\mathbf{F_c} \in \mathcal{R}^{1 \times H \times W}$ from the original feature map $\mathbf{F_b}$ by average $\mathbf{F_b}$ through the channels. At last, the $\mathbf{F_c}$ was concatenated with $\mathbf{F_p}$ for classification with a fully-connected layer FC and the activation function of $softmax$:

$$\hat{y} = softmax(FC(\mathbf{F_c} \oplus \mathbf{F_p})), \quad (8)$$

where $\oplus$ denotes the concatenating operation. The output $\hat{y}$ represents the probability that if the input image is private.

Experiments

Experimental Setup

Datasets

PicAlert PicAlert (Zerr et al. 2012) is the first dataset for privacy-leaking image detection on social media, which was built on an average community notion of privacy. They first crawled images from image-sharing social media Flickr, then asked external viewers to judge the privacy of the photos via a social annotation game. After removing invalid annotations, they finally proposed a dataset of images with user-classified privacy labels. The PicAlert we used contains 7,518 private images and 24,615 public images.

Image Privacy The PicAlert is somewhat biased as most of the private images in PicAlert is person-containing. To diversify private images, Yang et al. (2020) extended PicAlert to include more types of images reported in the previous study (Tran et al. 2016), such as driver licenses, ID cards, and legal documents. The Image Privacy dataset contains 13,910 private images and 24,615 public images.

Methods for Comparison We compared DRAG with state-of-the-art methods, including Privacy-CNH (Tran et al. 2016) and GIP (Yang et al. 2020) that only utilize visual information obtained from the images, as well as Combination of Object, Scene, and User Tags (Tonge, Caragea, and Squicciarini 2018) and DMFP (Tonge and Caragea 2019) that utilize extra user tags.

---

https://www.flickr.com/

Tags that users annotate when sharing images, often contain information that cannot be obtained from the image.

Privacy-CNH (Hereafter, PCNH) (Tran et al. 2016) proposed a framework that utilized both object and convolutional features for privacy-leaking image detection. The features are finally concatenated for classification.

GIP (Yang et al. 2020) is the first to adopt graph neural networks for privacy-leaking image detection. The GIP first detects objects in an image based on Faster-RCNN (Ren et al. 2015), and propagates the object features through a predefined graph which is extracted from the training set.

Combination of Object, Scene, and User Tags (Hereafter, Combination) (Tonge, Caragea, and Squicciarini 2018) combines object tags, scene tags, and user tags for privacy-leaking image detection, which is a basic multi-modal method. The object tags and scene tags are extracted from the visual features, while the user tags are extra collected.

DMFP (Tonge and Caragea 2019) is also a multi-modal method that utilizes object features, scene features, and tag features instead of the tags. The DMFP estimates the competence of the modalities and fuses the decisions dynamically. DMFP-O and DMFP-S denote DMFP that only utilize object features and scene features, respectively.

Implementation We conducted experiments on the two datasets to compare with state-of-the-art methods for privacy-leaking image detection. To make a fair comparison, we adopted the same experiment settings as Tonge and Caragea (2019) and Yang et al. (2020). The ratio of train, val, and test set is 15:7:10 in both datasets. The public and private images are in the ratio of about 3:1 in PicAlert and about 7:4 in Image Privacy.

The models were implemented with python 3.6.8, PyTorch 1.4.0 (Paszke et al. 2019), torchvision 0.5.0 (Marcel and Rodriguez 2010), and torch-geometric 1.6.1 (Fey and Lenssen 2019). We first pretrained a ResNet as the backbone model. We extracted the feature outputted by the last convolutional layer and obtained $\mathbf{F_b} \in \mathcal{R}^{2048 \times 14 \times 14}$. We clustered the channels into different regions following the process described in the Approach section and got the clustering result $\mathbf{cr} \in \mathcal{R}^{N \times 2048}$. We experimented with several region numbers N, including $4, 6, 8, 10$, and 12, to explore the influence of different region numbers. The clustering result $\mathbf{cr}$ was used to pretrain the CGL to let the $\mathbf{cr'}$ be as close to $\mathbf{cr}$ as possible, to enable the CGL to learn differentiated regions. We calculated the cross-entropy loss of all the N regions and 2048 channels to optimize the CGL:

$$L_{CGL} = -\sum_{i \in N} \sum_{j \in C} [y_{ij} \log (\hat{y_{ij}}) + (1 - y_{ij}) \log (1 - \hat{y_{ij}})], \quad (9)$$

where y_{ij} is the true label of the jth channel in the ith group, and $\hat{y_{ij}}$ is the corresponding predicted probability.

The correlation matrix $\mathbf{A}$ used for GCN was learned during training with the self-attention mechanism. For the self-attention module, the dimension of q and k was 64, while the dimension of v was N. For the GCN, the number of nodes was the same as the number of regions N. Feature of each region was used to initialize the corresponding node, and thus the feature of each node i was $\mathbf{F_{wi}} \in \mathcal{R}^{14 \times 14}$. After exploring the interaction among nodes, the GCN outputted $\mathbf{F_p} \in \mathcal{R}^{N \times 14 \times 14}$. Finally, by concatenating the global feature $\mathbf{F_c} \in \mathcal{R}^{1 \times 14 \times 14}$, $(\mathbf{F_c} \oplus \mathbf{F_p}) \in \mathcal{R}^{(N+1) \times 14 \times 14}$ was

used for classification. For the binary classification task, we adopted cross-entropy loss function:

$$L_{cls} = -\sum_i [y_i \log (\hat{y}_i) + (1 - y_i) \log (1 - \hat{y}_i)], \quad (10)$$

where y_i is the true label of the ith sample, and $\hat{y}_i$ is the corresponding predicted probability given by the model. The final loss function was:

$$L = L_{cls} + Dis(\cdot) + Div(\cdot). \quad (11)$$

We adopted *Adam* (Kingma and Ba 2015) as the optimizer with the weight decay of $1e - 7$. The L_{cls} and $(Dis(\cdot) + Div(\cdot))$ were optimized alternately. The learning rate was set to be $1e - 3$ for CGL, $1e - 3$ for GCN and $1e - 5$ for backbone during initialization. The models were trained with the same strategy: pretrain the backbone; pretrain the CGL; optimize the CGL; optimize the GCN for several epochs; optimize the GCN and the backbone; optimize the CGL again if necessary; fine-tune the backbone and the GCN for several epochs until convergence. Please refer to the source code for more details.

Experimental Results

Comparison with the State of the Art Following previous works, we compared DRAG with state-of-the-art methods on PicAlert and present the precision, recall, and F1 score of each class to validate the effectiveness. We select the models with the best performances on the validation set and report their performances on the test set. The comparisons between the DRAG and the state of the art are presented in Table 1. Note that Combination (Tonge, Caragea, and Squicciarini 2018) and DMFP (Tonge and Caragea 2019) are multi-modal methods that utilize extra user tags, while other methods only utilize the visual information obtained from the images. As the DRAG only utilizes visual information, we further compare with the state-of-the-art visual-based method (i.e., GIP) on the more challenging Image Privacy dataset.

We get several observations from the results. The DRAG outperforms other methods in most metrics on both datasets, which proves the effectiveness of the proposed framework. The accuracy and F-1 score of the DRAG is higher than all other methods, including visual-based and multi-modal ones. Specifically, the performances in the public class are similar for all methods, and the main difference lies in the private class, which is also the class that we need to pay more attention to. The DRAG achieved the highest F-1 score and also the highest recall in the private class, which means that the DRAG will significantly reduce the false-negative rate. Considering the practical task of privacy-leaking image detection, this phenomenon means that fewer private images will be classified as public incorrectly, and thus the DRAG will better help reduce the unintentional sharing of private images compared with other methods.

We observe that the DRAG achieved much better performance compared with GIP, especially on the harder Image Privacy dataset. We analyzed the rationality as follows, and we provide the corresponding Precision-Recall Curve in the

Dataset	Model	Source	Accuracy	Private			Public		
				Precision	Recall	F-1	Precision	Recall	F-1
PicAlert	PCNH	AAAI (2016)	83.15%	0.689	0.514	0.589	0.862	<u>0.929</u>	0.894
	GIP	PR (2020)	83.49%	0.552	<u>0.684</u>	0.610	**0.922**	0.871	0.895
	*Combination	AAAI (2018)	83.09%	0.671	0.551	0.605	0.869	0.912	0.892
	*DMFP	WWW (2019)	<u>86.36%</u>	**0.752**	0.627	<u>0.684</u>	0.891	**0.936**	<u>0.913</u>
	DRAG	Ours	**86.84%**	<u>0.719</u>	**0.719**	**0.719**	<u>0.914</u>	0.914	**0.914**
Image Privacy	GIP	PR (2020)	77.09%	**0.812**	0.751	0.780	0.730	0.795	0.761
	DRAG	Ours	**87.68%**	0.811	**0.842**	**0.826**	**0.914**	**0.895**	**0.905**

Table 1: Comparison with the state of the art. The best and second-best results in each column are boldfaced and underlined, respectively. "*" indicates multi-modal methods that utilize extra user tags, while other methods only utilize the visual information obtained from the images.

supplementary. First, as described in the Dataset section, the public images are the same in the two datasets, while Image Privacy contains more images in the private class. Therefore, Image Privacy is more balanced than PicAlert, and the performances in the public class of both methods dropped on Image Privacy. Second, objects are important clues for privacy-leaking image detection, and thus GIP that relies on the object detector performed well on PicAlert. But when dealing with a more complex dataset, the pretrained object detector limits the ability to focus on other crucial elements like unseen objects, scenes, and textures. Compared with GIP, the DRAG dynamically focus on regions of the crucial elements and thus achieved better performances in both classes. Our ablation studies in the next subsection also suggest that the model needs to pay more attention to differentiated regions for privacy-leaking image detection on Image Privacy than on PicAlert.

Ablation Study

Effectiveness of Dynamic Crucial Regions Exploring To obtain a variant without the ability to dynamically explore crucial regions, we fixed the CGL with its initial features that mainly focus on objects, because the used backbone was pretrained on the object-focused task. The results are presented in cyan in Fig. 3 ("w/o CGL fine-tuned"). The performances drop on both datasets, especially on the more complex Image Privacy. This phenomenon proves that the model needs to explore more elements besides objects, especially for a more complex task.

Effectiveness of Dynamic Correlation Modeling To obtain a variant without the ability to dynamically model the correlation among crucial regions, we adopted a graph that all nodes in the graph are connected with each other (i.e. a graph with a fixed all-ones correlation matrix). The performances are presented in green in Fig. 3 ("with fixed correlation"), which are worse than those of DRAG in general. We further proposed a variant that completely disregards the correlation, which is implemented by removing GCN from DRAG and directly adopts the region features $\mathbf{F_w}$ for classification. Similar to Eq. 8, we concatenate the features with a global feature, and feed them into a fully connected layer for final prediction: $\hat{y} = FC(\mathbf{F_c} \oplus \mathbf{F_w})$. The results are presented in orange in Fig. 3 ("w/o GCN"), and the perfor-
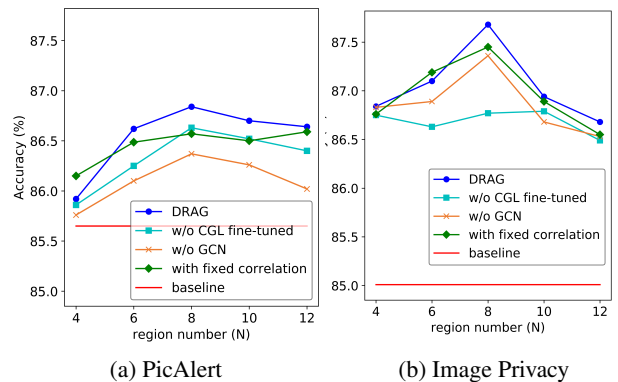

(a) PicAlert   (b) Image Privacy

Figure 3: Ablation study and hyperparameter sensitivity. The baseline refers to ResNet pretrained on the corresponding dataset (presented in red). The performances drop when removing components from the DRAG, proving the effectiveness of these components. The DRAG is relatively robust for region number N, while $N = 8$ achieved slightly better performance than other N on both datasets.

mances further degrade. These results prove that considering the correlation among the crucial elements is essential, while a dynamic correlation is better than a fixed one.

Discussion For DRAG, the performance is better on Image Privacy than on PicAlert, but for the baseline is the opposite. As described before, Image Privacy is an extension of PicAlert with challenging samples, which makes the basic model perform worse. The DRAG benefits from the ability to dynamically find out crucial regions and model their correlation, and still achieve remarkable performance.

The dynamic crucial regions exploring and the dynamic correlation modeling complement each other, but the importance of them is different for the two datasets. The dynamic correlation modeling affects the performances more for PicAlert, while the dynamic crucial regions exploring affects the performances more for Image Privacy. We speculate that for a simpler dataset, the pretrained CGL is good enough to

(a) Region features of a private image
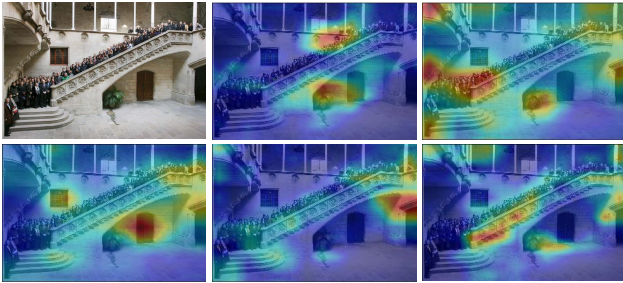
(b) Region features of a public image

Figure 4: Two examples of the learned region feature obtained from CGL. In the private image (a), the CGL focuses on the lamp, the window, the person, the doll on the chair, and the wall. In the public image (b), the CGL focuses on the plant, the people, the doors, and the railing. The results show that the CGL can capture regions of crucial elements to differentiate private and public images.

find out crucial regions, and the model should pay more attention to the interaction among the regions. But for a more complex dataset, the model needs to make more effort to focus on the crucial regions that reflect subtle differences. This may also explain why GIP that relies on the object detector and GNN performed well on PicAlert, but the performances dropped a lot on Image Privacy.

Hyperparameter Sensitivity To validate the robustness of the model under parameter variations, we investigated the sensitivity of the hyperparameter N, which determines the number of regions during channel grouping, and also the number of nodes in the GCN. The results are presented in Fig. 3. We explore the influence of N with the complete model as well as the models in the ablation studies. The tendencies are consistent for all models, and we can get several conclusions. First, for most models, the variances of the performances between different N are not very significant, suggesting that the DRAG is relatively robust. Second, comparing the subtle differences , we found that the region number of 8 is most suitable for our experimental setups on both datasets. During our experiments, we infer that the best selection of N may depend on the size of the feature map $\mathbf{F_b}$ used for clustering. A larger N may be more appropriate for a larger feature map. We will explain this inference based on the following case study.

(a) Public images that are mis-classified into private.

(b) Private images that are mis-classified into public.

Figure 5: Cases of misclassified images.

Case study

Qualitative Analyses of the Region Features We visualized the original image and corresponding features of the crucial regions obtained from CGL to illustrate the capability of CGL. From Fig. 4, we observe that the CGL learned differentiated regions of crucial elements as we expected. We also notice that there exist overlaps between the peak responses in the feature maps, and several feature maps are not very compact. During the training stage, we combined the losses to ensure the classification performance and did not make a quite strict constraint on the $Dis(\cdot)$ and $Div(\cdot)$, and thus the results are reasonable. This may also explain why the region number N of 8 is the most suitable one in our experiments — too small N may neglect important region features, while too large N may result in overlaps.

Limitation of DRAG We conduct this study to learn what kind of images are more likely to be misclassified. Fig. 5 (a) shows public images that were misclassified into *private*. Although group photos are often related to private occasions like family gatherings, the images here are actually art photography and photos of public events. In Fig. 5 (b), the misclassified private images contain elements such as ticket, medicine, age, and credit card number. The examples indicate that the model may fail to understand the given images when the external social context of sharing motivation and textual privacy is necessary. Therefore, we argue that future works may obtain a deeper understanding of the images by introducing social context. For example, techniques of text recognition and natural language understanding can be used to understand the specific types of card-like elements.

Conclusion

In this paper, we proposed the DRAG for privacy-leaking image detection. The DRAG dynamically finds out crucial regions and models their correlation adaptively for each input image without the limitation of pretrained object detectors. The experimental results show that the DRAG that only utilizes visual features outperformed existing methods, including visual-based and multi-modal ones. Further works may consider introducing external social context to obtain a deeper understanding of the images. The code will be released to facilitate further research.

Acknowledgements

References

Buschek, D.; Bader, M.; von Zezschwitz, E.; and De Luca, A. 2015. Automatic privacy classification of personal photos. In *IFIP Conference on Human-Computer Interaction*, 428–435.

Equifax. 2020. Protect against identity theft when sharing photos online. https://www.equifax.co.uk/resources/identity-protection/protect-against-identity-theft-when-sharing-photos-online.html/. Accessed: November, 2020.

Fan, L. 2018. Image pixelization with differential privacy. In *IFIP Annual Conference on Data and Applications Security and Privacy*, 148–162.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Fu, J.; Zheng, H.; and Mei, T. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4438–4446.

Hasan, R.; Shaffer, P.; Crandall, D.; Apu Kapadia, E. T.; et al. 2017. Cartooning for enhanced privacy in lifelogging and streaming videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 29–38.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hoyle, R.; Stark, L.; Ismail, Q.; Crandall, D.; Kapadia, A.; and Anthony, D. 2020. Privacy Norms and Preferences for Photos Posted Online. *ACM Transactions on Computer-Human Interaction*, 27(4): 1–27.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Li, J.; Wong, Y.; Zhao, Q.; and Kankanhalli, M. S. 2017a. Dual-glance model for deciphering social relationships. In *Proceedings of the IEEE International Conference on Computer Vision*, 2650–2659.

Li, Y.; Vishwamitra, N.; Knijnenburg, B. P.; Hu, H.; and Caine, K. 2017b. Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1343–1351.

Liu, C.; Zhu, T.; Zhang, J.; and Zhou, W. 2020. Privacy Intelligence: A Survey on Image Sharing on Online Social Networks. *arXiv preprint arXiv:2008.12199*.

Liu, Y.; Gummadi, K. P.; Krishnamurthy, B.; and Mislove, A. 2011. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, 61–70.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 281–297.

Marcel, S.; and Rodriguez, Y. 2010. Torchvision the machine-vision package of torch. In Bimbo, A. D.; Chang, S.; and Smeulders, A. W. M., eds., *Proceedings of the 18th International Conference on Multimedia*, 1485–1488.

Marino, K.; Salakhutdinov, R.; and Gupta, A. 2017. The More You Know: Using Knowledge Graphs for Image Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2673–2681.

Oh, S. J.; Benenson, R.; Fritz, M.; and Schiele, B. 2016. Faceless person recognition: Privacy implications in social media. In *European Conference on Computer Vision*, 19–35.

Oh, S. J.; Fritz, M.; and Schiele, B. 2017. Adversarial image perturbation for privacy protection a game theory perspective. In *Proceedings of the IEEE International Conference on Computer Vision*, 1491–1500.

Orekondy, T.; Schiele, B.; and Fritz, M. 2017. Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In *Proceedings of the IEEE International Conference on Computer Vision*, 3686–3695.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 8024–8035.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 91–99.

Shen, L.; Hong, R.; Zhang, H.; Zhang, H.; and Wang, M. 2019. Single-shot Semantic Image Inpainting with Densely Connected Generative Networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1861–1869.

Shetty, R. R.; Fritz, M.; and Schiele, B. 2018. Adversarial scene editing: Automatic object removal from weak supervision. In *Advances in Neural Information Processing Systems*, 7706–7716.

Simon, M.; and Rodner, E. 2015. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 1143–1151.

Solsman, J. E. 2020. Deepfake bot on Telegram is violating women by forging nudes from regular pics. https://www.cnet.com/news/deepfake-bot-on-telegram-is-violating-women-by-forging-nudes-from-regular-pics/. Accessed: October, 2020.

Sun, X.; Wu, P.; and Hoi, S. C. 2018. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing*, 299: 42–50.

Teney, D.; Liu, L.; and van den Hengel, A. 2017. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Tonge, A.; and Caragea, C. 2019. Dynamic deep multi-modal fusion for image privacy prediction. In *The World Wide Web Conference*, 1829–1840.

Tonge, A.; Caragea, C.; and Squicciarini, A. 2018. Uncovering scene context for predicting privacy of online shared images. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 8167–8168.

Tran, L.; Kong, D.; Jin, H.; and Liu, J. 2016. Privacy-CNH: a framework to detect photo privacy with convolutional neural network using hierarchical Features. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 1317–1323.

Tuunainen, V. K.; Pitkänen, O.; and Hovi, M. 2009. Users' awareness of privacy on online social networking sites-case Facebook. *Bled 2009 Proceedings*, 42.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, Y.; Norcie, G.; Komanduri, S.; Acquisti, A.; Leon, P. G.; and Cranor, L. F. 2011. " I regretted the minute I pressed share" a qualitative study of regrets on Facebook. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, 1–16.

Wang, Z.; Chen, T.; Ren, J.; Yu, W.; Cheng, H.; and Lin, L. 2018. Deep reasoning with knowledge graph for social relationship understanding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 1021–1028.

Wu, J.; Yang, Y.; Liu, H.; Liao, S.; Lei, Z.; and Li, S. Z. 2019. Unsupervised graph association for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8321–8330.

Yang, G.; Cao, J.; Chen, Z.; Guo, J.; and Li, J. 2020. Graph-Based Neural Networks for Explainable Image Privacy Inference. *Pattern Recognition*, 107360.

Ye, J.; He, J.; Peng, X.; Wu, W.; and Qiao, Y. 2020. Attention-Driven Dynamic Graph Convolutional Network for Multi-Label Image Recognition. In *European Conference on Computer Vision*, 649–665.

Zerr, S.; Siersdorfer, S.; and Hare, J. 2012. Picalert!: a system for privacy-aware image classification and retrieval. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2710–2712.

Zerr, S.; Siersdorfer, S.; Hare, J.; and Demidova, E. 2012. Privacy-aware image classification and search. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 35–44.

Zheng, H.; Fu, J.; Mei, T.; and Luo, J. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 5209–5217.

Zhou, W.; Li, H.; Lu, Y.; and Tian, Q. 2012. Principal visual word discovery for automatic license plate detection. *IEEE Transactions on Image Processing*, 21(9): 4269–4279.