2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Privacy Attributes-aware Message Passing Neural Network for Visual Privacy Attributes Classification

Hanbin Hong*, Wentao Bao*, Yuan Hong† and Yu Kong*
*Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester, NY 14623
†Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616
E-mail: hh9665@rit.edu, wb6219@rit.edu, yuan.hong@iit.edu, yu.kong@rit.edu

***Abstract*—Visual Privacy Attribute Classification (VPAC) identifies privacy information leakage via social media images. These images containing privacy attributes such as *skin color*, *face* or *gender* are classified into multiple privacy attribute categories in VPAC. With limited works in this task, current methods often extract features from images and simply classify the extracted feature into multiple privacy attribute classes. The dependencies between privacy attributes, e.g., *skin color* and *face* typically co-exist in the same image, are usually ignored in classification, which causes performance degradation in VPAC. In this paper, we propose a novel end-to-end Privacy Attributes-aware Message Passing Neural Network (PA-MPNN) to address VPAC. Privacy attributes are considered as nodes on a graph and an MPNN is introduced to model the privacy attribute dependencies. To generate representative features for privacy attribute nodes, a class-wise encoder-decoder is proposed to learn a latent space for each attribute. An attention mechanism with multiple correlation matrices is also introduced in MPNN to learn the privacy attributes graph automatically. Experimental results on the Privacy Attribute Dataset demonstrate that our framework achieves better performance than state-of-the-art methods for visual privacy attributes classification.**

I. Introduction

Concerns about leaking privacy information are increasingly attracting people's attention [1], [2]. The privacy information hidden in images is of higher risk since more and more people tend to post personal images on their social media sites. To prevent potential visual privacy information leakage, it is of great importance to predict the privacy attributes in images [3], [4]. The most straightforward way is to classify images into pre-defined privacy attributes [5]. Our goal in this paper is to develop a more effective architecture in visual privacy attributes classification.

Social media images contain various visual privacy attributes. One image often contains multiple privacy attributes. In this case, the visual privacy attributes classification can be considered as a multi-label classification task (MLC). As Figure 1 shows, the image is tagged with multiple privacy attributes. The classification result is regarded as correct only when all the attributes are classified correctly, which brings exponential difficulty [6]. For example, given binary privacy attribute labels, for multi-label classification tasks with n privacy attribute classes, the output space size is 2^n. The exponentially increasing output space is the main challenge in VPAC.

To reduce the output space, the most intuitive way is to divide it into smaller output spaces. Many methods target at reducing output space in multi-label classification by combining or clustering classes [7], [8], [9]. These methods achieve a better performance comparing to methods [10] that treat labels as isolated labels. However, methods simply separating the output space encounter difficulties when the dependencies between labels become complex. For example, in Figure 1, the image with *Face* privacy attribute can also contain *Occupation* privacy attribute, while in other images, *Face* and *Occupation* privacy attributes may not co-occur. In this case, simply separating attributes into smaller groups may cause performance degradation.

Fig. 1. An example in Privacy Attribute Dataset [5]. The task of visual privacy attribute classification is to classify images into multiple privacy attributes. In this example, 12 visual privacy attributes are tagged to this image. Some of them are highly dependent. For example, skin color and face are highly related because the skin color can be recognized from the face area. Thus, the challenge in VPAC is to model the privacy attributes dependencies correctly.

Some work follows the idea of connecting labels with sequential order [11], [12], [13]. However, considering multi-label classification as sequential prediction brings new problems. On one hand, the sequential prediction task is generally fulfilled with recurrent models which lead to the accumulation of false prediction, especially when there are many positive labels. Once the upstream label prediction is inaccurate, the downstream label prediction will be heavily affected. On the other hand, the structural dependencies between labels are not fully addressed. One privacy attribute can structurally rely on multiple privacy attributes rather than merely relying on previously predicted attributes.

Benefited from the great success in Graph Neural Network [14], [15], [16], recent work uses graph-based models to learn the structural dependencies between labels and achieve a better

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412853

performance [17], [18], [19], [20]. However, there are two main challenges that need to be addressed for graph-based models for the VPAC task. The first challenge is to define the graph of privacy attributes. The graph topology of privacy attributes is not explicitly given by the VPAC task so that how to explicitly learn the privacy attribute graph is of great importance. The second challenge is to generate representative features for privacy attribute nodes. The features extracted by traditional graph neural networks summarize the relationship between nodes, leading to a mixture of features from different privacy attributes. However, the class-wise feature for each privacy attribute node should be more discriminative for the VPAC task.

To model the privacy attribute dependencies and address the two main challenges in graph-based models, in this paper, we propose a novel Privacy Attributes-aware Message Passing Neural Network (PA-MPNN) framework. Our model learns a latent space for each class and using neural message passing to model the structural dependencies between labels. Specifically, we propose an attention mechanism with multiple correlation matrices (MCM) to adjust the graph structure automatically and a class-wise encoder-decoder (CED) to generate representative node features for label nodes. Experimental results on the Privacy Attribute Dataset [5] demonstrate that our method outperforms the existing methods.

The main contributions of this paper are as follows:

- We propose an end-to-end trainable framework for the multi-label VPAC task, which achieves better performance compared to existing methods.
- We propose a class-wise encoder-decoder which learns a latent space for each class to generate class-relevant features for label nodes.
- We propose an attention mechanism with multiple correlation matrices to explicitly learn the graph structure, which can handle the challenge of the attribute dependency problem.

II. Related Work

Visual privacy leakage is increasingly attracting attention especially when social media becomes popular. Detecting visual privacy attributes has great significance to protect the visual privacy of users. Some works focus on detecting specific visual privacy attributes such as license plates [3], ages [21], faces [22], landmarks [23] and occupations [24]. [5] is the first work proposing multiple privacy attributes classification tasks. They also collect a visual privacy attribute dataset and learn a privacy risk score to alarm potential privacy leakage in images.

With multiple privacy attributes as labels, the VPAC task faces challenges of large prediction space. Some works focus on reducing the output space dimension to better perform MLC. [7] combines a set of labels to a single label using the pruned sets method and achieves better and faster performance. [8] address hierarchical multi-label classification in protein function prediction using probabilistic clustering to construct class hierarchy. [25] projects high-dimension label vectors to low-dimension label vectors by embedding and constraints the distance between the nearest embedded labels. The performance of these methods is limited because they fail to model the complex dependencies between labels.

In order to model more complex labels dependencies, several works consider multi-label classification as a sequential prediction to model the sequential dependency. [11] considers multi-label classification as a set of binary classification and uses classifier chains to model the label correlation. [12] follows the same setting in [11] and uses recurrent neural networks instead of classifier chains to exploit the information from previous decisions. [13] combines RNN with CNN to jointly embed labels and images to learn the semantic label dependency. Although these methods model sequential label dependencies, they suffer from the error accumulation and fail to model the topology structure dependency.

Recent works consider labels as nodes on a graph to perform label interaction and achieve boosting performance. [18] and [19] both use Graph Convolution Network (GCN) to map label graph to object classifiers. [20] and [17] both use Message Passing Neural Networks [26] to model the label dependencies. When these graph-based methods are used in VPAC, two challenges need to be addressed: how to learn a privacy attribute label graph and how to generate representative node features. To learn the label graph, [20] and [17] both use attention mechanism to learn attention weights between labels. [18] and [19] both use re-weighted correlation matrix to define the label graph. However, the attribute dependency is heavily dependent on the defined graph determined by the correlation matrix, which may not accurately represent the natural relationship between labels. To address the second challenge, [18] uses attention mechanism to learn class-relevant attention map on image feature maps. However, since the attention map is also learned from multiple different privacy attributes, there is no guarantee that these attention maps can represent the corresponding label.

To this end, we propose an end-to-end PA-MPNN model to jointly generate class-relevant features for nodes, learn graph structure and model label dependencies.

III. Method

As Figure 2 shows, our model contains a ResNet-50 [27] as Feature Extractor, a Class-wise Encoder-Decoder(CED) and an MPNN. The ResNet-50 extracts features from input images. The CED is introduced to to learn latent space for each class, which computes the node features to highly represent the corresponding privacy attribute. The MPNN is employed to model the dependencies between privacy attributes. Besides, we propose an attention mechanism to compute the attention weights between privacy attributes from four correlation matrices which represent the pair-wise relationship between privacy attributes.

Given the input image $\mathbf{I}$ and the label $\mathbf{y} = [y_1, y_2, ..., y_n]$, where n is the number of privacy attribute class and $y_v = 1$ if the image is annotated with attribute v, otherwise $y_v = 0$. The visual privacy attributes classification task is to learn a

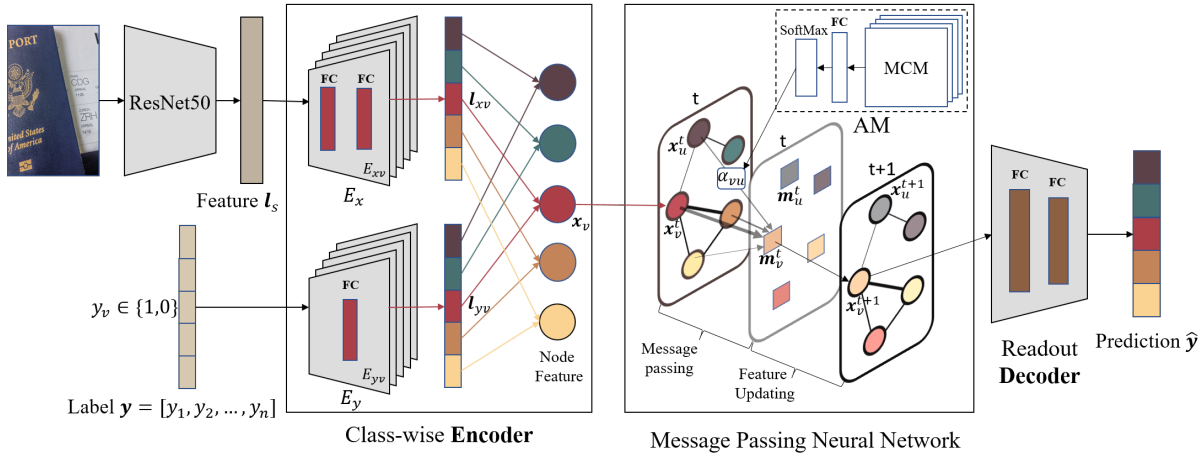

Fig. 2. Overall Framework of our PA-MPNN model for visual privacy attribute classification. We use the Message Passing Neural Network to model the dependencies between privacy attributes. An Attention Mechanism (AM) along with Multiple Correlation Matrices (MCM) is also proposed to learn attention weights for neighboring nodes. Before the MPNN, a ResNet-50 is used as our feature extraction network and a Class-wise Encoder-Decoder (CED) is proposed to learn latent space for node features which can represent privacy attributes. In the CED, the encoder $E_y(\cdot)$ is used to guide the learning of encoder $E_x(\cdot)$. After MPNN, the Readout Decoder will predict the privacy attributes according to the node features.

classifier to predict the probabilities of all the labels $\hat{\mathbf{y}} = f_c(\mathbf{I})$, $\hat{\mathbf{y}} \in \mathcal{R}^n$, which is also a multi-label classification task. Privacy attributes are considered as nodes on undirected label-interact graph G with node hidden features $\mathbf{x}_v$ and edge features $\mathbf{e}_{vu}$.

A. Class-wise Encoder-Decoder

Since we consider privacy attributes as nodes on graphs and learn the label graph with the guidance of correlation matrices, our node feature should be able to represent corresponding privacy attributes. In this case, the label $\mathbf{y}$ is encoded to guide the node feature generating. We achieve this goal by minimizing the l_2 distance between node features from encoder $E_y(\cdot)$ and $E_x(\cdot)$. The encoder $E_y(\cdot)$ is only used in training stage, so in the testing stage, we only have encoder E_x.

As the Figure 2 shows, the CED contains two encoders $E_x(\cdot)$ and $E_y(\cdot)$ and one decoder which is also the readout function of MPNN. The encoder $E_y(\cdot)$ encodes the label to guide the learning of encoder $E_x(\cdot)$. The encoder $E_y(\cdot)$ is a set of class-wise encoders $\{E_{yv}|v = 1, 2, ..., n\}$ with each encoder E_{yv} corresponding to class v. The encoder E_x encodes the feature from feature extractor to represent privacy attributes for each node. The encoder $E_x(\cdot)$ is also a set of encoders $\{E_{xv}|v = 1, 2, ..., n\}$.

To be more specific, given the extracted features $\mathbf{l}_s \in \mathcal{R}^s$ from last pooling layer of ResNet-50, for each class v, we use the encoder $E_{xv}(\cdot)$ to compute latent features $\mathbf{l}_{xv} = E_{xv}(\mathbf{l}_s)$, $\mathbf{l}_{xv} \in \mathcal{R}^d$. To enable node features to represent their corresponding classes, we use the encoder $E_{yv}(\cdot)$ to compute latent features $\mathbf{l}_{yv} = E_{yv}(\sigma(y_v))$, $\mathbf{l}_{yv} \in \mathcal{R}^d$, where $\sigma(y_v) = 1$ when $y_v = 1$ and $\sigma(y_v) = -1$ when $y_v = 0$. We use the Mean Square Error loss to minimize the distance between $\mathbf{l}_{xv}$ and $\mathbf{l}_{yv}$. In this way, we learn the same latent space in which the encoded features are able to represent privacy attributes. Then we consider $\mathbf{l}_{xv}$ as node feature $\mathbf{x}_v$ of graph G

B. Message Passing Neural Network

Message Passing Neural Network (MPNN) is first introduced in [26] as a generalization framework of Graph Neural Networks (GNN) [28]. The primary goal of MPNN is to predict the graph's general properties by synthesizing nodes' features. For each node on graph, the MPNN will update its node features using the features from neighboring nodes. In this case, the MPNN learns to build the dependencies between privacy attributes. The MPNN will update the node features in each layer t, and each layer contains two trainable functions. With trainable message passing function $M(\cdot)$, every node v learns to receive message $\mathbf{m}_v$ from its neighboring node u using their node features $\mathbf{x}_v$, $\mathbf{x}_u$ and edge features $\mathbf{e}_{vu}$.

$$\mathbf{m}_v^t = \sum_{u \in \mathcal{N}(v)} M_t(\mathbf{x}_v^t, \mathbf{x}_u^t, \mathbf{e}_{vu}) \tag{1}$$

Here, x_v^t denotes node features of node v in layer t, v and u denote different node indexes ($v \neq u$), e_{vu} denotes the edge vector between node v and node u, and $\mathbf{m}_v^t$ denotes messages passing from neighbor node set $\mathcal{N}(v)$ to node v.

By trainable node update function $U(\cdot)$, node v learns to update its node features $\mathbf{x}_v$ using received messages from neighboring nodes and its node features in last step.

$$\mathbf{x}_v^{t+1} = U_t(\mathbf{x}_v^t, \mathbf{m}_v^t) \tag{2}$$

As Figure 2 shows, the message passing and feature updating process happens in layer t for each node. After T layers, each node contains information of neighboring nodes within distance T on the graph. Then a trainable readout function $R(\cdot)$, which is also the decoder of the class-wise encoder-decoder architecture, is used to read the synthesized features in each node.

$$\hat{\mathbf{y}} = R(\{\mathbf{x}_v^T | v \in G\}) \tag{3}$$

Fig. 3. Examples of prediction results. Target privacy attributes, predicted privacy attributes of ResNet-50 and prediction of PA-MPNN are shown in red, blue and black boxes, respectively. The extra detected privacy attributes using PA-MPNN are marked in bold.

The message passing process and feature updating process are learnt by message passing function $M_t(\cdot)$ and node update function $U_t(\cdot)$. The prediction is made by the readout function $R(\cdot)$. Thus, the optimal interacting mechanisms between nodes are obtained by optimizing these three functions $M(\cdot)$, $U(\cdot)$, and $R(\cdot)$. Notably, there is no limitation on the forms of these functions, researchers can use different forms of these functions to fit their problems (e.g., multi-layer perception (MLP) message passing function [29], recurrent neural network update function or gated recurrent unit [30] update function).

Inspired by [20], we use the MPNN to model the dependencies between classes. Instead of defining the label graph using a single correlation matrix, we propose an attention mechanism to learn attention weights from multiple correlation matrices. In the message passing function, the attention weight is learned by embedding edge feature vector $\mathbf{e}_{vu}$. The edge feature vector is computed from multiple correlation matrices, which will be introduced in subsection C. Then we use a soft-max function to normalize the attention weight $\alpha_{vu} = softmax(W_e\mathbf{e_{vu}})$. Our message passing function is defined as equation (4). Since the message from node u to node v is not only dependent on one of them but dependent on both, we concatenate features of these two nodes and use an MLP to learn the message. The attention weight is also included because nodes with high dependencies contribute more.

$$M_t(\mathbf{x}_v^t, \mathbf{x}_u^t, \mathbf{e}_{vu}^t) = \alpha_{vu}^t \mathrm{MLP}_M(concat(\mathbf{x}_v^t, \mathbf{x}_u^t)) \quad (4)$$

For node update function, we concatenate node features and messages received from other nodes and use an MLP to learn the update mechanism. In this case, we enable our node update function to balance the information in received messages and existing features.

$$U_t(\mathbf{x}_v^t, \mathbf{m}_v^t) = \mathrm{MLP}_U(concat(\mathbf{x}_v^t, \mathbf{m}_v^t)) \quad (5)$$

Where $\mathbf{m}_v^t$ is computed using equation (1). After T layers, we adopt an MLP to read the information in node features.

$$R(\mathbf{x}_v^T) = \mathrm{MLP}_R(\mathbf{x}_v^T) \quad (6)$$

We use the Binary Cross Entropy loss as our prediction loss $\mathcal{L}_{BCE}$. Notably, we compute the prediction loss after each layer. To learn the latent space, we use the MSE loss $\mathcal{L}_{MSE}$. Thus, the total loss of our model is the sum of $\mathcal{L}_{MSE}$ and $\mathcal{L}_{BCE}$.

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + \mathcal{L}_{MSE} \quad (7)$$

C. Multiple Correlation Matrix

The dependencies between label nodes are guided by graphs. The graph is determined by edges between pairs-wise nodes. Edges are often pre-defined by the correlation matrix [19], which is a matrix with conditional co-occurrence probabilities between nodes as elements. For example, the matrix element can be the conditional co-occurrence probability of attribute u given attribute v, namely $p_{vu} = P(y_u|y_v)$. However, this single correlation matrix is insufficient to represent the relationship between privacy attribute nodes. We need to consider other possible conditional probabilities such as $P(y_v|y_u)$, $P(y_u|\neg y_v)$, and $P(y_v|\neg y_u)$, where $P(A|\neg B)$ denotes conditional probability of A without B. We use these four correlation matrices to represent the edge feature between nodes and use edge embedding MLP_e to learn the attention weights on node features.

IV. Experiments

We conduct privacy attributes classification experiments on Privacy Attributes Dataset [5]. The Privacy Attributes Dataset contains $22,167$ images and 68 visual privacy attributes. Each image is tagged with 68 binary labels describing which visual privacy attribute class it belongs to. The evaluation metrics and implementation details are first introduced. Then we report the results comparing with the methods described in [5]. We only compare our results with [5] since [5] is the only work on Privacy Attributes Dataset based on our best knowledge.

TABLE I
Comparison under mAP metrics on Privacy Attribute Dataset. The mAP metrics of CaffeNet, GoogleNet, and ResNet-50 are provided in [5]. Our method denotes our PA-MPNN model with the class-wise encoder-decoder, the attention mechanism, and the multiple correlation matrices.

Methods	CaffeNet [5]	GoogleNet [5]	ResNet-50 [5]	ours
mAP	42.99	43.29	47.45	**49.93**

A. Implementation Details

Our framework is consists of ResNet feature extractor, class-wise encoder-decoder and message passing neural network. We use the output from the last average pooling layer of ResNet-50 as the input of CED E_x, which has a dimension of $2,048$. The ResNet-50 is pre-trained on the Privacy Attributes Dataset. In each Class-wise Encoder $E_{xv}(\cdot)$, we use two MLP layers to compute the 68 nodes' features with a dimension of 256. As for MPNN, all the MLP have two layers. We also add instance normalization layer to normalize the latent features $\mathbf{l}_x$ and $\mathbf{l}_y$ in each layer.

Following the settings in [5], the dataset is divided into a training set with $10,000$ samples, a validation set with $4,167$ samples, and a test set with $8,000$ samples. We use Adam as the optimizer with an initial learning rate of 5×10^{-5}. The learning rate decays by a factor of 0.9 for every 3 epochs. We adopt an early stopping strategy to stop the training. The whole framework is implemented based on PyTorch.

B. Evaluation Metrics

Results in [5] are derived only under the metric mean average precision (mAP). Besides, we show the average precision (AP) over each class. To conduct a more comprehensive comparison, we also report the Micro average F1 score (miF1) and the Macro average F1 score (maF1) in Section V.

C. Experimental Results

The results on Privacy Attribute Dataset are shown in Table I. From Table I, we can see that our PA-MPNN model achieves a better performance on mAP with 2.48% gain comparing to methods in [5]. The CaffeNet, GoogleNet and ResNet-50 methods in [5] only contain CNNs to extract image features and fully connected layers to predict the privacy attributes, which regards labels as isolated labels. Instead of keeping the labels isolated, our method models the label dependencies which reduces the output space dimension of MLC tasks and improves our performance on mAP.

Figure 3 shows three examples of prediction results. In the left example, our method predicts *Email Address* and *Email Content* together since they are often co-occur. In the center example, our method is able to predict *Eye Color* and *Height* since the *Eye Color* attribute can be recognized when the complete face is recognized and human's height and weight are often exposed together. In the right example, results show that our method is able to capture the dependencies between *Passport* attribute and *Nationality* attribute. Comparing to ResNet-50, our method is able to model the privacy attribute dependencies, which may reflect the common sense of human.

TABLE II
Comparison of our methods under other matrics on the Privacy Attribute Dataset. 'CED' denotes our class-wise encoder-decoder. 'Att.' denotes our attention mechanism in MPNN. 'MCM' denotes multiple correlation matrices, and no tick means we use a single correlation matrix.

CED	Att.	MCM	mAP	miF1	maF1
	√	√	49.83	0.7725	0.4428
√		√	49.78	0.7645	0.4384
√	√		49.78	0.7683	0.4284
√	√	√	**49.93**	**0.7751**	**0.4456**

Here we also show the AP over each class in Figure 4. It is obvious that our method achieves high AP on highly related attributes like *race*, *color*, *gender*, *eye color*, *face* and *hair color*. For attributes with text content like *username*, *email*, *home address*, *first name* and *last name*, the APs are low which may be due to our framework is unable to distinguish texts. For attributes with tiny cues like *tattoo* and *marital status* which are often shown by marital rings, the APs are also low because our framework is unable to capture the tiny cues in image.

V. Ablation Studies

In this section, we evaluate the effectiveness of our class-wise encoder-decoder, attention mechanism, and multiple correlation matrices with both quantitative and qualitative analysis. We compare the results of our models on mAP, miF1 and maF1 matrics. We also visualize the node features to evaluate our class-wise encoder-decoder qualitatively.

A. Quantitative Comparison

In this section, we compare the performance of our methods on mAP, miF1, and maF1. As Table 2 shows, our model performs better on all the three matrices when the CED, attention mechanism and multiple correlation matrices are adopted. When CED is removed, our method performs worse because CED enables our PA-MPNN model to generate representative node features which are corresponding to its class. In this case, the MPNN can better model the label dependencies with each node's features corresponding to the class. When the attention mechanism is removed, our model is unable to learn a label graph so the label graph becomes a fully connected graph. The messages passing from all other nodes are equally important. This will harm the label dependencies modeling and cause degradation to all the matrices. When using a single correlation matrix instead of multiple correlation matrices to define the edges between nodes, our model is unable to learn a more accurate label graph, the performance on all the three metrics goes down. Thus, we can infer that the other three correlation matrices contain vital information to guide the graph structure learning.

B. Qualitative Comparison

We also use the t-SNE [31] method to visualize the node features. As the Figure 5 shows, our CED is able to gener-

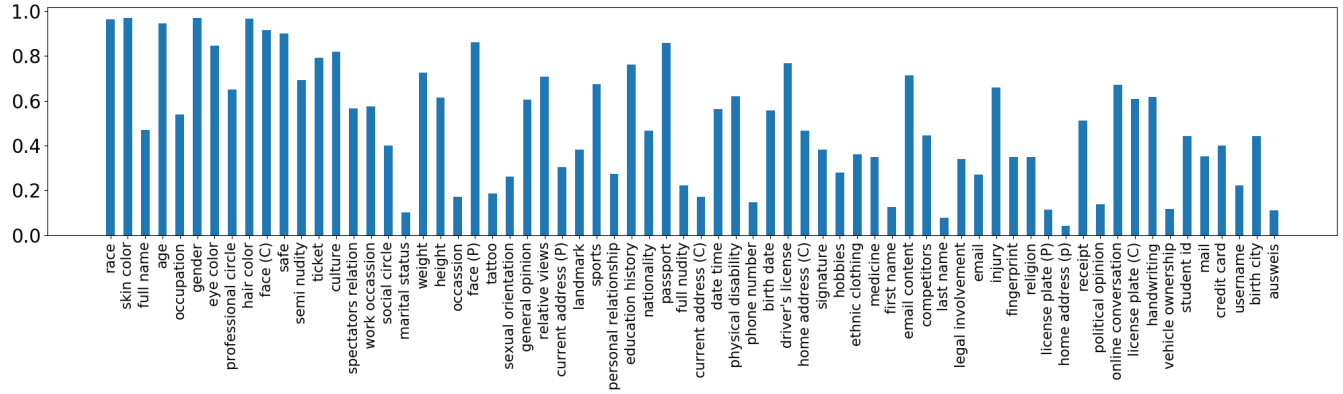

Fig. 4. Average Precision scores on all visual privacy attributes. (C) represents 'complete'. (P) represents 'partial'.

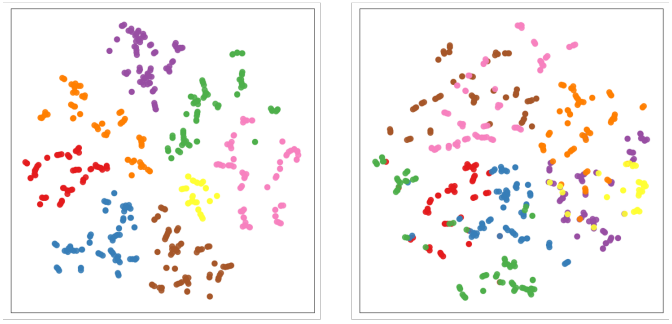

Fig. 5. The visualization of node features using the t-SNE method. The left subfigure visualizes the node features with a class-wise encoder-decoder. The right subfigure visualizes the node features without class-wise encoder-decoder. Each color represents one class and 8 classes are selected. Better class separation is observed on the left subfigure.

ate representative and disentangled node features. The node features from fully connected layers are overlapped, which indicates that the node features for one class can be recognized as other classes. The overlap node features may be because that the classifier can not distinguish highly related privacy attributes on one image. For example, the *eye color* and the *face* attributes often occur together.

VI. Conclusion

Visual Privacy Attribute Classification is of great importance in preventing privacy leakage, which is a multi-label classification problem. To model the dependencies between privacy attribute labels, we propose a novel end-to-end Privacy Attribute-aware Message Passing Neural Network (PA-MPNN) framework. To generate class-relevant features for label nodes, we propose a class-wise encoder-decoder to learn a latent space for node features. An attention mechanism and multiple correlation matrices are also proposed to improve the MPNN. Experiments on Privacy Attribute Dataset show that our PA-MPNN model outperforms the existing methods on visual privacy attribute classification. Further quantitative and qualitative experiments validate our proposed CED, attention mechanism and multiple correlation matrices on VPAC.

References

[1] S. Kokolakis, "Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon," Computers & security, vol. 64, pp. 122–134, 2017.

[2] H. Li, H. Zhu, S. Du, X. Liang, and X. S. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 4, pp. 646–660, 2016.

[3] J. Gao, L. Sun, and M. Cai, "Quantifying privacy vulnerability of individual mobility traces: a case study of license plate recognition data," Transportation research part C: emerging technologies, vol. 104, pp. 78–94, 2019.

[4] F. Peng, G.-l. Ping, and S.-k. Ge, "A face privacy protection scheme using cnn based roi editing," in 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019, pp. 345–352.

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

[6] F. Luo, W. Guo, Y. Yu, and G. Chen, "A multi-label classification algorithm based on kernel extreme learning machine," Neurocomputing, vol. 260, pp. 313–320, 2017.

[7] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in 2008 eighth IEEE international conference on data mining. IEEE, 2008, pp. 995–1000.

[8] R. C. Barros, R. Cerri, A. A. Freitas, and A. C. de Carvalho, "Probabilistic clustering for hierarchical multi-label classification of protein functions," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2013, pp. 385–400.

[9] G. Nasierding, G. Tsoumakas, and A. Z. Kouzani, "Clustering based multi-label classification for image annotation and retrieval," in 2009 IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2009, pp. 4514–4519.

[10] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," Pattern recognition, vol. 37, no. 9, pp. 1757–1771, 2004.

[11] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," Machine learning, vol. 85, no. 3, p. 333, 2011.

[12] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in Advances in neural information processing systems, 2017, pp. 5413–5423.

[13] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2285–2294.

[14] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," arXiv preprint arXiv:1711.04043, 2017.
[15] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," arXiv preprint arXiv:1812.08434, 2018.
[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017.
[17] K. Do, T. Tran, T. Nguyen, and S. Venkatesh, "Attentional multilabel learning over graphs: a message passing approach," Machine Learning, vol. 108, no. 10, pp. 1757–1781, 2019.
[18] Q. Meng and W. Zhang, "Multi-label image classification with attention mechanism and graph convolutional networks," in Proceedings of the ACM Multimedia Asia on ZZZ, 2019, pp. 1–6.
[19] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5177–5186.
[20] J. Lanchantin, A. Sekhon, and Y. Qi, "Neural message passing for multi-label classification," arXiv preprint arXiv:1904.08049, 2019.
[21] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-cnn for age estimation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5183–5192.
[22] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1143–1155, 2017.
[23] A. Boiarov and E. Tyantov, "Large scale landmark recognition via deep metric learning," in Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 169–178.
[24] M. Shao, L. Li, and Y. Fu, "What do you do? occupation recognition in a photo via social context," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3631–3638.
[25] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in Advances in neural information processing systems, 2015, pp. 730–738.
[26] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017, pp. 1263–1272.
[27] 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, pp. 770–778.
[28] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2008.
[29] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," Journal of computer-aided molecular design, vol. 30, no. 8, pp. 595–608, 2016.
[30] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.
[31] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.