Privacy-preserving Federated Learning for Industrial Defect Detection Systems via Differential Privacy and Image Obfuscation

Chia-Yu Lin*, Yu-Chen Yeh*, and Makena Lu†

* *Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan*

† *Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan*

Corresponding Author: Chia-Yu Lin (sallylin0121@ncu.edu.tw)

2024 IEEE Conference on Artificial Intelligence (CAI) | 979-8-3503-5409-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CAI59869.2024.00203

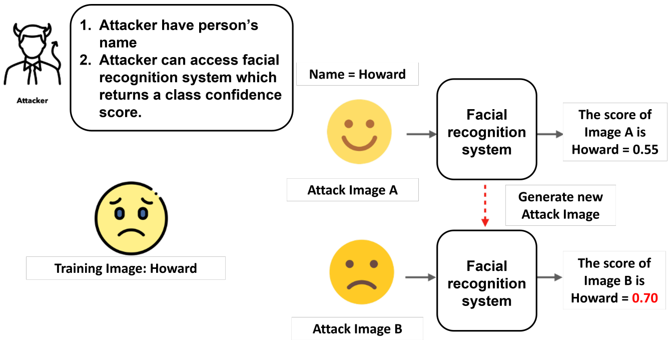
***Abstract*—Artificial Intelligence (AI) has been widely used in manufacturing to detect defects. AI models utilize product images to distinguish whether a product is normal or abnormal. If attackers use the model inversion attack to attack AI models, the input images can be roughly restored, resulting in product information leakage. In this paper, we propose a Privacy-preserving Industrial Defect Detection System (PIDS), which includes three image obfuscation methods to hide input image information and uses them to train the model. Federated learning and differential privacy are also applied to ensure that sensitive data remains decentralized and secure, even during training. Federated learning allows the model to be trained across multiple local datasets without centralized data collection, thereby reducing the risk of data exposure. Differential privacy adds another layer of protection by adding randomness to the learning process, making it hard for attackers to extract sensitive information from the trained model. Experiments show that the proposed system can achieve a high accuracy level of 96.5% in defect image classification. Therefore, the proposed system can detect defects accurately and preserve product information in terms of data and models.**

***Index Terms*—Industrial defect detection, federated learning, differential privacy, image obfuscation**

I. Introduction

Adopting artificial intelligence (AI) to detect defects is a trend in manufacturing. Products are captured as images for training AI models. These training images are closely related to the products of the production lines and may contain important business secrets. If an AI model identified on the production line suffers from the model inversion attack [1], attackers can roughly restore the input data, leading to compromised product information and causing considerable damage to the enterprise, as shown in Fig. 1.

Some researchers proposed image obfuscation techniques to protect data. In [2], the effectiveness of two obfuscation techniques, “blurring” and “blocking”, was investigated. Blurring, a conventional method, is often employed to obscure details like faces in images or license plates in street view photos. In contrast, blocking achieves obfuscation by overlaying a solid block over a specific object in an image. However, blurring images may compromise their quality, and the blurring effect might be reversible, implying certain limitations in privacy protection. On the other hand, blocking images in machine learning may result in less promising training outcomes. [3] proposed “Learnable image encryption” to protect data. This method utilized block-wise image encryption with moderate strength, making the data indecipherable to humans while remaining understandable to machines. Additionally, it leverages the computational process of deep neural networks, facilitating robust calculations to boost its efficiency. However, if the object is sufficiently distinct, the outline of the encrypted object could still be discerned.



Attacker
1. Attacker have person's name
2. Attacker can access facial recognition system which returns a class confidence score.
Name = Howard
Facial recognition system
The score of Image A is Howard = 0.55
Attack Image A
Generate new Attack Image
Training Image: Howard
Facial recognition system
The score of Image B is Howard = 0.70
Attack Image B

Fig. 1: Model inversion attack.

This paper introduces a Privacy-preserving Industrial Defect Detection System (PIDS) to preserve model and data privacy while maintaining high accuracy. We design three image obfuscation methods to hide the images' production information so the human eye cannot distinguish the content of input images. Meanwhile, the AI model can still learn the corresponding features and classify them, thereby preventing the leakage of product information. We also employ a federated learning framework for model training to avoid data leakage. Additionally, we use Differentially Private Stochastic Gradient Descent (DP-SGD) [4] as an optimizer to protect the model.

We conduct three experiments to evaluate PIDS based on a manufacturing dataset and an open dataset. From the experiment based on the manufacturing dataset, PIDS can achieve 96.5% accuracy for image classification and only drops 0.2% accuracy while applying image obfuscation. From the experiment based on the open dataset, PIDS only drops 1% accuracy while applying image obfuscation. The results show that PIDS can ensure accuracy and privacy at the same time. Furthermore, PIDS can adjust the methods based on different

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00203

characteristics of datasets while maintaining accuracy. Finally, we investigate the impact of the noise multiplier of the DP-SGD optimizer, which is the most important parameter of differential privacy (DP). The results show that different models require a trade-off between privacy protection and accuracy. Further research can explore alternative privacy protection methods and optimization strategies to find a better balance, ensuring the simultaneous consideration of data privacy and model accuracy.

In this paper, we achieve the following with PIDS:

- PIDS obfuscates images to a greater degree than traditional methods, such as learnable image encryption [3].
- PIDS achieves a balance between privacy protection and model performance.
- Although attackers can obtain input data through model inversion attacks, they still cannot recognize the product's data.

II. Background

A. Federated Learning

Federated Learning (FL) is a decentralized approach to machine learning that spreads datasets across multiple users' local devices, avoiding centralizing data on a single server. In FL, users preserve data privacy by performing model training on local devices without sharing the original data. Only the updated model parameters are sent to the central server for aggregation to form the global model.

FL has several advantages. First, it is resource-efficient since the model training process is performed on the local device, reducing the need for data transfer and saving bandwidth and computing resources. Second, FL can handle datasets scattered across different locations or organizations, providing greater flexibility and scope of application. This makes FL ideal for machine learning in a decentralized environment. Most importantly, FL has core strengths in privacy protection. Compared with traditional centralized machine learning, FL preserves the privacy of raw data by distributing model training among multiple users. Only the model parameters are transmitted, and details of personal data are protected, ensuring data security.

Overall, FL performs well in terms of privacy preservation, resource efficiency, and handling decentralized datasets. It introduces innovative approaches to address data privacy and machine learning challenges in a decentralized environment, providing safe and efficient solutions for machine learning.

B. Differential Privacy

Differential privacy, as elucidated in studies such as [5]–[7], constitutes a foundational method for preserving individual privacy while analyzing personal data. By incorporating slight perturbations, this approach guarantees that observers cannot discern precise data modifications before analysis, thus mitigating the risk of data leakage. Differential privacy endeavors to optimize query precision while minimizing the potential for privacy breaches.

The following formula is the definition of ε-differential privacy, which is used to measure the privacy protection strength of data processing algorithms:

- $Pr\left[M\left(d\right)\in S\right]$ indicates the probability that the algorithm's output M falls in the set S given the data set d. This probability measures how privacy-preserving the output of algorithm M is for the dataset d.
- $Pr\left[M\left(d'\right)\in S\right]$ indicates the probability that the algorithm's output M falls in the set S given the data set d' (one data sample difference from d). This probability measures how privacy-preserving the output of algorithm M is for the dataset d'.
- ε is a non-negative number and is called the privacy parameter. It quantifies the difference between the output of algorithm M for two similar datasets, d and d'. The smaller the ε, the smaller the impact of the difference between data sets on the algorithm output, and the higher the degree of privacy protection.
- δ is a non-negative number known as the failure probability, which introduces randomness, allowing the ε limit to be exceeded with a certain probability. A smaller δ implies a lower probability of failure for the privacy protection provided by algorithm M.

$$Pr\left[M\left(d\right)\in S\right]\leq e^{\varepsilon}Pr\left[M\left(d'\right)\in S\right]+\delta \qquad (1)$$

C. Differential Private Stochastic Gradient Descent (DP-SGD)

Differentially Private Stochastic Gradient Descent (DP-SGD) [4] is a differential privacy technique based on a variant of the original definition with the addition of δ, developed by Dwork et al. [8]. The primary purpose of DP-SGD is to provide robust protection for sensitive data, ensuring that their privacy is maintained throughout the training process. In machine learning, where vast quantities of data are often required to train models effectively, it becomes crucial to address the potential inclusion of sensitive information, such as personal identities, within these datasets. Therefore, DP-SGD aims to control the impact of the training data on the results during the training process, ensuring the privacy of sensitive information. The algorithm mainly restricts gradient updates, which are different from the original. The gradients are first clipped, then the noise is added to generate new gradients for subsequent updates.

III. Related Works

In terms of image obfuscation technology, some scholars have investigated this field. The author of [9] proposed the research on "Perceptual Indistinguishability-Net (PI-Net)", which explored how to apply differential privacy to facial image obfuscation. They achieved image obfuscation by introducing noise and manipulating the semantic attributes of the face to generate realistic facial images. In the work "Image Obfuscation for Privacy-Preserving Machine Learning" by Mathilde Raynal et al. [10], they contributed to quantifying image privacy through a series of obfuscation techniques

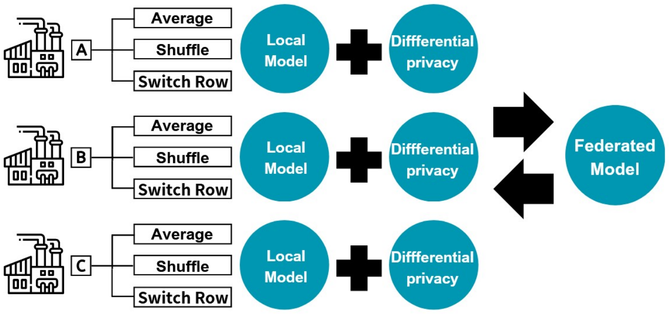


Fig. 2: The overview of PIDS.

such as Mixing, Pixel Grafting, Pixel Shuffling, adding noise, and Pixelizing. In [11], image obfuscation techniques (i.e., pixelating, blurring, and masking) have been developed to protect sensitive image information. The authors also implemented the proposed framework to reconstruct obscured facial images and evaluated the reconstructed images using structural or identity-based metrics. [12] aimed to enhance digital image privacy through watermarking and a QR code-based visual cryptography approach. This method, requiring no expansion, produces aesthetically pleasing QR codes for sharing meaningful data. Initially, the original secret image is watermarked with elements like copyright logos or signatures. Then, halftoning is applied to control pixel expansion. The halftoned image is divided into two shares using Visual Cryptography (VC) and combined with a QR code for meaningful content. [13] introduced a novel image obfuscation technique, merging a variational autoencoder (VAE) with random non-bijective pixel intensity mapping to protect medical image information. This method allowed deep learning models to be trained on obfuscated images without significant computational overhead, ensuring defense against visual and AI-based reconstruction attacks. In [14], we have published image obfuscation methods for industrial defect detection systems. However, only image obfuscation methods are not enough to protect data and AI models. Thus, this paper integrates federated learning and differential privacy with obfuscation methods to enhance data and model privacy protection for industrial defect detection systems. Furthermore, this paper provides more complete experiments to show the effectiveness of the proposed method.

IV. Privacy-preserving Industrial Defect Detection System

We propose a Privacy-preserving Industrial Defect Detection System (PIDS), as shown in Fig. 2. First, the input image is applied to a series of obfuscation operations, i.e., average, shuffle, and switch row methods. These procedures primarily hide image information from human eyes but can still be recognized by AI models. The obfuscated images are the training data for the classification model used for defect classification. To enhance data privacy, PIDS is developed as a federated learning architecture. Additionally, we use DP-SGD as an optimizer to preserve model gradients during training.

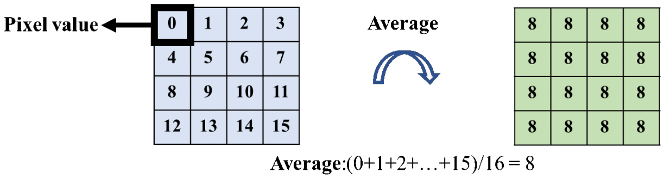


Fig. 3: An example of the average method.

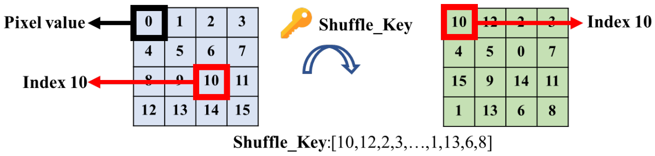


Fig. 4: An example of the shuffle method.

A. Image Obfuscation Module

1) Average Method: In the average method, a mask of a specified size moves from top to bottom and left to right. Each movement will calculate the average value from the mask pixels. Then, we replace all mask pixels with the average value. Fig. 3 demonstrates the process of the average method. The average method can blur images.

2) Shuffle Method: Based on [3], we design the shuffle method. The shuffle method randomly generates a Shuffle_Key whose length is the area of the mask. Then, it will move the mask from top to bottom and left to right and sort the mask pixels according to the value in Shuffle_Key. Fig. 4 is an example of the shuffle method. The shuffle method achieves image obfuscation.

3) Switch Row Method: The switch row method randomly generates a Row_Key whose length is the height of the image. Then, it will sort the row of the image according to the value of Row_Key. Fig. 5 is the process of the switch row method. The switch row method can achieve better image obfuscation.

B. Image Classification Model

For different complex classification tasks, we choose Convolutional Neural Network (CNN), ResNet50 [15], and MobileNetV2 [16] as the classification models. CNN is a basic classification model. ResNet50 is a complex model that can handle difficult image classification tasks. MobileNetV2 is a lightweight model suitable for embedded systems in manufacturing environments.

We first perform feature extraction in CNN through multiple convolutional layers and max pooling layers. Then, the feature maps are flattened into one-dimensional vectors using a flattening layer. Finally, we build fully connected layers for the final classification predictions and employ dropout layers to reduce overfitting.

In ResNet50 and MobileNetV2, we modify the architecture to improve the performance. First, we remove the original classification layers of ResNet50 and MobileNetV2. Next, we connect the output of ResNet50 and MobileNetV2 to a GlobalAveragePooling2D layer, which reduces the feature

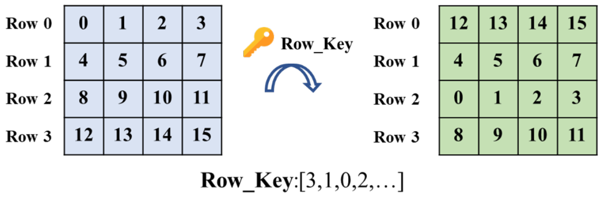


Fig. 5: An example of the switch row method.

maps' spatial dimensions. Subsequently, we add a dense layer with 128 nodes and ReLU activation to extract higher-level features. Finally, we replace the original classification layers with our dense layer, consisting of six nodes and a softmax activation for generating class probabilities.

C. Federated Learning Model

We develop PIDS as a federated learning architecture to enhance data privacy. Federated learning consists of a server and multiple clients. The server aggregates the models while the clients perform the training. Initially, the clients start training based on predefined parameters. Once the training is complete, the trained models are sent to the server for aggregation, completing one round of federated learning. The server distributes the aggregated model back to the participating clients for the next round of training. The iterative training, aggregation, and redistribution process continues until the desired model performance is achieved.

We build PIDS based on the Flower framework [17], an open-source framework for federated learning. Flower provides tools and APIs to simplify setting up federated learning systems, managing device communication, and aggregating model updates. First, we set up a Flower server to manage the federated learning process. Then, we define the server configuration, such as the IP address and port to listen to, and we define the model we will use. Once connected, clients can train their local models with their data. Flower handles the aggregation of model updates securely to ensure privacy and prevent information leakage. Using Flower's monitoring capabilities, we can monitor metrics, evaluate model performance, and manage federated learning tasks efficiently. By following these steps and leveraging Flower's features, we effectively implement federated learning in PIDS.

D. Differential Private Stochastic Gradient Descent (DP-SGD)

To safeguard the confidentiality of sensitive information, we further integrate differential privacy for PIDS. We employ the "DPKerasSGDOptimizer" of TensorFlow Privacy to merge differential privacy into our optimization process. The optimizer allows us to set several crucial parameters that control privacy protection while training our models.

- L2 Norm Clip: The L2 norm clip parameter applies an upper bound on the L2 norm of individual gradients during training. Limiting the scale of gradients prevents large updates that could unintentionally reveal sensitive information. We set the L2 Norm Clip to 1.0 to balance model performance and privacy preservation.
- Noise multiplier: Differential privacy relies on adding carefully calibrated noise to gradient updates. The noise multiplier parameter determines the scale of this noise, with higher values enhancing privacy at the cost of increased randomness in the learning process. To achieve a balance, we set the noise multiplier to 0.01, ensuring a reasonable level of privacy without significantly compromising model accuracy.
- Micro batch: Our federated learning approach leverages micro batches to compute gradients on smaller subsets of data, reducing the risk of data exposure. We configure the number of microbatch as 1, optimizing computational efficiency while maintaining differential privacy.
- Learning rate: The learning rate parameter checks the step size during gradient descent, influencing the convergence and stability of our models. We initialize the learning rate to 0.1, a reasonable value suitable for efficient optimization without risking divergence.
- Gradient Accumulation Steps: We utilize gradient accumulation steps to manage memory usage and optimize training with large batch sizes. This parameter determines the number of gradient accumulation steps before applying updates. We balance computational resources by setting gradient accumulation steps to 4 while ensuring effective model training.

By carefully configuring these differential privacy parameters within our federated learning setup, we balance privacy protection and model performance, enabling robust and privacy-preserving machine learning in distributed environments.

V. Experiment

We evaluated PIDS and compared it with CNN, ResNet50, and MobileNetV2 based on manufacturing and open datasets in the following experiments. We also analyze the effects of different noise multipliers since the noise multiplier is a critical factor in differential privacy.

A. The Experiment based on Manufacturing Dataset

We adopt the "AOI defect classification dataset" provided by AIdea [18] to evaluate PIDS. There are six classes, "normal, void, horizontal defect, vertical defect, edge defect, and particle," respectively. We resize the defective images from 512x512 to 224x224 to save computing resources.

First, we evaluate the effects of the obfuscation module. The resized images are applied with the average, shuffle, and switch row methods, with the average method's mask 2x2 and the shuffle method's mask 4x4. Fig. 6 illustrates the AOI defects before and after the image obfuscation module. The image obfuscation module achieves better results on the void, horizontal defect, and particle class. In Fig. 7, we use void class to compare the proposed obfuscation methods with the obfuscation method of [3]. Our method achieves a higher degree of obfuscation in the original images, making it more

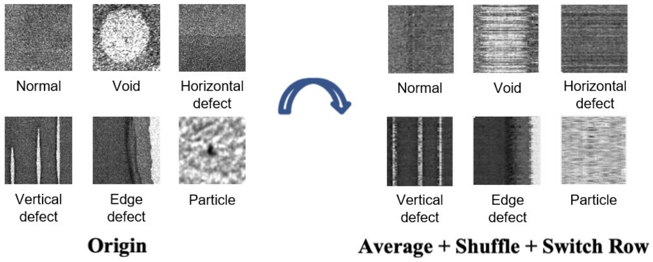
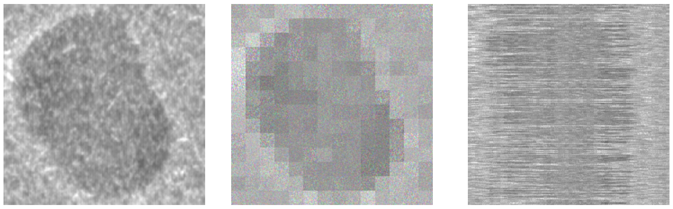


Fig. 6: AOI defect image after applied image obfuscation module.



(a) Original

(b) Learnable image encryption from [3]

(c) Our proposed obfuscation method

Fig. 7: Comparison of obfuscation methods.

challenging for the human eye to identify them as belonging to the void class.

Then, we compare the classification performance of CNN, MobileNetV2, and ResNet50. 80% of the obfuscated dataset goes into the training set and 20% of the obfuscated dataset goes into the testing set.

According to Table I, when employing image obfuscation, there is a slight decrease in the accuracy of the three models, yet the overall impact is deemed insignificant. CNN, in particular, dropped only 0.2%. That is, PIDS can balance between privacy protection and model performance.

B. The Experiment based on CIFAR-10

We also conduct experiments on the CIFAR-10 dataset [19]. For the CIFAR-10 experiments, we do not use CNN as the training model since it does not perform well on CIFAR-10. Instead, we choose MobileNetV2 and ResNet50 models and use DP-SGD as the optimizer to protect data privacy. We train the models using the original image size of 32x32.

Next, we evaluate the effectiveness of the obfuscation module. Fig. 8 and Fig. 9 demonstrate the impact of the average and shuffle methods on CIFAR-10. We can observe that the obfuscated images are difficult for human eyes to recognize.

Then, we compare the performance of MobileNetV2 and ResNet50. 80% of the obfuscated dataset is used for training, and the remaining 20% is used for testing.

The experimental results in Table II indicate that employing our PIDS system on CIFAR-10 could potentially lead to a decrease in accuracy when simultaneously utilizing all three methods (average, shuffle, and switch row). Therefore, we individually evaluate the accuracy of each method. The results show that the average and shuffle methods perform the best. Hence, we also combine these two methods to evaluate their accuracy on CIFAR-10. The final results indicate that combining average and shuffle methods can achieve the best accuracy. That is, selecting appropriate obfuscation methods in PIDS can protect data privacy while maintaining reasonable accuracy in image classification tasks.

TABLE I: Experimental results of different models based on manufacturing dataset

model and methods	accuracy	loss
CNN	0.938	0.254
CNN with three methods	0.936	0.270
MobileNetV2	0.946	0.330
MobileNetV2 with three methods	0.936	0.280
ResNet50	0.982	0.072
ResNet50 with three methods	0.965	0.111

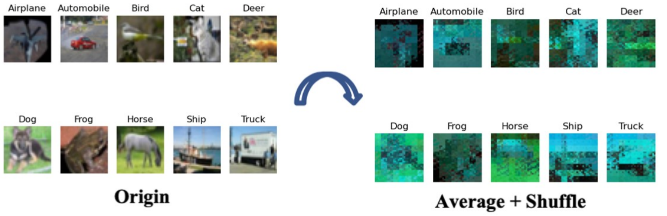


Fig. 8: CIFAR-10 images processed by average and shuffle.

C. The Experiment based on Noise multiplier

In addition to testing different datasets, we adjust the noise multiplier of the DP-SGD optimizer with the CNN, ResNet50, and MobileNetV2 models based on the manufacturing dataset. The noise multiplier value affects the level of privacy protection, with higher values indicating more robust protection but potentially lower accuracy. These experiments investigate the impact of different noise multiplier values on model accuracy.

We perform experiments with varying noise multiplier values. Initially, we set the noise multiplier value to 0.01, which is determined to provide higher accuracy in our experiments. Then, we further increase the noise multiplier value to 0.05 to test its amplified effect on different models.

According to the results in Table III, the accuracy of different models decreases to varying degrees as the noise multiplier value increases. Among them, CNN performs the best, while MobileNetV2 performs the worst. Selecting an appropriate noise multiplier value can protect data privacy to a certain extent while maintaining reasonable model accuracy.

Overall, different models require a trade-off between privacy protection and accuracy. Further research can explore alternative privacy protection methods and optimization strategies to find a better balance, ensuring the simultaneous consideration of data privacy and model accuracy.

VI. Conclusion

In this paper, we investigated the data privacy problem that came with industrial AI development. We presented a

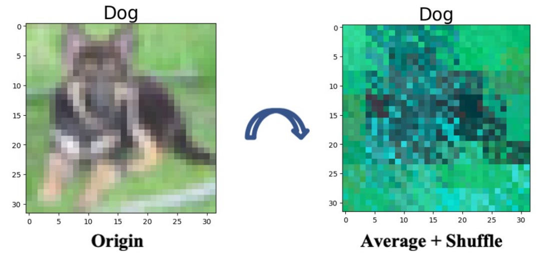


Fig. 9: Dog image of CIFAR-10 processed by average and shuffle.

TABLE II: Experimental results of MobileNetV2 and ResNet50 based on CIFAR-10 dataset

model and methods	accuracy	loss
MobileNetV2	0.740	0.771
MobileNetV2 with average	0.721	0.832
MobileNetV2 with shuffle	0.706	0.865
MobileNetV2 with switch row	0.473	1.471
MobileNetV2 with three methods	0.463	1.503
MobileNetV2 with average and shuffle	**0.710**	0.87
ResNet50	0.814	0.825
ResNet50 with average	0.790	0.848
ResNet50 with shuffle	0.803	0.781
ResNet50 with switch row	0.594	0.788
ResNet50 with three methods	0.597	1.369
ResNet50 with average and shuffle	**0.801**	0.78

TABLE III: Experimental results of different noise multipliers based on manufacturing dataset

model and methods	noise multiplier	accuracy	loss
CNN with three methods	0.01	0.936	0.270
CNN with three methods	0.05	**0.8913**	0.2923
MobileNetV2 with three methods	0.01	0.936	0.280
MobileNetV2 with three methods	0.05	0.25	3.573
ResNet50 with three methods	0.01	0.965	0.111
ResNet50 with three methods	0.05	0.34	1.67

Privacy-preserving Industrial Defect Detection System (PIDS), which integrated image obfuscation, differential privacy, and federated learning architecture. We also evaluate PIDS on the manufacturing dataset and CIFAR-10 dataset. Experiments and analysis showed that PIDS can achieve a high accuracy of 96.5% in defect image classification. PIDS only drops 0.2% accuracy while applying image obfuscation. That is, PIDS achieved high accuracy while preserving privacy both model-wise and data-wise. As a result, PIDS solved the problem of weak obfuscation and the model inversion attack.

ACKNOWLEDGEMENTS

This work is sponsored by the National Science and Technology Council (NSTC) under the projects NSTC 110-2222-E-008-008-MY3 and NSTC 112-2622-8-A49-021.

REFERENCES

[1] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[2] Nishant Vishwamitra, Bart Knijnenburg, Hongxin Hu, Yifang P Kelly Caine, et al., "Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 39–47.

[3] Masayuki Tanaka, "Learnable image encryption," in *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 2018.

[4] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, "Deep learning with differential privacy," in *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*. Springer, 2006.

[6] Cynthia Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[7] Cynthia Dwork, Aaron Roth, et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[8] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor, "Our data, ourselves: Privacy via distributed noise generation," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006.

[9] Jia-Wei Chen, Li-Ju Chen, Chia-Mu Yu, and Chun-Shien Lu, "Perceptual indistinguishability-net (pi-net): Facial image obfuscation with manipulable semantics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6478–6487.

[10] Mathilde Raynal, Radhakrishna Achanta, and Mathias Humbert, "Image obfuscation for privacy-preserving machine learning," *arXiv preprint arXiv:2010.10139*, 2020.

[11] Jimmy Tekli, Bechara Al Bouna, Raphaël Couturier, Gilbert Tekli, Zeinab Al Zein, and Marc Kamradt, "A framework for evaluating image obfuscation under deep learning-assisted privacy attacks," in *2019 17th International Conference on Privacy, Security and Trust (PST)*, 2019.

[12] Akanksha Arora, Hitendra Garg, Shivendra Shivani, et al., "Privacy protection of digital images using watermarking and qr code-based visual cryptography," *Advances in Multimedia*, 2023.

[13] Andreea Bianca Popescu, Ioana Antonia Taca, Anamaria Vizitiu, Cosmin Ioan Nita, Constantin Suciu, Lucian Mihai Itu, and Alexandru Scafa-Udriste, "Obfuscation algorithm for privacy-preserving deep learning-based medical image analysis," *Applied Sciences*, 2022.

[14] Hao-Yuan Chen, Yu-Chen Yeh, Makena Lu, and Chia-Yu Lin, "Image confusion applied to industrial defect detection system," in *IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2022.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[17] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al., "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[18] Taiwan Industrial Technology Research Institute, "Defect classifications of aoi," https://aidea-web.tw/topic/285ef3be-44eb-43dd-85cc-f0388bf85ea4.

[19] Vinod Nair Alex Krizhevsky and Geoffrey Hinton, "The cifar-10 dataset," https://www.cs.toronto.edu/~kriz/cifar.html.