Exploring Image Reconstruction Attack in Deep Learning Computation Offloading

Hyunseok Oh
ohsai@snu.ac.kr
Seoul National University
Seoul, South Korea

Youngki Lee
youngkilee@snu.ac.kr
Seoul National University
Seoul, South Korea

ABSTRACT

Deep learning (DL) computation offloading is commonly adopted to enable the use of computation-intensive DL techniques on resource-constrained devices. However, sending private user data to an external server raises a serious privacy concern. In this paper, we introduce a privacy-invading input reconstruction method which utilizes intermediate data of the DL computation pipeline. In doing so, we first define a Peak Signal-to-Noise Ratio (PSNR)-based metric for assessing input reconstruction quality. Then, we simulate a privacy attack on diverse DL models to find out the relationship between DL model structures and performance of privacy attacks. Finally, we provide several insights on DL model structure design to prevent reconstruction-based privacy attacks: using skip-connection, making model deeper, including various DL operations such as inception module.

KEYWORDS

Computation offloading; Privacy; Deep learning

ACM Reference Format:
Hyunseok Oh and Youngki Lee. 2019. Exploring Image Reconstruction Attack in Deep Learning Computation Offloading. In *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications (EMDL '19), June 21, 2019, Seoul, Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3325413.3329791

1 INTRODUCTION

With advances in smartphone and wearable device technologies, plenty of sensor-enabled life-immersive applications are emerging, providing proactive and situational services to their users. Many of them adopt deep learning (DL) techniques to monitor user activities, emotions, and surroundings in an accurate manner. However, the computational complexity of DL model-based inference makes it difficult to apply such techniques on resource-constrained devices. Although on-device optimization techniques could be useful to address such problems, it is inevitable to offload computation in many cases, especially if the devices have highly limited computing power or run computation-intensive models.

EMDL '19, June 21, 2019, Seoul, Korea

ACM ISBN 978-1-4503-6771-4/19/06…$15.00
https://doi.org/10.1145/3325413.3329791

DL offloading, however, raises serious privacy concerns as private user data (e.g. images, sound motion data, physiological signals) are sent to the external server [4]. One possible solution to protect private data is to encrypt or anonymize the data before transmitting it to a cloud computing service [28]. However, such methods often require removal of detailed features or perturbation of the data, affecting the accuracy of DL inference. Another promising approach is sending intermediate feature values (not the raw input data), which cannot easily be converted back to the original data.

In this paper, we explore a possible privacy attack called *reconstruction attack* in DL offloading, with which an adversary can rebuild the original data (e.g. private pictures) from the intercepted intermediate feature data. The key idea of our attack is to leverage the concept of a gradient descent to find the image producing the same intermediate feature data as the original user image. Our method feeds a random variable into the DL model and extracts intermediate data from the same layer from which the intercepted intermediate data is extracted. With the loss function computed using the two intermediate data, we optimize the variable with gradient descent by backpropagating through the DL model.

Our study shows that the suggested attack can reconstruct images from intermediate outputs (generated after multiple DL layers) for ten different state-of-the-art DL models. The reconstruction attack is possible even with the intermediate feature values processed through more than 80% of the layers for VGG networks and MobileNet. Also, such an attack can be efficient: with a single Tesla K80 GPU, an adversary can, on average, reconstruct the original data in 7.7 seconds per extracted data. This suggests that an adversary can quickly rebuild a large number of images.

Based on the results, we provide some insights for building privacy-preserving models suitable for (partial) DL offloading. For instance, we found that incorporating skip-connections into the model structure makes the DL model significantly resistant to image reconstruction (after the first couple of layers, which can be handled by the client devices).

The contribution of our paper is as follows:

- We introduce a new type of privacy attack in DL offloading, i.e. reconstruction attack, which rebuilds the raw user image from the intermediate data of DL pipeline transferred from the user device.
- We propose a new image reconstruction method, which rebuilds original input images from intermediate feature values of DL computation. Image reconstruction is possible on arbitrary DL models, taking about 7.7 seconds per image with a single Tesla K80 GPU.
- We show that our image reconstruction method can successfully regenerate input images from very deep layers of

various models (e.g., VGG Network, MobileNet) and discuss useful insights to design privacy-preserving DL models and offloading mechanisms.

2 IMAGE RECONSTRUCTION ATTACK

In common DL offloading scenarios, a user either sends raw data to the server or executes the front part of DL pipeline (i.e. user-side model) first and then sends the intermediate feature data to the server. The server runs the full or the later part of the DL pipeline (i.e. the server-side model) and sends the inference output to the user device.

In this offloading process, reconstruction attack can be a serious privacy threat. Our attack method, for instance, can intercept and use the intermediate feature data and the user-side model data in some scenarios. Firstly, the intermediate features can be intercepted through the Man-in-the-Middle attack in the network, from malicious applications, or by the administrator of the server. Also, an adversary can fetch the model as DL-based services typically use DL models open to the public. In the case that custom DL models are used, an adversary can hack into a vulnerable client device and extract the user-side model data. The user-side model data are often identical across all the client devices, and model protection measures are harder to apply to the user-side model due to the performance issues of the device.

3 RECONSTRUCTION METHOD

There are several possible approaches to reconstruct the original input image when an adversary has access to the intermediate feature data of the DL pipeline. One straightforward method is creating an inverse mapping from the model structure and the corresponding parameters. However, common DL layer operations like max pooling, fully connected, and convolution are not reversible. Another approach is using a generative neural network to approximate an inverse mapping. However, this approach needs a large set of training data to map the inputs to the intermediate outputs accurately.

We developed an iteration-based reconstruction method for the attack (as shown in Figure 1). The main idea is to optimize an input variable through backpropagation on the DL model to create an image similar to the original image. First, the method creates a random input variable and feed it into the model and extract intermediate data from the same layer the intercepted intermediate data is fetched. Then it evaluates loss function between created intermediate data and intercepted intermediate data, and backpropagate to update the input variable. Iteration of this process makes the input variable similar to the original input image.

The iteration-based reconstruction method consists of a loss network and an image variable. The loss network is the neural network which an intermediate data are calculated. The input image is generated on an image variable through backpropagation.

Suppose we have an original intermediate data of DL pipeline from a user input. Let early part of the DL pipeline from the input to l-th layer as mapping f^l, an image variable x, the original image $\hat{x}$. Our method feeds the image variable x into the DL model and extracts the intermediate data $f^l(x)$ from the same layer the original intermediate data $f^l(\hat{x})$ is created. Then, it computes the reconstruction loss L_R, which is defined as L2 loss between the two

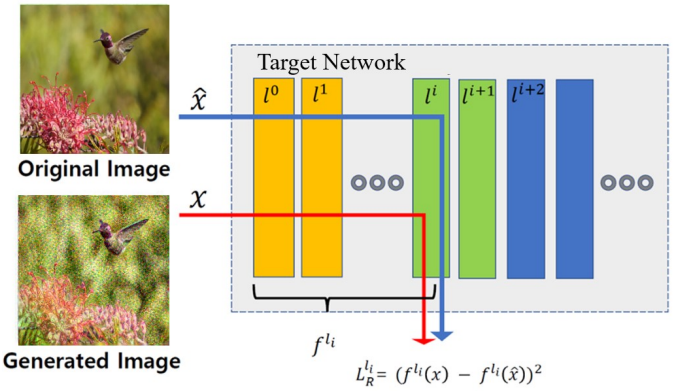


Figure 1: Representation of image reconstruction process. It shows reconstruction loss of i-th layer of original image and generated image.

intermediate data. L_R is precisely defined as follows.

$$L_R = \sum_{l \in L} (f^l(x) - f^l(\hat{x}))^2$$

Let the total variation loss of the image variable as L_{TV}. $x_{i,j}$ is (i,j)-th pixel value of x.

$$L_{TV} = \sum_{i,j} ((x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2)$$

The objective function is the lagrange multiplier of two L_R and L_{TV}[24]. Now the problem becomes optimizing the image variable x^* as to minimize the objective function to obtain the original image.

$$x^* = \text{argmin}_x (L_R + \lambda L_{TV})$$

We can obtain the image variable through backpropagation if we know the model structure and parameters of f^l mapping.

4 EVALUATION

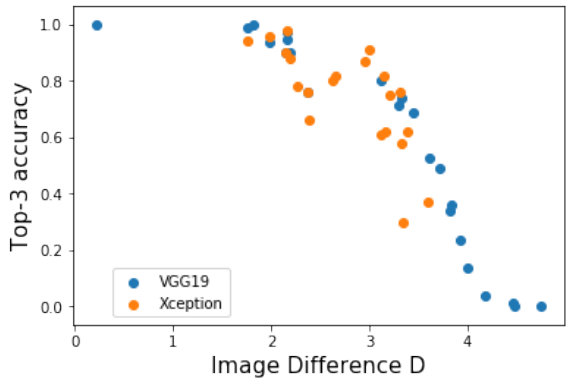
4.1 Experimental Setup

Models. We have simulated the privacy attack in diverse DL models including various DL components like convolution, pooling, residual layer, skip-connection, depth-wise convolution, and inception module. Target models of our attack simulation are VGG16 and VGG19 [27], ResNet50 [12], InceptionV3 [30], InceptionResNetV2 [29], Xception [5], MobileNet [14], MobileNetv2 [25], DenseNet121 [15], and NASNetMobile [33], which are provided from Keras, pre-trained with the ImageNet dataset [8].

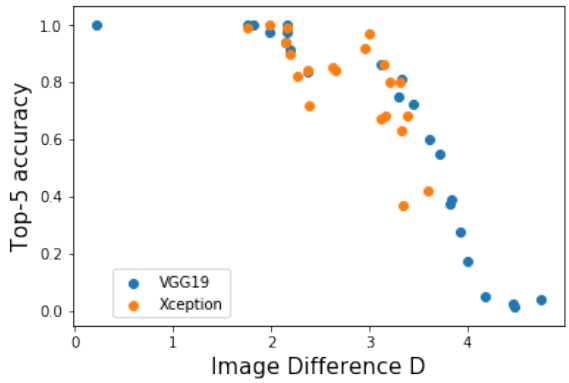
Input Data. We have used 20 images in our experiment, 10 of them chosen from the result of random keyword-based Google image search, and rest, from the ImageNet dataset.

Default Hyperparameter Setting. We have chosen the following default values for the initial hyperparameter settings: learning rate 5×10^1, lambda of objective function 1×10^{-3}, number of iterations 500, and the optimizer, Adam [16].

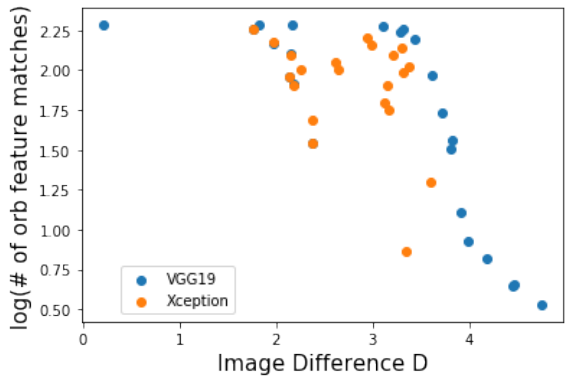
Metric. We define and use a new metric based on PSNR [13] to evaluate the quality of the image reconstruction. Our metric modifies PSNR to represent the difference between two images



(a) Top-3 accuracy



(b) Top-5 accuracy



(c) Number of ORB feature matching

Figure 2: Relation of image difference and indistinguishability from DL models, represented by top-k accuracy,(a)(b) number of ORB feature matching between original input image and rebuilt image.(c)

and satisfy distance properties such as non-negativity, symmetry, triangle inequality. Specifically, the metric for the two images, $\hat{x}$ and x, is denoted as $D(x, \hat{x})$. The smaller $D(x, \hat{x})$ is, the more similar the two images are. $D(x, \hat{x})$ is defined as the following:

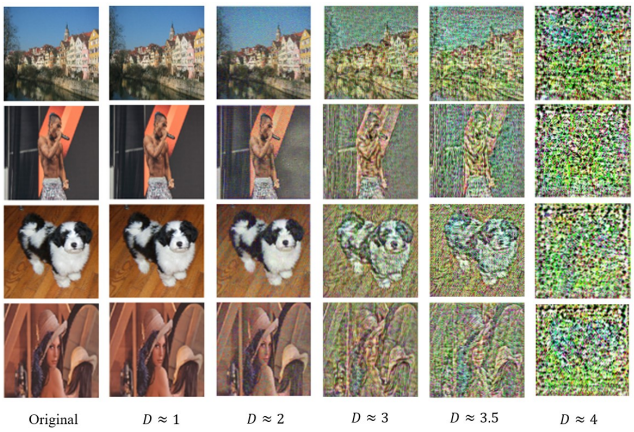


Figure 3: Reconstruction of 4 input images from intermediate features of VGG19 Loss network. Rebuilt images sorted by image difference, rounded on the first decimal place.

$$D(x, \hat{x}) = \log(\frac{1}{HWC} \sum_{i,j,c} |x_{i,j,c} - \hat{x}_{i,j,c}|^2 + 1).$$

where H, W, C are the height, width, and the number of channels of the image and the pixel values of $x, \hat{x}$ ranges from 0 to 255. The two image are completely identical if $D(x, \hat{x})$ is zero. D_{batch} is the mean value of the image differences among a set of images.

We adopt the above new similarity metric instead of using commonly used metrics like SIFT [18], Oriented Fast and rotated BRIEF (ORB) [23], Color histogram [22]. This is mainly because our image reconstruction method aims to recreate pixel-by-pixel similar images, so the metric based on pixel-wise difference is more valid. Also, images reconstructed from our method typically contain a lot of noises, so SIFT or ORB-based image similarity metrics does not work well. Our metric is conceptually similar to PSNR, but the direct use of PSNR is not suitable as it simply represents the signal to noise ratio for a single image, not the similarity between the two images.

Validity of our metric. We have validated our metric with two experiments. We compare inference results of original and reconstructed images on various DL models. We set the prediction result from the original image as ground truth to get top-k accuracy of the reconstructed image. We also compute the number of ORB feature matches between the original and the reconstructed images.

There is a strong negative correlation between image difference and indistinguishability from DL models. Figure 2a-2b show that the rebuilt image and the original image are likely to be indistinguishable to neural networks when the image difference is low.

Image difference is shown to have a strong negative correlation with the number of ORB feature matches. Figure 2c shows that the reconstructed image tends to preserve the features of the original image when the image difference is low.

Figure 3 shows the relationship between the image difference metric and the perceptual difference. If the image difference is lower than 2, then it is almost indistinguishable from the original image. If the metric is higher than 4, the original image seems to be not

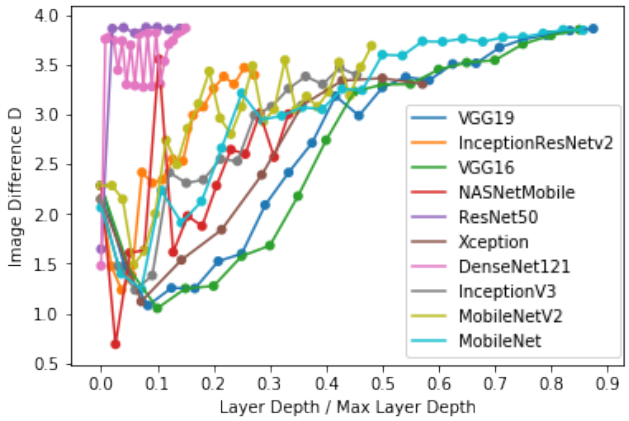


Figure 4: Relation between layer depth and D_{batch} from image reconstruction at the layer, on 10 distinct DL models. X axis is layer depth of certain layer operation divided by maximum layer depth of DL model. Image difference tend to increase as layer gets deeper on all the models.

reconstructed at all. Overall, as the image difference gets higher, it gets harder to think the two images as similar. We can think of 4 as a possible threshold of image reconstruction.

4.2 Reconstructability for Various Models

We have experimented image reconstruction on diverse DL models and found out that image reconstruction was possible on a very deep layer of some models. Layers with D_{batch} over 4 is not recorded, according to the experiments on our metric.

Figure 4 shows the relationship between layer depth and D_{batch} on 10 distinct target networks. Layer depth represents an approximate depth of a DL operation on the network. Layer depth gets deeper on a group of DL operations like convolution cell, pooling, inception module, residual block, and reduction block. Maximum layer depth of a DL model is a maximum of layer depth value of all the DL operations on the model. If a single layer depth has several DL operations to extract the intermediate data, then the smallest D_{batch} from them is chosen to represent the layer depth.

Quality of the reconstructed image tends to decrease as the intermediate data is extracted from deeper layers. All the graphs in Figure 4 show that the image difference tends to increase as intermediate data is extracted from the deeper layer.

Table 1 shows that reconstruction-based privacy attack is possible on 85 ~ 87.5% of the layers of VGG networks. VGG network structure is simply designed, mainly with the sequential arrangement of convolution and pooling layer operations. Therefore it is not optimized to extract only necessary features for inference, and lots of information on original image persists through deep layers.

Reconstruction attack does not work well even in shallow layers of the DL models with skip-connection. Image reconstruction is possible only on 2 ~ 15% of these models. ResNet adds output from previous residual block to current residual block output to pass it on to the next block. DenseNet concatenates outputs from all the previous cells with current cell output to pass it on.

Model	Percent
VGG19	87.5%
VGG16	85%
MobileNet	82.1%
Xception	57.1%
MobileNetV2	48.1%
InceptionV3	45.5%
NASNetMobile	33.3%
InceptionResNetV2	27.3%
DenseNet121	15.0%
ResNet50	2%

Table 1: Percentage of portion of DL model where image reconstruction is possible. Sorted by percentage.

In case of the inception module-based DL models, reconstruction attack is possible on a smaller portion of the network than VGG networks. The attack works only on 27.3 ~ 57.1% of inception module-based models. InceptionResNetv2 is a combination of residual connection and inception module, and image reconstruction is possible on less than 30% of the entire model structure. Xception is inception module-based network without skip-connection, and attack is possible on the largest proportion among them.

Reconstruction attack performs differently on various DL models optimized for resource-restricted devices. On MobileNet, which has the simplest structure of stacking sequence of 3x3 depthwise-separable convolution and 1x1 convolution, the attack is possible on 82.1% of the entire model. MobileNetV2 and NASNetMobile, with more complex cell structure including skip-connection, are more robust to the attack than MobileNet.

4.3 Effect of Hyperparameters

From this experiment, we observe the effect of hyperparameters on latency and output quality of the image reconstruction process. Hyperparameters of interest are the type of optimizer, learning rate, number of iteration, and lambda of the objective function. The experiment is done on an Ubuntu Linux instance with a single Tesla K80 GPU.

We can rebuild an original image from a single intermediate data in 7.7 seconds with a single Tesla K80 GPU when hyperparameters are set to values described below. Results in Table 2 shows that it is optimal when the learning rate is 5×10^{1}, and when the lambda is 1×10^{-3}. Higher iterations monotonically showed lower image difference, but the execution time increased proportionally to the number of iterations. D_{batch} increased marginally after 500 iterations, so we chose the number of iterations as 500. In the case of optimizer Adam and L-BFGS [17] showed similar D_{batch} values, but L-BFGS takes twice the execution time on the same number of iterations.

5 INSIGHTS FOR PRIVACY-PRESERVING MODELS

We provide insights to design DL model structure which is difficult to privacy attack. To counter reconstruction-based privacy attack on DL computation offloading system, we suggest reorganizing the user-side DL model structure to be resistant to attack. We below describe a few possible design guidelines for the model design.

Optimizer	Time(s)	D_{batch}
Adam	7.818	3.207
L-BFGS	17.784	3.171
RMSprop	8.533	3.633
SGD	8.365	3.771
Adagrad	8.329	3.458

(a) Optimizer

Learning rate	D_{batch}
1×10^{3}	4.585
5×10^{2}	4.181
1×10^{2}	2.924
5×10^{1}	2.762
1×10^{1}	2.819
1×10^{0}	3.278
1×10^{-1}	3.839
1×10^{-2}	3.980

(b) Learning rate

Iterations	Time(sec)	D_{batch}
100	1.558	3.470
200	3.081	3.357
300	4.614	3.282
400	6.153	3.240
500	7.751	3.212
600	9.204	3.189
700	10.748	3.162
800	12.296	3.155
900	13.832	3.144
1000	15.341	3.144

(c) Number of Iterations

lambda	D_{batch}
1×10^{-1}	3.395
1×10^{-2}	3.162
1×10^{-3}	2.755
1×10^{-4}	2.756
1×10^{-5}	2.967
1×10^{-6}	3.727

(d) Lambda

Table 2: Results of Hyperparameter experiments showing D_{batch} according to each hyperparameters. DL model used for experiment is Xception, (a)(b) in conv2d_1 layer and (c)(d) in conv2d_2 layer.

Skip-connections. It is crucial to use skip-connection on a DL model for privacy protection. Image reconstruction is possible only on the very first few layers of the DL pipeline for the DL models like ResNet or DenseNet that uses skip-connection as a primary technique for designing the model structure. Even for the DL models partially including skip-connection in the structure such as MobileNetV2, InceptionResNetV2, and NASNet, the image reconstruction is nontrivial compared to other similarly structured DL models without skip-connection. Therefore, using skip-connection for the DL model structure design makes the model highly resistant to the reconstruction attack.

Deeper Networks. Processing a large portion of DL computation pipeline on a user device reduces the possibility of the reconstruction attack. In this case, intermediate data available on the server is extracted from a deeper layer of the DL model, which makes reconstruction attack challenging. However, the user device usually has limited computing power and it is critical to carefully consider the tradeoff between the privacy protection and the performance of DL computation.

Complex Networks. It makes makes reconstruction attack more difficult to incorporate various DL operations such as an inception module on a single layer. A simple model structure like the VGG network or MobileNet, which has a single DL operation per layer, is likely to be compromised by our reconstruction attack. However, DL models with diverse DL operations on a layer like Xception or InceptionV3, are more resistant to the reconstruction attack even without skip-connection.

Cost Optimization Techniques. DL optimization techniques do not affect the performance of the reconstruction attack. The comparison between the MobileNet and VGG networks show that optimization techniques like depth-wise separable convolution or dimensionality reduction with 1x1 convolution hardly affect the portion of a DL pipeline to offload from which image reconstruction is possible. Comparison of InceptionResNetV2 and NASNetMobile also shows that the optimization techniques on the inception module does not affect attack performance.

Our insights for privacy protection can be applied as follows to offload VGG-19 model computation. Without considering privacy, the usual way would be to execute the front part of the VGG-19 network on a user device and runs the rest of the pipeline in the server. However, this typical way of splitting the VGG network is prone to image reconstruction attack, which makes the redesign of the network essential. We can apply our above-mentioned insights for the redesign; for instance, we can redesign convolution cells of the user-side model with skip-connections and inception module. Training parameters of the re-designed model can be done by applying transfer learning on the entire model. We expect that such a modified model can better protect privacy from the reconstruction attack. The effect of the model redesign needs to be further studied.

6 RELATED WORK

There have been several previous works on privacy attack on a neural network: For instance, a white-box membership inference attack with discriminative DL model determines whether a given input data is a member of training dataset or not [19]. Other work creates a meta-classifier to determine whether a training dataset of a DL model has a specific property, using a pattern of inference output data [2]. These attacks inspect the training dataset stored in DL models, thereby extracting sensitive information on the training dataset. Our work studies a different type of privacy attack, i.e., reconstructing the exact input data which created a given intermediate data.

The model inversion attack uses Stochastic Gradient Descent (SGD) to search an input image maximizing the confidence of a single output class [9]. Our work is closely related to this type of attack, but it has key differences. The model inversion attack focuses on visualizing a representative image of an output class, whereas our work target on reconstructing the exact input image which made a given intermediate data. Also, our work adopts a different objective function to rebuild the precise input image from the intermediate data of the DL pipeline.

Our image reconstruction method is broadly related to DL visualization and neural style transfer in terms of using SGD. DL visualization is to find an image visualizing the activation of a specific DL layer, with SGD [32]. Neural style transfer creates an artistic image by calculating gradient descent on a multi-objective function of the content image and artistic style image [10]. Our work applies SGD for a different problem domain, i.e., exploring privacy threats and building countermeasures.

Privacy protection measures for deep learning models are actively being studied. Differential privacy-aware approaches aim to protect sensitive information of training data [1, 20, 21]. Homomorphic encryption(HE)-based approaches makes entire intermediate

data on DL computation encrypted [3, 6, 11]. Distributed system and multi-party computation based methods aim to disperse model data or computation on several machines so an administrator of a single machine could not have full access on the entire process [7, 26]. Trusted Execution Environment(TEE)-based approaches aim to hide DL computation inside the TEE [31].

Compared to these, our countermeasure focuses on protecting against image reconstruction attack and requires absolutely zero computational overhead besides DL computation. These various privacy protection approaches are suggested to protect different privacy aspects, and most of them bring a significant computational overhead or accuracy degradation.

7 CONCLUSION

We introduce an input image reconstruction method using only intermediate data of DL pipeline, and the front portion of DL model data. We explain privacy attack on DL computation offloading system exploiting this method. We also present a valid image reconstruction quality metric called image difference. On single Tesla K80 GPU, we can rebuild input image in 7.7 sec. We experimented our privacy attack on various DL models to see the influence of model structures on image reconstruction quality. Image reconstruction was possible on 82.1 $\sim$ 87.5% of VGG networks and MobileNet and only 2 $\sim$ 15% on ResNet and DenseNet. We have inferred several insights to design DL model structure resistant to certain privacy attack. First, use skip-connection eagerly for the design. Second, make the model as deep as possible. Third, include diverse DL operations on a single layer depth like inception module.

ACKNOWLEDGMENTS

We thank Do-Il Yoon for providing valuable feedback to improve the quality of this paper. This work was supported by the National Research Foundation of Korea (NRF) grant (No. 2019R1C1C1006088).

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 308–318.

[2] Giuseppe Ateniese, Giovanni Felici, Luigi V Mancini, Angelo Spognardi, Antonio Villani, and Domenico Vitali. 2013. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *arXiv preprint arXiv:1306.4447* (2013).

[3] Fabian Boemer, Yixing Lao, and Casimir Wierzynski. 2018. nGraph-HE: A Graph Compiler for Deep Learning on Homomorphically Encrypted Data. *arXiv preprint arXiv:1810.10121* (2018).

[4] Deyan Chen and Hong Zhao. 2012. Data security and privacy protection issues in cloud computing. In *2012 International Conference on Computer Science and Electronics Engineering*, Vol. 1. IEEE, 647–651.

[5] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.

[6] Edward Chou, Josh Beal, Daniel Levy, Serena Yeung, Albert Haque, and Li Fei-Fei. 2018. Faster CryptoNets: Leveraging sparsity for real-world encrypted inference. *arXiv preprint arXiv:1811.09953* (2018).

[7] Morten Dahl, Jason Mancuso, Yann Dupis, Ben Decoste, Morgan Giraud, Ian Livingstone, Justin Patriquin, and Gavin Uhma. 2018. Private Machine Learning in TensorFlow using Secure Computation. *arXiv preprint arXiv:1810.08130* (2018).

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[9] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1322–1333.

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423.

[11] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*. 201–210.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Alain Hore and Djemel Ziou. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*. IEEE, 2366–2369.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45, 1-3 (1989), 503–528.

[18] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.

[19] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks. *arXiv preprint arXiv:1812.00910* (2018).

[20] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755* (2016).

[21] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable private learning with PATE. *arXiv preprint arXiv:1802.08908* (2018).

[22] Kalyan Roy and Joydeep Mukherjee. 2013. Image similarity measure using color histogram, color coherence vector, and sobel method. *International Journal of Science and Research (IJSR)* 2, 1 (2013), 538–543.

[23] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. 2011. ORB: An efficient alternative to SIFT or SURF.. In *ICCV*, Vol. 11. Citeseer, 2.

[24] Leonid I Rudin, Stanley Osher, and Emad Fatemi. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena* 60, 1-4 (1992), 259–268.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4510–4520.

[26] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 1310–1321.

[27] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[28] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[29] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[31] Florian Tramer and Dan Boneh. 2018. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287* (2018).

[32] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).

[33] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8697–8710.