Federated-Cloud Based Deep Neural Networks with Privacy Preserving Image Filtering Techniques

Isabelle Choi
Reservoir High School
Fulton, MD, USA
isabellechoi11@gmail.com

Qiyang Song
Tsinghua University, China
George Mason University, USA
songqy17@mails.tsinghua.edu.cn

Kun Sun
Center for Secure Information Systems
George Mason University
Fairfax, VA, USA
ksun3@gmu.edu

***Abstract*— Training Deep Neural Network (DNN) models often require significant computational resources due to the large dataset sizes and a huge number of parameters to be optimized. A cloud-based approach may be utilized to accommodate such resource needs with flexibility and efficiency. But, protecting data privacy is a challenge in such approaches. Most of the encryption-based approaches for providing privacy incurs substantial overheads. However, in many instances, only part of data information needs to be protected, and the level of privacy is often dependent on the application requirements. Various types of image filtering techniques are utilized to generate distorted DNN datasets with application-specific privacy requirements satisfied. In general, high distortion level provides strong protection on data privacy but degrades the DNN accuracy. To find the appropriate type and level of image filtering prior to the training process, we identify an image similarity metric that can be used as a DNN accuracy predictor as well as the distortion level indicator. Furthermore, to improve the DNN accuracy of highly distorted datasets, we propose a privacy-preserving federated-cloud DNN training/classification on multiple distorted datasets. Each cloud trains an independent DNN model with a different image filtering algorithm, and then the client combines and utilizes the multiple models to obtain a well-performing model. Experiments were conducted to validate the effectiveness of the proposed schemes.**

Keywords—Deep Neural Network, Cloud Computing, Privacy Protection, Image Filtering, Federated Learning

I. INTRODUCTION

Deep neural networks are being adopted in more and more application domains for their excellent pattern recognition capabilities including computer vision, image processing, natural language processing, etc. However, since the training of DNN models needs a large number of parameters to be optimized and huge datasets, we need to dedicate a significant amount of computational and storage resources to train a well-performing model. For resource-limited clients, outsourcing the training process to cloud servers (CSPs) is a popular alternative. Nowadays cloud based training and classification have been proposed as a viable training approach and there are several commercial cloud systems already available [32, 33].

One of the challenges in cloud based training/classification is to provide a required level of privacy protection on datasets used in the training and classification process. Recently, researchers have proposed many privacy-preserving machine learning models based on homomorphic encryption [15, 16, 17] and secure multiple party encryption [25, 26, 27]. However, they suffer from the limitations of significant computation and communication resource overheads.

Various levels of privacy protection requirement may exist depending on the sensitivity of the information included in dataset samples. In many instances, only a subset of information needs to be protected. For example, if a set of automobile images are being used for training a DNN for classifying their brands and models, only the license plate numbers on images need to be protected while the rest of the features can be disclosed for training purposes. In such scenarios, a subset of image filtering techniques may be effectively utilized for privacy-preserving machine learning by removing the features that need to be protected for privacy purposes.

Image filtering (or sometimes called distortion or deformation) techniques have been developed in the areas of image processing and computer graphics. They can modify or enhance an image. Figure 1 shows the overview of the approach for utilizing filtered image sets for DNN training. Even though image filtering can provide application-specific security protection, there are still several challenges to balance the achievable DNN accuracy and privacy protection level. Due to the overall downgrade of the image quality, the DNN models trained on distorted datasets may underperform than the cases where the original datasets are used. Specifically, the more distorted, the higher protection level but with lower DNN classification accuracy. Moreover, considering a number of resources needed for DNN model training, it would be desirable to make a decision on the type of filtering technique and the level of distortion prior to the actual training.

To address these issues, we identify a similarity metric to quantify distortion levels and DNN accuracies from datasets. Researchers have proposed many similarity metrics to indicate the difference between the original and filtered datasets, e.g. Mean Square Error (MSE) and Spatial Correlation Coefficient (SCC), etc. Since these metrics may provide some levels of indications on information hiding and achievable DNN accuracy, we can identify a similarity metric that is most correlated to achievable DNN accuracy. This metric may be utilized to select an appropriate image filtering algorithm and parameters prior to the actual training. Figure 2 shows the selection of image filtering algorithms. In this paper, we investigate 3 datasets and 33 different filtering algorithms , and conduct a correlation analysis to select a similarity that is mostly

978-1-7281-2319-6/19/$31.00 ©2019 IEEE

related to the achievable DNN accuracy. We find out the SCC metric is an optimal metric to reveal achievable DNN accuracy. Therefore, we utilize the SCC metric to make a decision on image filtering.

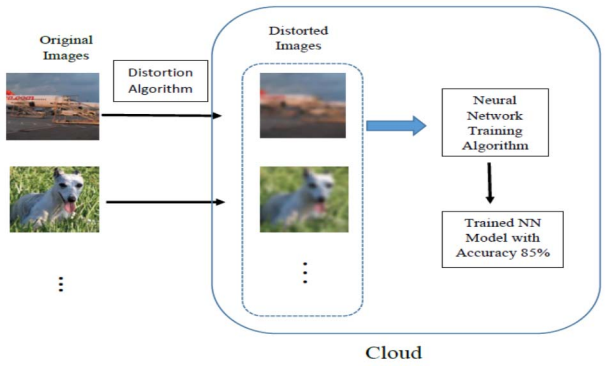


Fig. 1. Filtered images may be uploaded into the cloud for training a neural network model instead of the original images. Due to the degraded image quality and removed features, the DNN training on distorted images usually results in a lower accuracy DNN model.

Although we can make a trade-off between the privacy protection level and achievable DNN accuracy, if there are plenty of features that need to be protected, the achievable DNN accuracy may degrade significantly due to low image qualities and faint features. Therefore, we further propose a collaborative cloud-based learning approach where different filtered datasets are distributed to multiple clouds for independent training. The datasets in different clouds are generated with distinct filtering algorithms, which means the trained DNN models in different clouds extract different profiles of data. Therefore, the client can combine multiple DNN models to obtain a well-performing model. Note that data privacy can still be guaranteed since each cloud only stores a distorted dataset that satisfies information hiding requirements. Figure 3 shows the overview of our approach. Once the training process is completed in the clouds, the client utilizes the trained models in a way similar to that of the transfer learning. The client can obtain a well-performing DNN model by creating an additional top layer(s) on DNN models in the clouds and locally optimizing the parameters of the top layer(s).

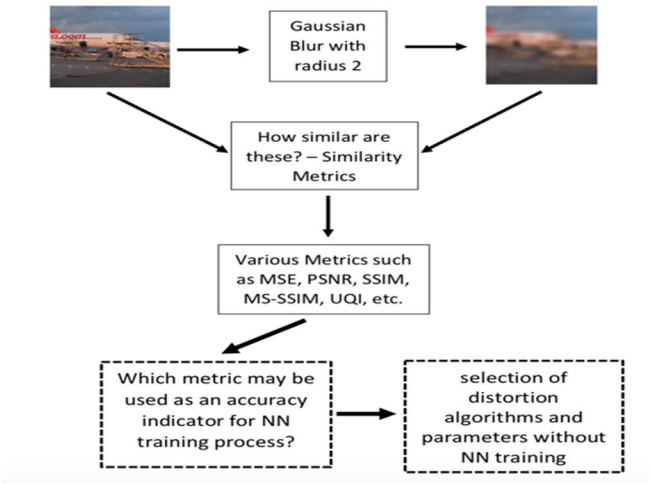


Fig. 2. Similarity metric that is highly correlated to the DNN training accuracy needs to be identified.

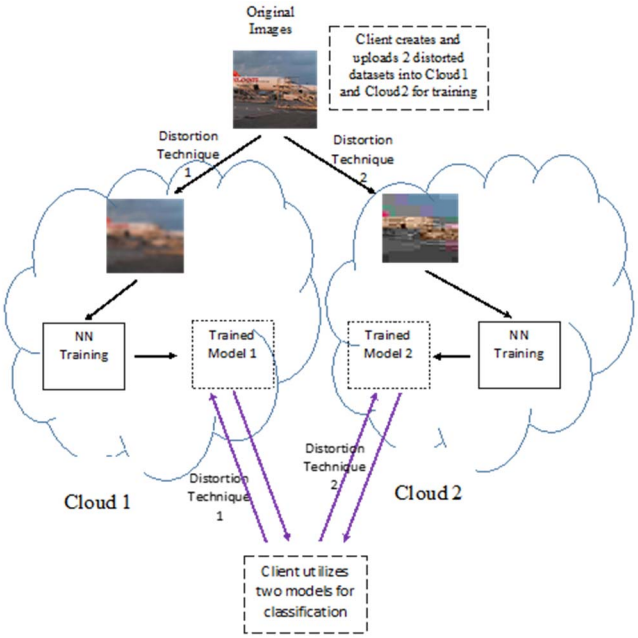


Fig. 3. Federated Cloud Based DNN Learning

An experiment was conducted with 33 different filtering algorithms applied to 3 datasets, i.e., CIFAR10, CIFAR100, and STL10, to find out whether there exists a candidate metric that may be used as an indicator for both image quality reduction and DNN accuracy. We incorporated 11 similarity metrics that have been proposed and studied in image processing [8]. The similarity metrics have been developed and used to measure the image difference in various aspects, and some of the elaborated metrics may be used as an effective indicator for image distortion (or information hiding) indicator [1-7, 9-14]. The results from our experiment show that one of the metrics, Spatial Correlation Coefficients [5], exhibits a high level of correlation with the achievable DNN accuracy with all 3 datasets. Especially, for similarity levels of 40% and up, it is shown that the correlation is very high. Moreover, we also carried out experiments to demonstrate the performance of federated-cloud learning approach. We applied 6 image filtering algorithms to the STL 10 dataset and measured the accuracy of two-cloud and four-cloud DNN models. It is shown that the accuracy in the combined models is higher than the single cloud model, and the accuracies for 4-cloud models are close to those with the original datasets.

Here, we summarize our contributions as follows:

- Identification of a similarity metric that can be used as a DNN accuracy predictor without actual training: This may be used in finding out a proper image filtering algorithm and distortion level parameter for filtering-based DNN training and classification at a cloud.
- Collaborative cloud-based learning approach: The DNN accuracy may be enhanced by utilizing distinct filtered datasets and building separate DNN models in multiple clouds.

This paper is organized as follows. Section II provides related work over privacy-preserving deep learning on clouds. Section III presents background on image filtering techniques

image and similarity metrics. In Section IV, we investigate multiple similarity metrics and identify a predictive similarity metric that may be utilized as an indicator for DNN accuracy. Federated-cloud approach is presented in Section V. Section VI provides the experiment details and results of the federated-cloud approach. Finally, the conclusions are presented in Section VII.

II. RELATED WORKS

With the advancement in cloud computing, there have been numerous works on providing privacy-preserving deep learning model and the training dataset in the cloud [23, 24]. Existing works are mostly based on secure multi-party computation [25, 26, 27] or homomorphic encryption [15, 16, 17]. However, these proposed schemes might lead to the expensive communication and computation overhead. A novel privacy-preserving model called CryptoNet was proposed in [28] which allows the data provider to outsource a homomorphically encrypted data to ensure that the datasets remain private. The model was also applied for evaluating deep convolutional neural network with two convolutional layers and two fully connected layers [29]. CryptoDL [29] is a combination of the convolutional neural network with leveled homomorphic encryption based on Shamir's secret sharing model. SecureML [30] was proposed for privacy-preserving neural network training using the stochastic gradient descent method. MSCryptoNet [31] was proposed to train the DNN model on the encrypted aggregated dataset from multiple data providers. However, most of the encryption based approaches suffer from the limitations of excessive resource requirements in terms of computation and communication.

III. BACKGROUND

A. Image Distortion Techniques

There are a significant number of image filtering/distortion techniques proposed and studied in the context of image processing [22]. We chose the following set of well-known techniques:

- Gaussian Blur: blurs (i.e., convolves) image by a Gaussian function. Levels may be specified by providing different values of radius. We consider radius of 1, 2, 3, and 4.
- Box Blur: blur the image by setting each pixel to the average value of the pixels in a square box extending radius pixels in each direction. We consider radius of 1, 2, 3, and 4.
- Median Filter: picks the median pixel value in a window with the given size. We consider the sizes of 3, 5, 7, and 9.
- Mode Filter: Picks the most frequent pixel value in a box with the given size. Pixel values that occur only once or twice are ignored; if no pixel value occurs more than twice, the original pixel value is preserved. Sizes of 3, 5, 7, and 9 are considered.
- CONTOUR: finds a closed curve of points or line segments, representing the boundaries of an object in an image. In other words, contours represent the shapes of objects found in an image.
- EMBOSS: each pixel of an image is replaced either by a highlight or a shadow, depending on light/dark boundaries on the original image. Low contrast areas are replaced by a gray background.
- FIND_EDGES: finds the boundaries of objects within images. It works by detecting discontinuities in brightness.
- JPEG Encoding with quality settings: uses a lossy compression based on the discrete cosine transform (DCT) [13]. The quality setting of the encoder (for example 50 or 95 on a scale of 0–100 in the Independent JPEG Group's library) affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether. We consider the quality settings of 1%, 5%, 10%, 20% and 30%.
- Privacy Preserving Photo Sharing (P3) [20]: designed to protect the privacy of JPEG images hosted on social media sites. The main idea is to split each JPEG into a public and a secret image. The public image contains most of the original image but is intended to exclude the sensitive information. The DC and AC coefficients of a JPEG image are replaced by a given threshold value if their original values are greater (or less if negative) than the threshold value. All the coefficients whose values are within the threshold remain without any change. We consider the threshold values of 150, 200, 250, and 300.

Figure 4 shows the example filtered images obtained from a sample image in STL10 dataset.

Fig. 4. Example images created by applying some of the image filtering techniques. (a) Original image (b) Gaussian Blurring with radius 2 (c) FIND_EDGES filter (d) EMBOSS filter (e) JPEG with 20% quality (f) Median Filter with radius 5

Additional filtering/distortion techniques may also be incorporated into the proposed mechanism to expand the scope of image distortions considered. Even different techniques may be combined to introduce more diverse distortion/filtering capabilities. For example, by applying a blurring and edge detection filter in sequence, fewer edges may be detected in the generated image which is thus more distorted.

B. *Image Similarity Metrics*

Several metrics for image similarity have been developed and some of them are shown to perform well even in the presence of severe noises and structural changes [1-7, 9-14]. These include the structural similarity index metric (SSIM) and multi-scale SSIM (MS-SSIM), Spatial Correlation Coefficients, etc. [5] These metrics may well be used as decision criteria for a distorted dataset's privacy protection. For example, by examining image samples from filtered datasets with different similarity metric values, the acceptable criteria (e.g., max SCC value should be 0.8) may be derived on the distorted dataset.

The following metrics are adopted and implemented in our experiments by using the Sewar library [21] APIs.

- Mean Square Error (MSE): cumulative squared error of corresponding pixel values between two images
- Root Mean Square Error (RMSE): square root of MSE
- Peak Signal-to-Noise Ratio (PSNR) [1]: measure of peak error between two images
- Structural Similarity Index (SSIM) [1]: compares the local patterns of pixel intensities between two images. considers image degradation as perceived change in structural information.
- Universal Image Quality Index (UQI) [2]: model image distortion as a combination of correlation loss, luminance distortion, and contrast distortion.
- Multi-scale SSIM (MS-SSIM) [3]: SSIM conducted over multiple scales through a process of multiple stages of sub-sampling, reminiscent of multiscale processing in the early vision system.
- Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS) [4]: Computes the quality of distorted image in terms of normalized average error of each band.
- Spatial Correlation Coefficient (SCC) [5]: tries to utilize the facts that more of the spatial information is concentrated in the high frequency domain. The higher correlation between the high frequency components of two images indicate more similarity between them. The Laplacian filter is used in calculating this metric.
- Spectral Angle Mapper (SAM) [6]: determines the pixel-by-pixel spectral similarity between two images by calculating the angle between the spectra.
- Pixel Based Visual Information Fidelity [7]: based on natural scene statistics and the notion of image information extracted by the human visual system.

IV. Identifying Predictive Similarity Metric

In this section, we focus on the performance of the chosen similarity metrics in terms of their possibility to be used as a predictor for the achievable DNN accuracy through the training process. If any of the metrics is shown to have a high correlation with the achievable NN accuracy, then it may be used in selecting the candidate filtering techniques and parameters (or datasets) without carrying out the actual NN training process.

First, 33 filtering techniques were applied to the 3 datasets, CIFAR10, CIFAR100, and STL10, to create 33 distorted datasets. In that process 11 similarity metric values were obtained for each pair of original and distorted images. The mean and standard deviations collected at the end.

Then, DNNs were trained with the original and distorted datasets to get the DNN accuracy for testing sets. The following are more details on the datasets and training process:

- CIFAR10 – consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. Resnet18 [18] architecture is used as a DNN model with the initial learning rate of 0.1 (decaying by 0.1 factor every 20 epochs) and with a total number of epochs of 60.
- CIFAR100 – This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The same DNN model and parameters are used as in CIFAR10.
- STL10 - image recognition dataset for developing various machine learning algorithms. Consists of 10 classes of 96x96 pixel color images. 500 images for training and 800 images for testing per class. Pre-trained Resnet18 model for ImageNet [19] was adopted and a transfer learning process was utilized to train DNN models with the initial learning rate of 0.01 (decaying by 0.1 factor every 7 epochs) and with a total number of epochs of 25.

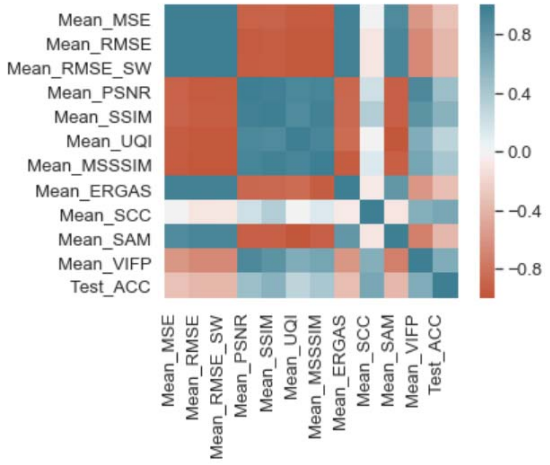


Fig. 5. Correlation Heatmap for similarity metrics (averages and standard deviatios) and DNN accuracies for STL10 datasets.

Figure 5 shows the correlation heatmap for 11 similarity metrics and DNN accuracies obtained for the STL10 dataset. It is shown that one of the similarity metrics, Spatial Correlation Coefficients [5], has the strongest correlation with the DNN accuracies. The correlations obtained between the 4 most dominant metrics and DNN accuracies are shown in Figure 6 for 3 datasets. The entire set of averages and standard deviations of the similarity metrics and DNN accuracies is given in the Appendix for STL10 dataset.

	Mean_SSIM	Mean_PSNR	Mean_SCC	Mean_VIFP	Test_ACC
Mean_SSIM	1.000000	0.937822	0.516180	0.750980	0.157620
Mean_PSNR	0.937822	1.000000	0.458991	0.914907	0.137454
Mean_SCC	0.516180	0.458991	1.000000	0.735224	0.750399
Mean_VIFP	0.750980	0.914907	0.735224	1.000000	0.494622
Test_ACC	0.157620	0.137454	0.750399	0.494622	1.000000

(a)

	Mean_SSIM	Mean_PSNR	Mean_SCC	Mean_VIFP	Test_ACC
Mean_SSIM	1.000000	0.927866	0.542178	0.746682	0.057029
Mean_PSNR	0.927866	1.000000	0.481424	0.916542	0.042885
Mean_SCC	0.542178	0.481424	1.000000	0.741304	0.663810
Mean_VIFP	0.746682	0.916542	0.741304	1.000000	0.399484
Test_ACC	0.057029	0.042885	0.663810	0.399484	1.000000

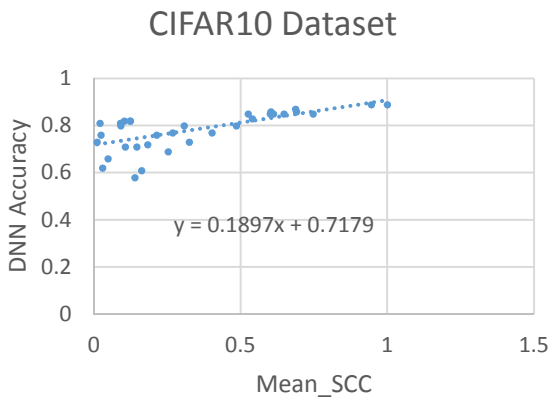
(b)

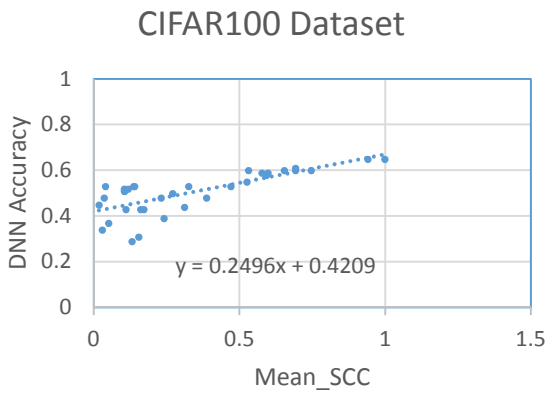
	Mean_SSIM	Mean_PSNR	Mean_SCC	Mean_VIFP	Test_ACC
Mean_SSIM	1.000000	0.968802	0.362393	0.823604	0.577769
Mean_PSNR	0.968802	1.000000	0.214836	0.908317	0.471985
Mean_SCC	0.362393	0.214836	1.000000	0.603327	0.677006
Mean_VIFP	0.823604	0.908317	0.603327	1.000000	0.628681
Test_ACC	0.577769	0.471985	0.677006	0.628681	1.000000

(c)

Fig. 6. Mean Spatial Correlation Coefficients (M-SCC) are shown to have the highest correlation with the DNN Accuracies. Tables were obtained for (a) CIFAR10 (b) CIFAR100 (c) STL10 with dominant metrics.

Figure 7 shows the actual correlation line and equation between Mean-SCC and DNN Accuracy for CIFAR10, CIFAR100, and STL10 datasets.





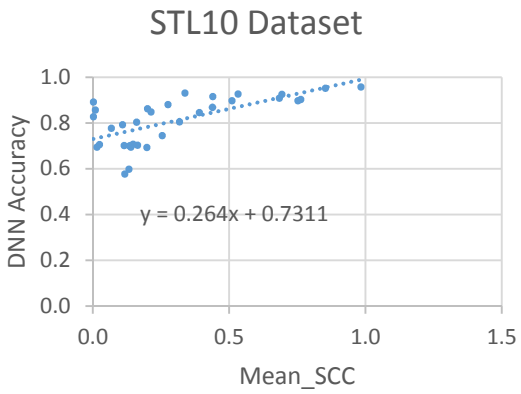


Fig. 7. The correlation between Mean SCC and DNN Accuracy for CIFAR10, CIFAR100, and STL10 datasets.

V. Federated Cloud Based DNN Models

In federated cloud based DNN model we need to decide which filtering techniques and parameters should be used to generate multiple datasets to be distributed to different clouds. We utilize the Mean-SCC predictor metric in making such decisions. To make a trade-off between data privacy and model performance, we assume that the following bounds are given with respect to the acceptable Mean-SCC values:

- MSCCmax : maximum Mean-SCC value allowed for any dataset to be chosen. This is assumed to be given at a design time by considering the privacy requirement. For example, if MSCCmax=0.8, then a dataset whose Mean-SCC value is higher than 0.8 shouldn't be used since the enough privacy protection is not provided.

- MSCCmin : minimum Mean-SCC value allowed for any dataset. This is obtained from the correlations obtained between Mean-SCC and DNN accuracies. For example, for 3 datasets we are considering in this paper, if the minimum acceptable DNN accuracy is within 7% of the best achievable accuracy with the original dataset, then MSCCmin=0.5 can be used.

The selection process we adopted is shown in Figure 8. In this process, the client selects several image filtering techniques and outsources filtered datasets to different clouds. Then, each cloud trains an independent DNN model with a filtered dataset. Due to the low quality of filtered images, the accuracy of each trained model is much lower than that of the original model. Since different trained models extract different data information, the client can merge multiple models into a well-performing model. Similar to the transfer learning, the client only needs to optimize the parameters of a few layers based on trained models. Therefore, the client can obtain a well-performing model with limited computation resources.

Figure 9 shows the two-cloud federated model. The client utilizes two clouds to train two DNN models. Once two clouds complete the training process with two filtered datasets, the client adds a classification layer whose input is the output of two models. The final layer is added as an additional layer which is trained at a client through a separate training process.

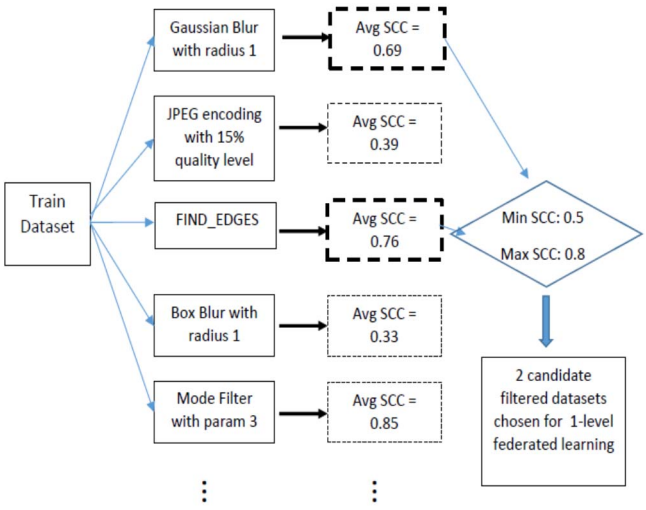


Fig. 8. 1-Level Federated Cloud Learning: 2 datasets are chosen by examining their average SCC metric values.

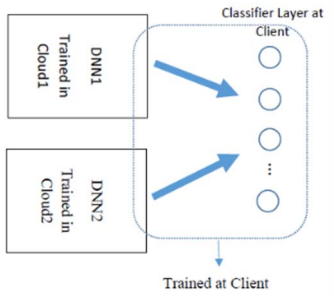


Fig. 9. 1-Level 2-Cloud Federated Learning. Only the top classification layer newly added on top is trained at clouds.

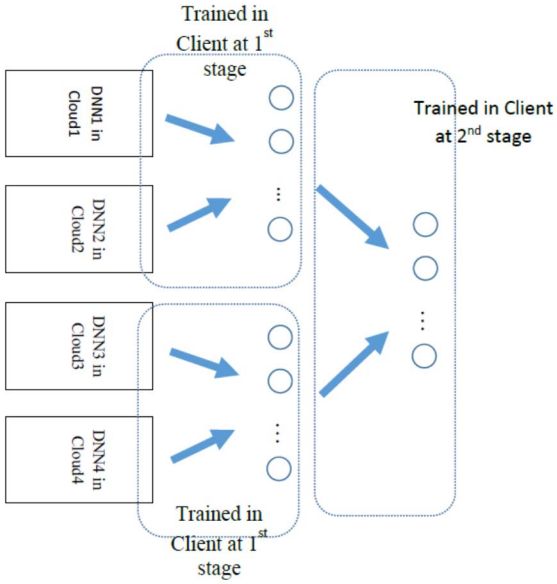


Fig. 10. 2-Level 4-Cloud Federated Learning. 2-stage training of the 2 additional classification layers are carried out in Client.

When we increase the number of clouds participating in the federated learning, the number of NN nodes in the classification layer would increase if we simply merge multiple DNNs with a single classification layer on top. This would result in a classification layer with linearly increasing number of input nodes. Since the number of input nodes in a DNN classification layer is typically large (e.g., 512 in ResNet18), simple merging would yield a classification layer with huge number of input nodes, which may introduce difficulty in the transfer learning process. Hence, we adopted the hierarchical multi-stage transfer learning approach in merging and training multi-level classification layers, whose effectiveness is shown through experiments. 2-Level Federated clouds example is shown in Figure 10 where the DNN models in 4 clouds are merged with two-level classification layers. Note that the training is conducted in 2-stages, in the first one two DNN models are combined and their first classification layer is trained, and in the second state the classification layers trained in the first stage will be combined through another final classification layer and will be trained.

VI. Federated Cloud Based DNN Experiments

To show the applicability of the federated cloud based approach we conducted experiments on STL10 dataset. In our experiment, we chose the candidate distorted datasets whose Mean-SCC values lie in between 0.5 and 0.8. With STL10 dataset case, the following are the 6 filtered image datasets chosen within this range:

(1) Blur_1, (2) JPEG_30, (3) MedianFilter_3, (4) CONTOUR, (5) EMBOSS, (6) FIND_EDGES

Table I is showing the Mean-SCC and DNN accuracies for 6 distorted datasets chosen by the introduced selection criteria. The DNN accuracy achieved with the original image dataset was 0.959.

TABLE I. 6 Filtering Techniques and Parameters Chosen for Federated Learning

	(1) Blur_1	(2) CONTOUR	(3) EMBOSS	(4) FIND_EDGES	(5) JPEG_30	(6) MedianFilter_3
Mean-SCC	0.69	0.75	0.68	0.76	0.51	0.53
DNN Accuarcy	0.92	0.9	0.91	0.9	0.9	0.91

We carried out the 2-cloud federated learning on 9 combinations of the DNNs trained with these 6 distorted datasets. The results are given in Table II.

TABLE II. 9 Example Combinations for 1-Level 2-Cloud Federated Learning

	(1)+(4)	(1)+(5)	(1)+(6)	(2)+(4)	(2)+(5)	(2)+(6)	(3)+(4)	(3)+(5)	(3)+(6)
Federated DNN Accuracy	0.94	0.933	0.935	0.923	0.918	0.925	0.922	0.924	0.93
Approx. %	98 %	97.3 %	97.5 %	96.2 %	95.7 %	96.5 %	96.1 %	96.4 %	97%

Table III shows the DNN accuracies obtained from the 4-cloud federated learning.

TABLE III. 3 EXAMPLE COMBINATIONS FOR 2-LEVEL 4-CLOUD FEDERATED LEARNING

	(1)+(4), (3)+(6)	(1)+(5), (2)+(6)	(2)+(4), (3)+(5)
Federated DNN Accuracy	0.951	0.945	0.941
Approx. %	99.2%	98.5%	98.1%

It is shown that the approximation percentage in Table III is quite close to 100%. Especially, the combination of 4 different filtered datasets, (1) & (4) and (3) & (6), yields the best approximation of 99.2% to the accuracy of DNN from the original unfiltered dataset.

VII. CONCLUSION

A new mechanism to privacy preserving cloud based DNN training/classification is proposed in this paper by utilizing image filtering techniques. Application specific privacy requirements may be tailored and accommodated. A similarity metric has been identified that can be utilized in the selection process of the image filtering techniques and parameters without carrying out the actual DNN training.

To increase the applicability of our approach, we propose a federated cloud based approach to make use of multiple clouds with different distorted datasets, and showed that the comparable DNN accuracy can be achieved to that with the original dataset.

ACKNOWLEDGEMENTS

This work is supported by the U.S. ARO grant W911NF-17-1-0447.

REFERENCES

[1] Z. Wang, et. al.,"Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, pp. 600-612, April 2004.
[2] Z. Wang and A. Bovik, "A universal image quality index," IEEE Signal Processing Letters, vol. 9, pp. 81-84, March 2002.
[3] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in Proceedings for The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, Nov. 2003.
[4] Lucien Wald, "Quality of high resolution synthesized images: Is there a simple criterion?" in Proceedings for the third conference "Fusion of Earth data: merging point measurements, raster maps and remotely sensed images," Jan 2000, Sophia Antipolis, France. pp.99-103.
[5] J. Zhou , D. L. Civco, and J. A. Silander, "A wavelet transform method to merge Landsat TM and SPOT panchromatic data," International Journal of Remote Sensing, vol. 19, pp. 743-757, 1998.
[6] R. Yuhas, A. Goetz, and J. Boardman, "Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm," in Summaries of the Third Annual JPL Airborne Geoscience Workshop, vol. 1, pp. 147-149, June 1992.
[7] H. Sheikh and A. Bovik, "Image information and visual quality," IEEE Transactions on Image Processing, vol. 15, pp. 430-444, Feb. 2006.
[8] R. Sivagami, et. al., "Review of Image Fusion Techniques and Evaluation Metrics for Remote Sensing Applications," Indian Journal of Science and Technology, vol. 8, December 2015.
[9] J. Søgaard, et. al., "Applicability of Existing Objective Metrics of Perceptual Quality for Adaptive Video Streaming," Electronic Imaging. 2016 (13), pp. 1–72, 016.
[10] L. Zhang, X. Mou, and D. Zhang, "A comprehensive evaluation of full reference image quality assessment algorithms," 2012 19th IEEE International Conference on Image Processing. pp. 1477–1480, September 2012.
[11] Z. Wang and Q. Li, Qiang, "Information Content Weighting for Perceptual Image Quality Assessment". IEEE Transactions on Image Processing. 20 (5): 1185–1198. May 2011.
[12] S. Channappayya, A. Bovik, C. Caramanis, and R. Heath, "SSIM-optimal linear image restoration. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 765–768. March 2008.
[13] A. Gore and S. Gupta, "Full reference image quality metrics for JPEG compressed images". AEU - International Journal of Electronics and Communications, vol. 69, 2015.
[14] Z. Wang and E. Simoncelli, (September 2008). "Maximum differentiation (MAD) competition: a methodology for comparing computational models of perceptual quantities," Journal of Vision, vol. 8, pp. 1–13, September 2008.
[15] N. Dowlin, et. al., "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," Microsoft Technical report, MSR-TR-2016-3, February 2016.
[16] E. Hesamifard, H. Takabi, M. Ghasemi, and R. Wright, "Privacy preserving machine learning as a service," PoPETs, 2018(3):123–142, 2018.
[17] X. Jiang, M. Kim, K. Lauter, and Y. Song, "Secure outsourced matrix computation and application to neural networks," In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, 2018.
[18] 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, pp. 770-778. 2016.
[19] Torchvision.Models. https://pytorch.org/docs/stable/torchvision/models.html
[20] M.-R. Ra, R. Govindan, and A. Ortega, " P3: Toward privacy-preserving photo sharing," In NSDI, 2013.
[21] SEWAR Library, https://pypi.org/project/sewar/
[22] Pillow Library, https://pillow.readthedocs.io/en/stable/reference/ImageFilter.html
[23] R. Shokri and V. Shmatikov, ''Privacy-preserving deep learning,'' in Proc. 53rd Annu. Allerton Conf. Commun., 2016, pp. 1310–1321.
[24] X. Ma, F. Zhang, X. Chen, and J. Shen, ''Privacy preserving multi-party computation delegation for deep learning in cloud computing,'' Inf. Sci., vol. 459, pp. 103–116, Aug. 2018.
[25] T. Chen and S. Zhong, ''Privacy-preserving backpropagation neural network learning,'' IEEE Trans. Neural Netw., vol. 20, no. 10, pp. 1554–1564, Oct. 2009.
[26] A. Bansal, T. Chen, and S. Zhong, ''Privacy preserving back-propagation neural network learning over arbitrarily partitioned data,'' Neural Comput. Appl., vol. 20, no. 1, pp. 143–150, 2011.
[27] J. Yuan and S. Yu, ''Privacy preserving back-propagation neural network learning made practical with cloud computing,'' IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 212–224, Jan. 2014.
[28] J. J. W. Bos, K. K. Lauter, J. Loftus, and M. Naehrig, ''Improved security for a ring-based fully homomorphic encryption scheme,'' Cryptography and Coding, vol. 8308. New York, NY, USA: Springer, 2013, pp. 45–64.
[29] M. Hesamifard, E. Takabi, H. Ghasemi, E. Hesamifard, H. Takabi, and M. Ghasemi. (2017). ''CryptoDL: Deep neural networks over encrypted data.'' [Online]. Available: https://arxiv.org/abs/1711.05189
[30] Payman Mohassel and Yupeng Zhang, ''SecureML: A System for Scalable Privacy-Preserving Machine Learning,'' 2017 IEEE Symposium on Security and Privacy, May 2017.
[31] O. Kwabena, et. al., "MSCryptoNet: Multi-Scheme Privacy-Preserving Deep Learning in Cloud Computing," IEEE Access, vol. 7, pp. 29344-29354, February 2019.
[32] Google, "Google GPUs," 2019. [Online]. Available: https://cloud.google.com/gpu/
[33] Amazon, "AWS Deep Learning AMIs", 2019. [Online]. Available: https://aws.amazon.com/cn/machine-learning/amis/

A1: Measurements on similarity metrics and DNN accuracies on STL10 dataset

distortion	Mean_MSE	Mean_RMSE	Mean_RMSE_SW	Mean_PSNR	Mean_SSIM	Mean_UQI	Mean_MSSSIM	Mean_ERGAS	Mean_SCC	Mean_SAM	Mean_VIFP	Test_ACC
No Distortion	0.000	0.000	0.000	inf	1.000	1.000	1.000	0.000	0.982	0.000	1.000	0.959
Blur_1	189.798	13.257	11.348	26.051	0.886	0.983	0.974	8028.516	0.692	0.111	0.438	0.927
Blur_2	484.440	21.287	18.290	21.888	0.704	0.958	0.895	13845.617	0.199	0.179	0.255	0.865
Blur_3	709.484	25.809	22.299	20.197	0.574	0.938	0.809	17843.383	0.067	0.217	0.174	0.780
Blur_4	896.731	29.038	25.256	19.166	0.478	0.922	0.730	21325.940	0.023	0.244	0.130	0.708
CONTOUR_0	17947.183	131.479	124.568	5.932	0.336	0.566	0.286	259387.107	0.751	0.490	0.072	0.899
EMBOSS_0	4577.653	65.982	61.929	11.982	0.407	0.736	0.250	128759.519	0.682	0.476	0.041	0.911
FIND_EDGES_0	11480.023	103.746	95.984	8.101	0.280	0.355	0.341	41389.719	0.762	0.949	0.083	0.905
JPEG_1	724.083	26.589	24.887	19.746	0.598	0.913	0.836	16893.527	0.116	0.222	0.156	0.579
JPEG_3	666.128	25.499	23.898	20.110	0.629	0.917	0.854	16311.337	0.131	0.212	0.173	0.601
JPEG_5	452.009	20.947	19.649	21.842	0.712	0.940	0.904	16712.097	0.197	0.176	0.228	0.695
JPEG_7	346.951	18.306	17.082	23.036	0.767	0.955	0.929	11950.161	0.254	0.155	0.267	0.747
JPEG_10	268.712	16.065	14.940	24.195	0.812	0.964	0.948	10536.587	0.316	0.136	0.308	0.808
JPEG_15	204.624	13.978	12.950	25.432	0.849	0.966	0.963	9627.549	0.389	0.119	0.353	0.848
JPEG_20	170.518	12.735	11.751	26.259	0.874	0.975	0.971	8401.708	0.439	0.108	0.384	0.869
JPEG_30	131.996	11.178	10.267	27.414	0.900	0.979	0.979	7335.481	0.509	0.095	0.427	0.899
MedianFilter_3	138.346	11.104	9.234	27.781	0.909	0.989	0.982	5007.575	0.531	0.094	0.492	0.930
MedianFilter_5	309.844	16.779	13.917	24.099	0.798	0.975	0.940	7523.030	0.274	0.142	0.354	0.883
MedianFilter_7	453.697	20.408	16.886	22.347	0.712	0.964	0.892	9093.798	0.213	0.173	0.278	0.850
MedianFilter_9	584.734	23.238	19.197	21.189	0.640	0.954	0.844	10314.749	0.159	0.197	0.226	0.807
ModeFilter_3	48.917	5.061	2.586	36.877	0.986	0.994	0.998	2177.196	0.851	0.041	0.821	0.954
ModeFilter_5	342.651	17.046	13.157	24.184	0.857	0.965	0.963	8171.373	0.439	0.142	0.358	0.918
ModeFilter_7	961.054	29.657	24.440	19.088	0.631	0.913	0.859	13774.239	0.108	0.250	0.161	0.795
ModeFilter_9	1501.235	37.131	30.324	17.137	0.490	0.872	0.757	16559.630	0.013	0.314	0.101	0.697
Mosaic_1	189.963	13.219	11.322	26.109	0.885	0.984	0.978	8105.427	0.337	0.111	0.437	0.933
Mosaic_2	404.889	19.402	16.636	22.723	0.754	0.967	0.928	12282.113	-0.012	0.163	0.297	0.894
Mosaic_3	573.713	23.154	19.900	21.162	0.653	0.953	0.869	15217.515	0.008	0.195	0.219	0.859
Mosaic_4	719.462	25.963	22.391	20.155	0.569	0.941	0.810	17708.186	-0.006	0.219	0.168	0.830
P3_125	13756.495	115.694	106.297	6.993	0.099	0.603	0.050	191901.034	0.163	0.606	0.027	0.705
P3_150	14814.594	120.086	110.064	6.667	0.103	0.592	0.046	200684.317	0.146	0.615	0.028	0.710
P3_175	15362.310	122.291	112.010	6.508	0.107	0.584	0.045	207595.567	0.133	0.620	0.030	0.703
P3_200	15748.214	123.798	113.219	6.404	0.117	0.578	0.055	214468.294	0.138	0.631	0.033	0.697
P3_250	16580.616	126.860	115.233	6.205	0.128	0.568	0.057	228768.347	0.138	0.640	0.038	0.704
P3_300	17218.161	129.118	116.367	6.063	0.132	0.561	0.067	242218.579	0.113	0.664	0.042	0.703