CareNets: Efficient Homomorphic CNN for High Resolution Images

Chao Jin[1], **Ahmad Al Badawi**[1], **Balagopal Unnikrishnan**[1], **Jie Lin**[1], **Chan Fook Mun**[1], **James M. Brown**[2], **J. Peter Campbell**[3], **Michael Chiang**[3], **Jayashree Kalpathy-Cramer**[2], **Vijay Ramaseshan Chandrasekhar**[1], **Pavitra Krishnaswamy**[1*], **Khin Mi Mi Aung**[1*]

1: Institute for Infocomm Research, A*STAR, Singapore
2: Massachusetts General Hospital, Harvard Medical School, USA
3: Oregon Health & Sciences University, USA

{jin_chao, pavitrak, mi_mi_aung}@i2r.a-star.edu.sg

Abstract

Deep learning as a service paradigms are increasingly employed for image-based applications spanning surveillance, healthcare, biometrics, and e-commerce. Typically, trained convolutional neural networks (CNNs) are hosted on cloud infrastructure, and applied for inference on input images. There is interest in approaches to enhance data privacy and security in such settings. Fully homomorphic encryption (FHE) can address this need as it caters to computations on encrypted data, but poses intensive computational burden. Prior works have proposed approaches to alleviate this burden for 32×32 images, but practical applications require at least 10X higher resolution. Here, we present CareNets: Compact and Resource Efficient CNN for homomorphic inference on encrypted high-resolution images. Our approach is based on a novel compact packing scheme that packs CNN inputs, weights and activations densely into HE ciphertexts; and integrates them into the CNN computation flow. We implement CareNets using a GPU-accelerated FHE library for CNN inference on encrypted retinal images of size 96×96 and 256×256. Our results show that CareNets achieves over $32.78\times$ speedup, $45\times$ improvement in memory efficiency, and $5851\times$ reduction in transferred message size while maintaining accuracy within 3% of the non-encrypted CNN baselines.

1 Introduction

Deep learning has enabled significant performance leaps for a variety of image-based applications ranging from face recognition [18] and biometric identification [17] to surveillance image analysis, medical image classification [7], and QR code identification [8] for economic transactions. Increasingly, trained models are deployed in a Deep Learning as a Service paradigm and hosted on cloud infrastructure [11]. Typically, a user would upload input images to the cloud, wherein a previously trained model is evaluated on the input images. The prediction results are then returned to the user in exchange for a service fee. Although the cloud offers low cost yet scalable deployment, it poses important data privacy concerns, particularly for images with sensitive information. Hence, there is interest in secure deep learning inference on the cloud for image-based use cases.

Three approaches exist to address this need: 1) secure multi-party computation (MPC), 2) differential privacy (DP) and 3) fully homomorphic encryption (FHE). Each has its challenges. MPC is limited

*Equal Contributors.

PriML'19 - NeurIPS Workshop on Privacy in Machine Learning, Vancouver, Canada.

by communication complexity and bandwidth requirements across multiple parties [14, 13, 16]. DP adds noise to the data and may result in loss of prediction accuracy at inference time [1]. FHE evaluates functions on encrypted data and hence requires enormous amount of computation. We here consider FHE for deep learning inference and focus on improving efficiency without compromising security.

FHE has been extensively applied for simple computations on images such as correlations and histogram analysis [19]. Recent works have applied FHE for inference with convolutional neural networks (CNNs) on image datasets such as MNIST [6, 4, 12, 2] and CIFAR-10 [10, 4, 2], and demonstrated feasibility for higher complexity computations. However, these studies have been limited to low resolution (32×32) images. In contrast, most of the image-based applications above require at least 10X higher resolution (96×96 to 512×512) images [7].

One recent report explored FHE for CNN inference with 224×224 images [5], but requires 55 minutes for each image, imposes over 1 TB data communication burden on the user, and imposes significant RAM load on the cloud server. Practical scenarios might require latencies of a few minutes, MB-level data communication for users, and GB-level memory efficiency for inference on cloud servers.

To address the above requirements, we propose CareNets, a homomorphic CNN architecture based on a novel compact packing scheme that packs CNN inputs, weights, and activations densely into HE ciphertexts. Moreover, CareNets is very flexible as it can be adapted to perform computations on matrices and vectors with arbitrary sizes. We implement CareNets for CNN inference on encrypted retinal images on top of a GPU-accelerated FHE library, and demonstrate substantial performance gains while maintaining accuracy and security.

Our major contributions are as follows:

- We propose a new compact homomorphic CNN architecture that packs high-dimensional vectors of CNN inputs, weights and activations densely into FHE-encrypted ciphertexts, and apply highly parallel execution for homomorphic CNN operations.
- We design and develop FHE-friendly CNN models, and show their applicability for image classification problems involving real-world high resolution retinal image datasets.
- We provide the first GPU implementation for homomorphic CNN on high resolution images, and demonstrate significant improvements over several state-of-the-art baselines. Specifically, we achieve best-in-class inference latency, reduce client communication overhead by $5851\times$, improve memory efficiency by up to $45\times$, all while maintaining accuracy within 3% of the non-encrypted CNN baselines.

2 CareNets

2.1 Packing Strategy

In FHE, packing is used to compact an array of plaintext messages into into a ciphertext, to reduce complexity, parallelize homomorphic computations and accelerate performance. Prior works such as CryptoNets used *interleaved* packing, which is a batch processing method to perform CNN inference concurrently on as many images as the number of slots in each ciphertext. While interleaved packing can provide high throughput on the inference service, it also suffers from high resource usage, since it creates the same amount of ciphertexts even when predicting a single image. The problem becomes more severe when dealing with high-resolution images and wider CNNs, due to the creation of a large number of ciphertexts that can quickly exhaust system memory resources. Instead, CareNets is built on a *compact* packing approach to pack all inputs and outputs from each CNN layer into ciphertext slots. Specifically, compact packing packs pixels of a single image one by one into the same ciphertext. CareNets does not restrict the number of ciphertexts used during the compact packing process. When the slots in one ciphertext are fully used, it will continue with a new ciphertext.

In CareNets, the intermediate output in each layer is also converted into the compact packing format, which is subsequently fed into the next layer for computation. Compared with interleaved packing, the number of ciphertexts created in compact packing for the input image as well as the layer-wise intermediate data is dramatically reduced, resulting in dramatically lower memory demand. Moreover, we detail how the compact-packed data can be integrated into the CNN computations, and demonstrate that the computational latency for an entire run of the CNN is also reduced, due to the minimized number of ciphertexts and associated addition and multiplication operations.

2.2 HCNN Computation

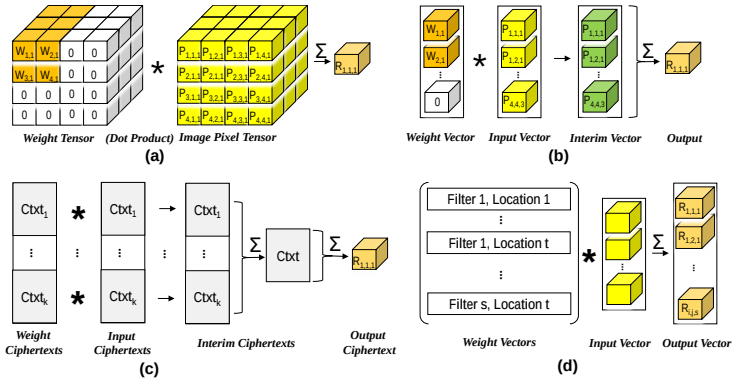

Figure 1: (a) An extended representation of one convolution computation operation. The filter is extended into the same dimension as the input tensor by padding zeros. (b) The extended filter tensor and the input tensor are flattened into weight and input vectors respectively, where the convolution (dot product) is done through element-wise multiplication followed by summing all the elements in the resultant vector. (c) The weight and input vectors are mapped into ciphertexts through compact packing. The convolution is evaluated through slot-wise multiplication between each pair of weight ciphertext and input ciphertext, and then accumulating all the resultant ciphertexts through slot-wise addition, and finally adding up all the slots inside this summed ciphertext. (d) The entire convolution layer is evaluated by permuting all the filters and shifted locations of each filter, doing convolution with the inputs, and arranging all the convolution results contiguously in the final output ciphertexts to form the compact-packed outputs.

Figure 1 illustrates how CareNets computes a convolution layer. A key idea in the design is to pad the filters into the same dimension as the input data, so every convolution step will always be computing the dot product between the extended filter and input data. It operates directly on compact-packed data, and does not require any sophisticated rearrangement of data in the ciphertexts. It also poses no restriction on the size and dimension of the filters and input data.

Complexity Analysis. CareNets computations mainly focus on dot products between extended filters and the layer-wise input. Suppose an extended filter and the layer input are split into k and l ciphertexts respectively (note k is not necessarily equal to l as all-zero ciphertexts in the extended filter will not be created). First, we compute slot-wise multiplication through k ciphertext multiplications. Second, the k resultant ciphertexts from the first step are added together through $k-1$ additions. Third, we need to add together all the slots in the resultant ciphertext from the second step. Here, we adopt the fast slot-summation algorithm [9] which requires $log_2 n$ rotations and additions respectively on the ciphertext, given n to be the total number of slots in one ciphertext[2]. This process will produce a ciphertext with the sum result filled in each of the slots. Lastly, we need to arrange the dot product result into the final output ciphertext. This is achieved by multiplying a mask with the resultant ciphertext from step three, and add it into the final output ciphertext. The mask is actually a one-hot vector containing all zeros expect a one at the target slot index. This step requires a multiplication with plaintext and addition.

Parallel Processing. The architecture of CareNets is easily parallelized. In Figure 1(c), the multiple pairs of weight-input multiplications can be processed in parallel. In Figure 1(d), each dot product of one weight-matrix row with the input-data vector can be computed independently, thus all the dot-product computations can also be processed in parallel.

3 Experiments and Evaluations

Data: As retinal images are relevant for both biometric and healthcare applications, we demonstrate the proposed homomorphic evaluation packing strategy on two retinal image datasets. 1) **Retinopathy of Prematurity (ROP)**: We obtained 1000 posterior pole retinal RGB photographs from the Imaging

[2]For BFV scheme, slots are organized into two groups of $n/2$. We can operate on the two groups concurrently.

and Informatics in ROP (i-ROP) study [3]. We preprocessed the images and converted to grayscale, segmented the retinal vessels [3], and downsampled to 96×96. 2) **Diabetic Retinopathy (DR)**: We obtained 249 color retinal fundus images from the IDRiD challenge dataset collected at an eye clinic located in India [15]. We resized the original RGB images to 256×256 and normalized them by the maximum intensity value in each image. In both cases, images were clinically assessed for abnormal features and denoted as healthy or diseased; and the task focuses on binary image classification.

Implementation: We implement a 5-layer FHE-compatible CNN model in CareNets. First, a convolution layer with 25 filters of size 5×5 and stride 2×2. Second, a square activation layer which applies square function on the output of the first layer. Third, another convolution layer with 50 filters of size 5×5 and stride 2×2. Fourth, another square activation layer. Fifth and finally, a fully-connected layer, weighted sum that generates 2 outputs (corresponding to 0 or 1 for the abnormality detection) from the entire outputs of the previous layer.

We use the same network architecture for both ROP data and DR data, except that DR data requires a wider network due to higher dimensionality. The FHE-compatible CNN achieves good prediction accuracy for these datasets, within 3% of the accuracy provided by a state-of-the-art Inception or CIFAR-Net network on the non-encrypted data. Since the homomorphic CNN is evaluated in memory sequentially one layer at a time, the dominant factor driving memory resource overhead is network width not depth. Hence, the 5-layer network is representative enough to demonstrate the superiority of CareNets in memory savings.

We implement CareNets on both multi-core CPU and GPU server platforms. The CPU server has two Intel Xeon processors each with 26 cores at 2.10 GHz frequency, and has 187.5 GB memory. The GPU server has a NVIDIA Tesla V100 GPU card, which has 16 GB GPU memory.

Table 1: Performance comparison of CareNets against traditional homomorphic CNN with interleaved packing. Units of Latency, Memory and Message Size are Second, GB and MB, respectively. FHE Scheme is BFV, and ring dimension $n = 2^{14}$. Security level $\lambda = 80$ bits.

Dataset	Architecture	Platform	$\log_2 q$	t	Latency	Memory	Message size
ROP	HCNN-Interleaved Packing	CPU	450	4503599627763713	3946.4	135	16200
$(96 \times 96 \times 1)$	CareNets	CPU	660	4503599627763713	994.9	**2.94**	**2.90**
	CareNets	GPU	660	2251799814045697	**120.4**	3.5	**2.90**
DR	HCNN-Interleaved Packing	CPU	450	4503599627763713	Failed	Failed	345600
$(256 \times 256 \times 3)$	CareNets	CPU	660	4503599627763713	6004.7	17.2	61.88
	CareNets	GPU	630	281474977595393	**1667.1**	**15.2**	**59.06**

Performance Evaluation: Table 1 provides performance comparisons of our method against the traditional homomorphic CNN with interleaved packing. We were unable to run GPU experiments for interleaved packing due to high memory requirements.

For the ROP dataset, CareNets enables a 32.78× improvement in runtime, 45.9× improvement in memory, and 5586.2× reduction in message size. These results are reasonable since an ROP image is $96 \times 96 \times 1$ and requires a large number of ciphertexts (one per pixel) to run traditional homomorphic CNN with interleaved packing. For the more challenging DR dataset with $256 \times 256 \times 3$ pixel images, CareNets enables an evaluation time of less than 28 minutes using 15.2 GB memory, requiring less than 60 MB message size on GPU. In contrast, the memory resources (187.5 GB) on our CPU server were insufficient to run the interleaved packing baseline. These results suggest that CareNets offers larger gains (over interleaved packing baselines) with higher resolution images.

4 Conclusion

Many practical applications require the ability to securely compute on high resolution encrypted images, in the absence of a decryption key. FHE provides security guarantees alongside the ability to compute, but is bottlenecked by high computation overhead. We have described a novel resource efficient FHE strategy that leverages a compact packing scheme to enable homomorhpic CNN inference for high resolution images. Evaluations on both CPU and GPU platforms using two retinal image datasets show that our approach enables significant improvements for memory-efficient low-latency FHE inference without compromising accuracy.

Acknowledgments

This project was supported by funding from the Deep Learning 2.0 program at the Institute for Infocomm Research (I2R), A*STAR, Singapore; research grants from the US National Institutes of Health (NIH grants R01EY19474, P30EY010572, and K12EY027720) and the US National Science Foundation (NSF grants SCH-1622679 and SCH-1622542); unrestricted departmental funding from the Oregon Health Sciences University, and a Career Development Award from Research to Prevent Blindness (New York, NY).

References

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

[2] Ahmad Al Badawi, Jin Chao, Jie Lin, Chan Fook Mun, Sim Jun Jie, Benjamin Hong Meng Tan, Xiao Nan, Khin Mi Mi Aung, and Vijay Ramaseshan Chandrasekhar. The alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. *arXiv preprint arXiv:1811.00778*, 2018.

[3] James M Brown, J Peter Campbell, Andrew Beers, Ken Chang, Susan Ostmo, RV Paul Chan, Jennifer Dy, Deniz Erdogmus, Stratis Ioannidis, Jayashree Kalpathy-Cramer, et al. Automated diagnosis of plus disease in retinopathy of prematurity using deep convolutional neural networks. *JAMA ophthalmology*, 2018.

[4] Alon Brutzkus, Ran Gilad-Bachrach, and Oren Elisha. Low latency privacy preserving inference. In *International Conference on Machine Learning*, pages 812–821, 2019.

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

[6] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. Technical report, Microsoft, February 2016.

[7] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[8] Tamás Grósz, Péter Bodnár, László Tóth, and László G. Nyúl. Qr code localization using deep neural networks. *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2014.

[9] Shai Halevi and Victor Shoup. Algorithms in helib. In *International Cryptology Conference*, pages 554–571. Springer, 2014.

[10] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*, 2017.

[11] Yashpalsinh Jadeja and Kirit Modi. Cloud computing-concepts, architecture and challenges. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, pages 877–880. IEEE, 2012.

[12] Xiaoqian Jiang, Miran Kim, Kristin Lauter, and Yongsoo Song. Secure outsourced matrix computation and application to neural networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1209–1222. ACM, 2018.

[13] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669. USENIX Association, 2018.

[14] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via MiniONN transformations. In *ACM CCS 17*, pages 619–631. ACM Press, 2017.

[15] P Porwal, S Pachade, R Kamble, M Kokare, G Deshmukh, V Sahasrabuddhe, and F Meriaudeau. Indian diabetic retinopathy image dataset (idrid). *IEEE Dataport*, 2018.

[16] Riazi Sadegh, Samragh Mohammad, Chen Hao, Laine Kim, Lauter Kristin, and Koushanfar Farinaz. XONN: Xnor-based oblivious deep neural network inference. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1501–1518. USENIX Association, 2019.

[17] Kalaivani Sundararajan and Damon Woodard. Deep learning for biometrics: A survey. *ACM Computing Surveys*, 51:1–34, 05 2018.

[18] Mei Wang and Weihong Deng. Deep face recognition: A survey. *ArXiv e-prints*, 2018.

[19] Lu Wen-Jie, Shohei Kawasaki, and Jun Sakuma. Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data. Cryptology ePrint Archive, Report 2016/1163, 2016. `https://eprint.iacr.org/2016/1163`.