A REPORT FOR ICLR PAPER REPRODUCTION: NETWORK DECONVOLUTION

Ziyi Guo, Zongze Li, Annan Wu

Department of Electronics and Computer Science, University of Southampton,
{zg2u21, zl18n21, aw1u21}@sonton.ac.uk

ABSTRACT

Convolutional Neural Network(CNNs) uses convolution kernels to scan across an image and captures key information. However, the strong correlations in real data force the convolution kernels to re-learn repeated information. This paper reproduces the exploration of *Network Deconvolution*, examines the reproducibility of the previous work, and clarifies the actual performance. Such Implementations are in *https://github.com/COMP6248-Reproducability-Challenge/Deconvlution-COMP6248-Reproducability-Challenge.*

1 INTRODUCTION

Images of nature captured by the human eyes or cameras with adjacent pixels are highly correlated (Hyvrinen et al., 2009), which is similar to the correlations illustrated by crossing an image with a Gaussian kernel. Containing redundant information, the convolution process of image with unknown filters can be challenging in training on account of the correlation effects. In animal brains, there has been proved to be visual correlation removal processes to reduce visual information redundancy (Hubel & Wiesel, 1961; 1962). Furthermore, arguments point out that natural brains take data compression as an essential step, involving removing redundant information and keeping the most salient features (Richert et al., 2016).

This paper reproduces the work of (Ye et al., 2019) at ICLR 2020, in which the ***network deconvolution*** is proposed as an image redundant correlation reduction method. Known that a correlated signal is generated by a convolution: $b = k * x = Kx$, where k is the kernel and K is the corresponding convolution matrix, the aim of network *deconvolution* is to remove correlation effects applying: $x = K^{-1}b$. The proposed method removes correlations from both single image with high pixel-wise correlation and hidden layers with channel-wise correlation.

The contributions in this paper are the following:

- Reproduce the generation and operation process of *network deconvolution* and analyze the reasonableness of related mathematical theories concerning matrix representation and kernel estimation optimality.
- Reproduce the effect of the implicit deconvolution and subsampling based *deconvolution* acceleration technique.
- Reproduce the redundancy reduction ability of *network deconvolution* in the real image data.
- Re-demonstrate the general improvement of performance applying *network deconvolution* on the CIFAR-10, CIFAR-100, Fashion-MNIST, and ImageNet datasets, based on 10 modern neural network models.
- Comprehensively analyze the reproducibility of each part of the proposed work and make possible adaptations and improvements to the current method.

2 LITERATURE REVIEW

The paper (Ye et al., 2019) firstly summarizes the limitations of current researches applying batch normalization methods on the ImageNet dataset and the shortage of being appropriate for the standard linear transform layers in fully-connected networks for most current techniques and therefore having not captured the nature of the convolution operation to deals with the pixels.

Illustrated as the initial motivation, (Ye et al., 2019) proposed the basic convergence problem of simplest linear regression problem and proved the one-iteration convergence condition of $\frac{1}{N}X^tX = I$ based on the optimal solution $w = (X^tX)^{-1}X^t\hat{y}$, which is applied in the paper as the baseline of coordinates transformation on the original data. Then, considerable work is carried out as pre-experiment from the perspectives of convolution layer representation properties, deconvolutional operation design, mathematics justification and visualization analysis, *deconvolution* process acceleration, *deconvolution* sparsification effect and the influence of parameter initialization. Finally, based on the proposed model, abundant work is carried out on the validation of model on various tasks on a wide range of datasets.

3 Pre-experiment reproduction and analysis

3.1 The Matrix Representation of Convolution Layer and Analysis

In the first part of the pre-experiment, the work of (Ye et al., 2019) demonstrating the process of convolution process as the foundation of deconvolution designing is reproduced and compared as below (Fig 1).

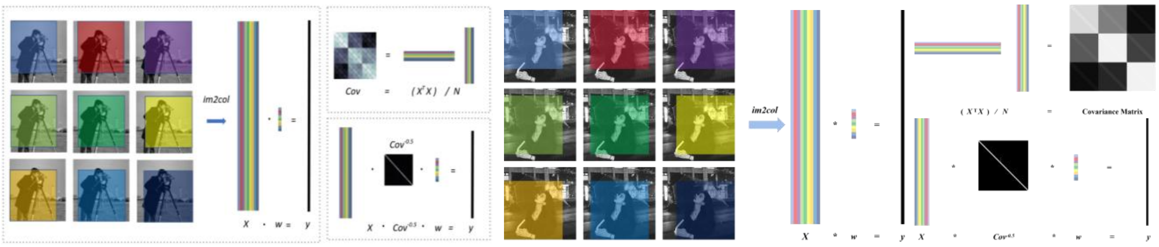

Figure 1: (Left) Matrix representation of convolution and visualization of **VGG** layer. (Right) Process reproduction.

The reproduction work confirmed the proposed high correlation caused by shifting a single pixel as well as the stronger pixel-wise correlation of natural images compared to cross-channel correlation, with the observation of a similarly brighter diagonal block in the figures from both proposed work and the corresponding reproduction.

3.2 Deconvolution Design and Kernel Visualization

In the second part, the deconvolution kernel is proposed as $Cov^{-0.5} * vec(\delta)$ based on the previous analysis, given by covariance matrix and a Vectorize function reshaping the kernel size, and visualized as below (Fig 2).

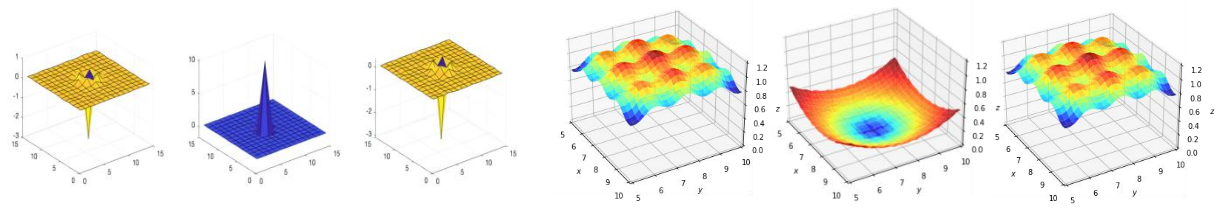

Figure 2: (Left) *Deconvolution* kernel visualization in images from ImageNet. (Right) Visualization reproduction.

Despite of a smaller random dataset and different visualization details, the reproduced visualization shows similar center-surround structures to the proposed work, which coincides with the *deconvolution* process in biological experiments (Hubel & Wiesel, 1961; 1962), and thus proving the rational nature of the algorithm to reduce data redundancy by focusing on respectively central positions. In turn, it proposes the potential optimality for the kernel estimation of *deconvolution*.

3.3 Deconvolution Accelerations

In the third part, observing the slower training speed applying *deconvolution* network, an acceleration method based on implicit modelling, fast computation and inverse square root of the covariance matrix is proposed (Fig 3).

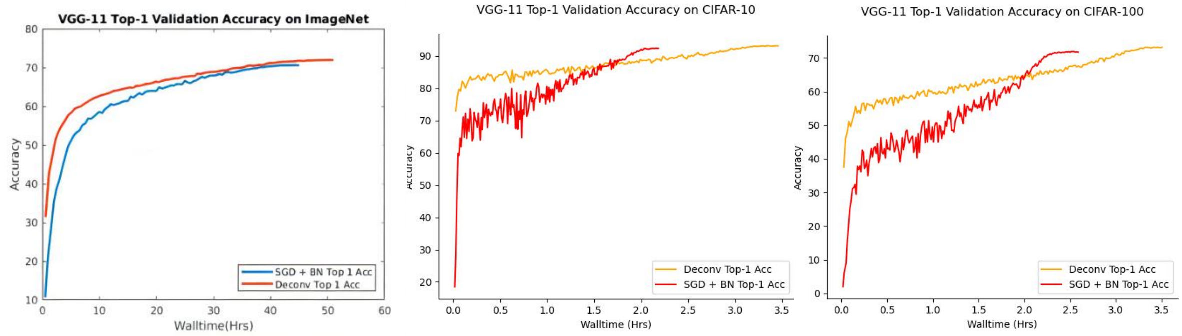

Figure 3: (Left) Loss Curves of VGG-11 Networks on ImageNet. (Right) Reproduction on CIFAR-10 and CIFAR-100.

According to the observation of the above work, the *network deconvolution* is significantly decaying faster than batch normalization as proposed. The proposed *FastDeconv* algorithm is effective on the improvement of the model efficiency and can be reproduced in other deconvolution-based models but probably works better on quite large dataset.

3.4 The Sparse Effect of Deconvolution

In the next part, experiments are carried out to valid the sparser representations produced by *network deconvolution* than the generally normalized image (Fig 4, Fig 5).

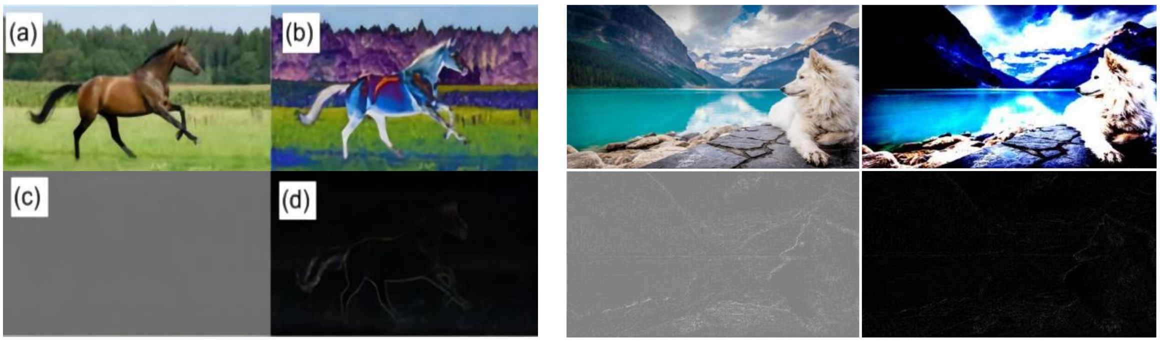

Figure 4: (Left) The input (a), the absolute-value zero-mean input (b), the deconvolved input (c), and the absolute-value deconvolved input (d). (Right) Reproduction of deconvolutional sparse effect on other specific image.

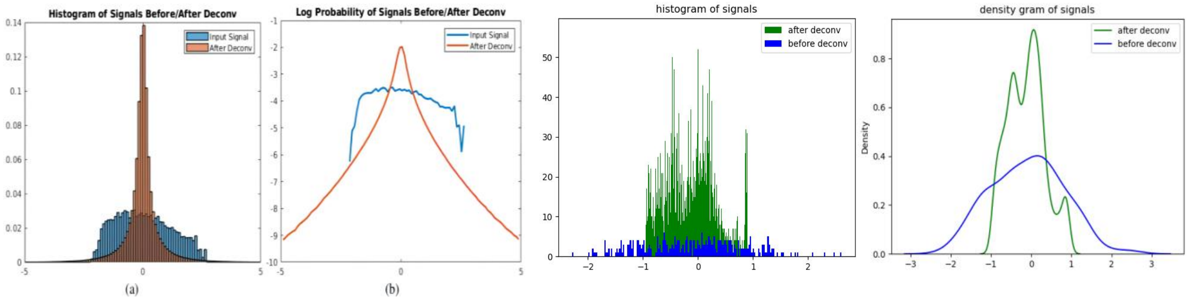

Figure 5: (Left) (a) Pixel values before/after *deconvolution*, and (b) the log pixel value density on ImageNet. (Right) Reproduction of deconvolutional sparse effect on random images.

Generally, despite of no available codes for pre-experiment architectures, this paper generates similar or even better reproduced results to the previous, which are not quite difficult to implement. The effect of *network deconvolution* is theoretically feasible and demonstrable, as the foundation of the following formal experiments.

4 Experiment reproduction and expansion

In the formal experiment, (Ye et al., 2019) carries out validations of network *deconvolution* performance on a range of networks and open datasets.

Firstly, (Ye et al., 2019) tests the *deconvolution* efficiency on simple linear regression task on the Fashion MNIST.

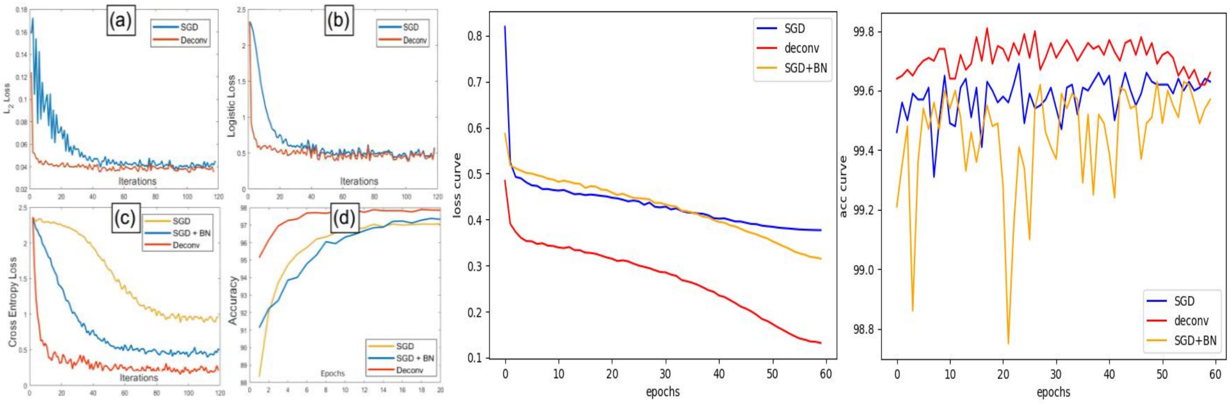

Figure 6: (Left) Training loss of Linear One-layer Model and test accuracy of Linear MLP. (Right) Reproduction figures.

This paper test the related performances with a self-created linear MLP model in the regression task on Fashion MNIST. The reproduction figures shows great similarity to the proposed work, indicating the capacity of *deconvolution* to improve both test accuracy and convergence speed on simple models without neural networks or simple networks.

Secondly, (Ye et al., 2019) tests *deconvolution* on CIFAR-10 and CIFAR-100 compared to batch normalization against epoch.

		CIFAR-10						CIFAR-100					
	Net Size	BN 1	ND 1	BN 20	ND 20	BN 100	ND 100	BN 1	ND 1	BN 20	ND 20	BN 100	ND 100
VGG-16	14.71M	14.12%	**74.18%**	90.07%	**93.25%**	93.58%	**94.56%**	2.01%	**37.94%**	63.22%	**71.97%**	72.75%	**75.32%**
ResNet-18	11.17M	56.25%	**72.89%**	92.64%	**94.07%**	94.87%	**95.40%**	16.16%	**35.73%**	72.67%	**76.55%**	77.70%	**78.63%**
Preact-18	11.17M	55.15%	**72.70%**	91.93%	**94.10%**	94.37%	**95.44%**	15.17%	**36.52%**	70.79%	**76.04%**	76.14%	**79.14%**
DenseNet-121	6.88M	59.56%	**76.63%**	93.25%	**94.89%**	94.71%	**95.88%**	17.90%	**42.91%**	74.79%	**77.63%**	77.99%	**80.69%**
ResNext-29	4.76M	52.14%	**69.22%**	93.12%	**94.05%**	95.15%	**95.80%**	17.98%	**30.93%**	74.26%	**77.35%**	78.60%	**80.34%**
MobileNet v2	2.28M	54.29%	**65.40%**	89.86%	**92.52%**	90.51%	**94.35%**	15.88%	**29.01%**	66.31%	**72.33%**	67.52%	**74.90%**
DPN-92	34.18M	34.00%	**53.20%**	92.87%	**93.74%**	95.14%	**95.82%**	8.84%	**21.89%**	74.87%	**76.12%**	78.87%	**80.38%**
PNASNetA	0.13M	21.81%	**64.19%**	75.85%	**81.97%**	81.22%	**84.45%**	10.49%	**36.52%**	44.60%	**55.65%**	54.52%	**59.44%**
SENet-18	11.26M	57.63%	**67.21%**	92.37%	**94.11%**	94.57%	**95.38%**	16.60%	**32.22%**	71.10%	**75.79%**	76.41%	**78.63%**
EfficientNet	2.91M	35.40%	**55.67%**	84.21%	**86.78%**	86.07%	**88.42%**	19.03%	**22.40%**	57.23%	**57.59%**	59.09%	**62.37%**

Table 1: Comparison on CIFAR-10/100 over 10 modern CNN architectures.

With available codes from (Ye et al., 2019), this paper reproduces almost all the experiments in the previous work, except for validations with DPN-92 (Netsize: 34.18M) limited by available GPU resources. The results are as below.

	CIFAR-10						CIFAR-100					
	BN 1	ND 1	BN 20	ND 20	BN 100	ND 100	BN 1	ND 1	BN 20	ND 20	BN 100	ND 100
VGG-16	13.61%	**50.10%**	88.02%	**89.48%**	91.08%	**94.10%**	1.49%	**31.01%**	60.31%	**68.09%**	73.14%	**75.00%**
ResNet-18	37.22%	**58.00%**	84.69%	**86.26%**	87.81%	**88.83%**	16.11%	**24.97%**	57.09%	**59.24%**	62.45%	**63.29%**
Preact-18	11.40%	**72.18%**	89.85%	**94.00%**	94.04%	**95.38%**	12.42%	**35.96%**	71.23%	**76.12%**	75.74%	**78.68%**
DenseNet-121	55.15%	**76.80%**	92.87%	**94.84%**	94.42%	**96.00%**	17.77%	**45.23%**	73.99%	**78.58%**	77.61%	**82.17%**
ResNext-29	42.14%	**69.18%**	92.11%	**94.03%**	94.98%	**95.83%**	17.78%	**30.74%**	74.38%	**78.42%**	78.29%	**81.89%**
MobileNet v2	62.29%	**72.10%**	83.87%	**93.26%**	83.75%	**85.92%**	27.09%	**35.41%**	64.90%	**74.02%**	68.75%	**76.36%**
PNASNetA	9.93%	**66.11%**	69.18&	**78.86%**	80.30%	**85.04%**	7.29%	**30.45%**	47.82%	**50.85%**	56.06%	**63.28%**
SENet-18	51.20%	**66.05%**	91.09%	**94,89%**	94.25%	**96.88%**	18.77%	**31.63%**	73.22%	**75.54%**	76.90%	**79.92%**
EfficientNet	38.87%	**45.98%**	84.39%	**87.81%**	83.74%	**86.95%**	12.44%	**23.72%**	54.09%	**57.78%**	61.20%	**62.55%**

Table 2: Reproduction of Table 1.

According to the comparison of the results, it can be observed that there is no significant gap between the proposed and the reproduction work; some experiments even acquires higher scores than the previous work, showing the great efficiency of *deconvolution* compared to batch normalization. However, several complex models take significantly great time to train, for example, the *DenseNet-121* model applying *deconvolution* for 100 *epochs* takes over 11 hours to train.

	VGG-11			ResNet-18		DenseNet-121		VGG-11		ResNet-18		DenseNet-121	
	Simple	BN	Deconv	BN	Deconv	BN	Deconv	BN	Deconv	BN	Deconv	BN	Deconv
ImageNet top-1	69.02%	70.38%	**71.95%**	69.76%	**71.24%**	74.65%	**75.73%**	68.83%	74.59%	62.67%	**72.42%**	70.56%	**73.37%**
ImageNet top-5	88.63%	89.81%	**90.49%**	89.08%	**90.14%**	92.17%	**92.75%**	84.18%	90.23%	81.80%	**91.41%**	91.71%	**93.80%**

Table 3: (Left)VGG-11, ResNet-18, DenseNet-121 on ImagNet and the reproduction(right).

Moreover, this paper reproduces the validation on ImageNet, but reduces the data volume and the classification number to an amount of 1000 and sets the batch size to 128 due to the limited GPU resource. The results above show *network deconvolution* remain the better performance than batch normalization in such challenging task, especially in the top-5 accuracy.

In addition, this paper explores the influence on *deconvolution* of various parameter settings, the results are as below.

Batch Size	*Accuracy*	*Learning Rate*	ε	*Epoch*
2	**87.91%**	0.001	0.01	2
8	**92.34%**	0.01	0.01	2
32	**91.89%**	0.01	1e-5	5
128	**92.55%**	0.1	1e-5	5

Table 4: Different performances with various parameter settings.

5 Conclusion

This paper reproduces the work of (Ye et al., 2019), and proves the outstanding performance of *network deconvolution*, indicating the good reproducibility of the proposed work as well. Specifically, this paper reproduces and visualized the biological nature, the mathematical optimality and the possible acceleration of network deconvolution entirely based on new programming architectures in the pre-experiment, showing the broad generality of proposed *deconvolution* theories. Next and more importantly, abundant experiments are carried out on the actual performances of *deconvolution* on a wide range of models on common tasks, where the baseline model of this paper as well as the reproduction of previous available codes both show conclusions of consistency with the proposed work, proving the general applicability of *deconvolution* and the portability of the previous work.

In conclusion, *network deconvolution* is an effective decorrelation method for convolutions beyond standard batch normalization and is largely reproducible and straightforward in terms of implementation.

References

Aapo Hyvrinen, Jarmo Hurri, and Patrick O. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 1848824904, 9781848824904

D. H. Hubel and T. N. Wiesel. Integrative action in the cat's lateral geniculate body. *The Journal of Physiology*, 155(2):385–398, 1961. doi: 10.1113/jphysiol. 1961.sp006635. URL https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1961.sp006635.

Micah Richert, Dimitry Fisher, Filip Piekniewski, Eugene M Izhikevich, and Todd L Hylton. Fundamental principles of cortical computation: unsupervised learning with prediction, compression and feedback. *arXiv preprint arXiv:1608.06277, 2016.*

Ye, C., Evanusa, M., He, H., Mitrokhin, A., Goldstein, T., Yorke, J. A., ... & Aloimonos, Y. (2019). Network deconvolution. *arXiv preprint arXiv:1905.11926*.