SUPER-RESOLUTION OF LARGE VOLUMES OF SENTINEL-2 IMAGES WITH HIGH PERFORMANCE DISTRIBUTED DEEP LEARNING

Run Zhang[1,2]*, Gabriele Cavallaro*[2] *and Jenia Jitsev*[2]

[1] RWTH Aachen University, Germany
[2] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany

ABSTRACT

This work proposes a novel distributed deep learning model for Remote Sensing (RS) images super-resolution. High Performance Computing (HPC) systems with GPUs are used to accelerate the learning of the unknown low to high resolution mapping from large volumes of Sentinel-2 data. The proposed deep learning model is based on self-attention mechanism and residual learning. The results demonstrate that state-of-the-art performance can be achieved by keeping the size of the model relatively small. Synchronous data parallelism is applied to scale up the training process without severe performance loss. Distributed training is thus shown to speed up learning substantially while keeping performance intact.

Index Terms— Sentinel-2, super-resolution, distributed deep learning, high performance computing

1. INTRODUCTION

With the development of aviation technologies and growing industrial demands, RS has become an increasingly popular field in the modern society. One important challenge of RS is to acquire high-quality images from sensors mounted on satellites. The spatial resolution is an important indicator of data quality since it determines the distance between two consecutive pixel centers measured on the ground (i.e., Ground Sampling Distance (GSD)). Images with higher spatial resolution represent more detailed information of the earth surface. However, due to the limitation of sensor accuracy, satellite orbital altitudes, space-ground communication bandwidth, etc., many satellites can not meet the fast-growing spatial resolution requirements of new generation scientific and industrial applications. Therefore, it is necessary to develop novel post-correction methods that can enhance the spatial resolution of raw observations.

Super-resolution techniques have attracted much attention by which the low quality low resolution RS images are enhanced. They include non-learning [1] and learning -based [2] methods. However, the scalability of the proposed algorithms for large data processing is usually not guaranteed. Methods that are public available are usually implemented and evaluated either on relatively small datasets or on shared-memory systems. Operational RS data processing workflows are expected to include parallel algorithms that can scale with the increasing of earth's observation data resulting from the continuous proliferation and improvement of RS platforms. For instance, the two twin satellites of the Sentinel-2 can acquire around 23 TB/day of multispectral images [1]. On the one hand, the rapid increase of RS data availability, makes both the storage and processing of the data more difficult to handle within shared-memory systems. On the other hand, algorithms such as deep learning networks can take benefit from the availability of large amounts data. They can compute more better generalization performance than traditional machine learning methods when large amounts of training data are available [3]. Recently, Lanaras et al., [4] trained a super-resolution model with a large Sentinel-2 dataset, which is a collection of tiles randomly selected on the globe and evenly covering all climate zones.

This paper proposes a deep super-resolution model based on self-attention mechanism. It is based on a distributed algorithm that can scale-up the training and testing process on distribute memory computers. Furthermore, the impacts of scaling the batch size and learning rate on the model accuracy are studied. The model is trained with the same large RS image dataset that was used in [4], and the experiments are run on HPC systems that are installed in the Jülich Supercomputing Centre. The experimental results show state-of-the-art performance with a relatively small model size, and a significant speed-up of the training and prediction phases.

2. PROBLEM FORMULATION

Sentinel-2 is an earth observation mission, part of the European Space Agency's Copernicus program and provides multi-spectral optical observation over global terrestrial surfaces with a high revisit frequency [5]. The 13 spectral bands in a Sentinel-2 product can be divided to three sets according

Research leading to these results has in parts been carried out on the Human Brain Project PCP Pilot Systems at the Juelich Supercomputing Centre, which received co-funding from the European Union (Grant Agreement no. 604102).

[1] https://sentinels.copernicus.eu/web/sentinel/news/-/article/2018-sentinel-data-access-annual-report

to their GSD, 10m bands: $A = \{B2, B3, B4, B8\}$, 20m bands: $B = \{B5, B6, B7, B8a, B11, B12\}$, and 60m bands: $C = \{B1, B9\}$. $B10$ is excluded because of comparatively poor radiometric quality. Supposing the pixel resolution of a 10m band in A is $w \times h$, the pixel resolution of the corresponding 20m and 60m bands in B and C will be $\frac{w}{2} \times \frac{h}{2}$, $\frac{w}{6} \times \frac{h}{6}$ respectively. This paper tackles the problem of super resolving the spatial resolution of low-resolution bands in B and C to 10m GSD, and two models, $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$, are developed correspondingly. $\mathcal{S}_{2\times}$, super-resolves the 20m bands in B with bands from both A and B

$$\mathcal{S}_{2\times} : \mathbb{R}^{w\times h\times 4} \times \mathbb{R}^{\frac{w}{2}\times\frac{h}{2}\times 6} \mapsto \mathbb{R}^{w\times h\times 6}. \quad (1)$$

C is excluded because Lanaras et al., [4] have showed that it does not contribute to 20m $\mapsto$ 10m super-resolution by experiments. $\mathcal{S}_{6\times}$, super-resolves the 60m bands C with bands from A, B and C

$$\mathcal{S}_{6\times} : \mathbb{R}^{w\times h\times 4} \times \mathbb{R}^{\frac{w}{2}\times\frac{h}{2}\times 6} \times \mathbb{R}^{\frac{w}{6}\times\frac{h}{6}\times 2} \mapsto \mathbb{R}^{w\times h\times 2}. \quad (2)$$

3. PROPOSED METHOD

3.1. Network architecture

The network architecture of $\mathcal{S}_{2\times}$ is shown in Figure 1. Supposing the input of $\mathcal{S}_{2\times}$ is given by (a, b), where a and b are the set of 10m and 20m bands respectively, the first step is to up-sample the 20m bands b 2 times by bilinear interpolation $H_{2\uparrow}(b)$. Next, the up-sampled $H_{2\uparrow}(b)$ proceed to fuse with 10m bands a through a band fusion module:

$$F_{fusion_{2\times}} = H_{fusion_{2\times}}(a, H_{2\uparrow}(b)). \quad (3)$$

The Figure 2 (b) shows the mixture correlations of multiple spectral bands. Then, $F_{fusion_{2\times}}$ goes through a residual self-attention module (RSA) and a final convolution layer to learn the residual between the high-resolution reference and $H_{2\uparrow}(b)$:

$$F_{diff_{2\times}} = F_{Conv}(F_{RSA}(F_{fusion_{2\times}})). \quad (4)$$

Finally, the output of the generator $\mathcal{S}_{2\times}$ is computed:

$$F_{\mathcal{S}_{2\times}} = H_{2\uparrow}(b) + F_{diff_{2\times}}. \quad (5)$$

The RSA module is made of 6 residual blocks (see Figure 2 (c)) with a self-attention module (see Figure 2 (a)) in the middle. Usually, a convolution layer can't explore the global structures inside an image because of spatial limitations of receptive fields. The self-attention mechanism was proposed by Zhang et al., [6] to capture the long-range dependencies over entire input feature maps. 6 residual blocks are used to show fairness on model complexity with the comparable method Dsen2.

Similarly, the model $\mathcal{S}_{6\times}$ can be represent by Equation (6) and (7) when given the input (a, b, c), where a, b, c is the 10m, 20m, 60m bands respectively. The architecture of $\mathcal{S}_{6\times}$ is the same with $\mathcal{S}_{2\times}$ except the one more input branch in the entire network and band fusion module:

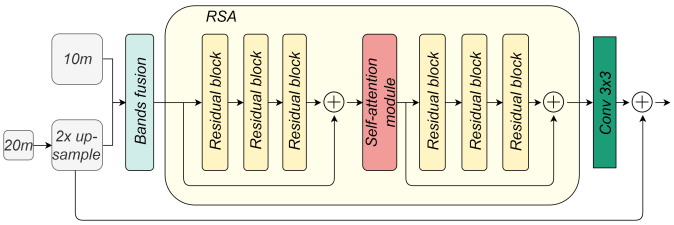


Fig. 1: Network architecture of model $\mathcal{S}_{2\times}$

$$F_{fusion_{6\times}} = H_{fusion_{6\times}}(a, H_{2\uparrow}(b), H_{6\uparrow}(c)), \quad (6)$$

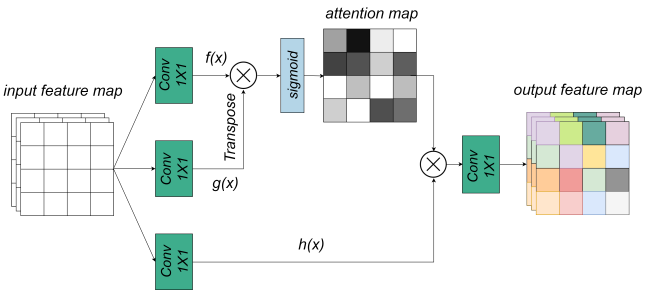
$$F_{\mathcal{S}_{6\times}} = H_{6\uparrow}(c) + F_{Conv}(F_{RSA}(F_{fusion_{6\times}})). \quad (7)$$

3.2. Distributed training

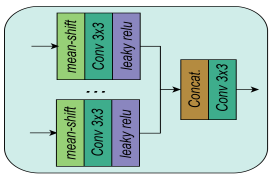
Jeff et al., [7] proposed two paradigms, *model parallelism* and *data parallelism*, to parallelize the training of a deep model. This paper applies *synchronized data parallelism* to scale-up the training of super-resolution model to HPC clusters. More specifically, for t th stochastic gradient descent (SGD) iteration of model θ, a mini-batch M is split to multiple partitions where each partition is consumed by its own work to calculate gradients. Finally, the gradients are accumulated from all works and update the model θ with the averaging gradient $\frac{\sum_i \nabla_\theta \mathcal{L}(\theta, x_i)}{|M|}$, where x_i is the i th instance in M and $\mathcal{L}$ is the loss function. Instead of setting up one or several parameter servers (built-in distribution strategy in Tensorflow), this paper used the library Horovod [8] to aggregate and average gradient over multiple workers, which relies on the *Ring Reduction Mechanism* and has been proven to be bandwidth-optimal, without system bottleneck [9].

To make distributed learning efficient, the per-worker workload must be large, which implies a corresponding growth in the SGD mini-batch size when increasing the number of workers. To improve the model convergence rate with the scaled mini-batch size, Priya et al., [10] proposed linear learning rate scale rule, that helps to train an object recognition model in an hour with large mini-batch sizes up to 8192 images on Imagenet. With the considered remote sensing dataset, it was found that this rule often causes a failure in the training (loss explodes). Therefore, the following modified linear learning rate scale rule is used:

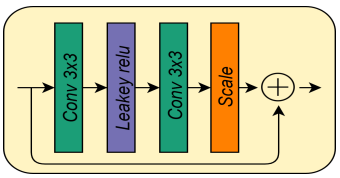
Modified linear learning rate scale rule: when the mini-batch size is multiplied by k, the initial learning rate is also multiplied by k and decays to half every $\frac{n}{k}$ SGD iterations.



(a) self-attention module



(b) band fusion module



(c) residual block

Fig. 2: Network modules in models $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$. (a) self-attention module, where $\otimes$ denotes matrix multiplication. (b) band fusion module, where the operation mean-shift moves the mean of input to 0 to suppress the impact of large brightness changes of training and testing patches. (c) residual block, scaling with 0.1 is used instead of batch normalization to speed up the training process. All convolution layers in this paper have 128 filters except $f(x)$ and $g(x)$ has 16 filters.

4. EXPERIMENTAL RESULTS

4.1. Experimental Setup

To train and evaluate $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$, the same Sentinel-2 tiles that were used in Dsen2 [4] are considered. They are free of charge and publicly available on the Copernicus services data hub[2]. Similarly to [4], the Wald's protocol [11] is adopted to generate the groudtruth, since the high-resolution references of the original Sentinel-2 tiles are not available. Furthermore, since the size of each Sentinel-2 tile is too large to fit into one GPU memory, small training and test patches are extracted. The model is implemented with Tensorflow 1.13, and updated by ADAM optimizer [12] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10e^{-8}$, and modified linear learning rate scale rule where $n = 64000$. The two model are trained in the JURON and JUWELS [13] HPC systems installed in Juelich supercomputing center. Each node in JURON is equipped with 4 Tesla P100 GPUs and each pair of GPUs are connected to one CPU socket via fast NVlink. The two GPUs in each pair are connected with NVlink. In JUWELS, each accelerated computing node is equipped with four Tesla V100 GPUs and the four GPUs are interconnected via NVLink in an all-to-all topology. Under this setting, the largest mini-batch size when training the model $\mathcal{S}_{2\times}$ (or $\mathcal{S}_{6\times}$) on a single GPU is 128 (or 32).

[2] https://scihub.copernicus.eu/

4.2. Evaluation

To have a comprehensive assessment, the metrics rooted mean square error (RMSE), signal-to-reconstruction error (SRE), spectral angle mapper (SAM), Erreur Relative Globale Adimensionnelle de Synthese (ERGARS) [14], structural similarity index (SSIM) and peak signal to noise ratio (PSNR) are considered for the evaluation of the model. The comparison methods include 1) naive Bicubic interpolation, tested with the library OpenCV. 2) DSen2 [4], tested with the model published in the repository[3].

Table 1 and Table 2 shows the synthetic performance of $\mathcal{S}_{2\times}$, $\mathcal{S}_{6\times}$ with scaled mini-batch size and scaled learning rate trained on JURON when super-resolving the degraded Sentinel-2 patches to original scale. When training with 4 GPUs in 24 hours, $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$ achieved better performance than DSen2, and when scaling up to 16 GPUs, $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$ can converge faster in 4 hours and have no severe performance loss. Figure 3 and 4 shows the per-second throughput of training patches when training $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$ on JURON and JUWELS with different number of GPUs. The training speed can not grow linearly, because when training with more GPUs, the communication cost to aggregate gradients on all GPUs also increases, and it also gets influenced by the mini-batch size and infrastructure architecture, reflected by the different data throughput when training $\mathcal{S}_{2\times}$ and $\mathcal{S}_{6\times}$ and the performance difference of JURONS and JUWELS.

Method	No. GPU	Batch size	Training time	RMSE	SRE	SAM	ERGAS	SSIM	PSNR
Bicubic	-	-	-	125.69	25.64	1.22	3.48	0.82	44.9998
DSen2	1	128	96h	35.85	35.94	0.78	1.07	0.9322	55.5416
Proposed	1	128	24h	36.42	35.83	0.78	1.08	0.9320	55.4393
Proposed	2	256	24h	35.67	36.03	0.77	1.06	0.9329	55.6199
Proposed	4	512	24h	**34.99**	**36.19**	**0.75**	**1.03**	**0.9336**	**55.7756**
Proposed	8	1028	12h	35.61	36.05	0.76	1.05	0.9329	55.6393
Proposed	16	2056	**4h**	38.58	35.27	0.81	1.16	0.9291	54.9243

Table 1: The synthetic performance of model $\mathcal{S}_{2\times}$ with scaled batch size and scaled learning rate. The learning rate of the experiment in each row is initialized with $0.0001\times$ No. GPUs

Method	No. GPU	Batch size	Training time	RMSE	SRE	SAM	ERGAS	SSIM	PSNR
Bicubic	-	-	-	161.85	19.79	1.78	7.30	0.36	37.6785
DSen2	1	128	96h	28.11	34.47	0.36	1.38	0.8953	52.4984
Proposed	1	32	24h	29.20	34.00	0.37	1.43	0.8917	51.9991
Proposed	2	64	24h	27.23	34.69	0.35	1.32	0.8959	52.7027
Proposed	4	128	24h	**26.80**	**34.98**	**0.34**	**1.29**	**0.8991**	**52.9451**
Proposed	8	256	12h	27.74	34.54	0.36	1.36	0.8959	52.5506
Proposed	16	512	**4h**	32.28	32.97	0.42	1.62	0.8828	50.9784

Table 2: The synthetic performance of model $\mathcal{S}_{6\times}$ with scaled batch size and scaled learning rate. The learning rate of the experiments in each row is initialized with $0.0001\times$ No. GPUs

[3] https://github.com/lanha/DSen2/tree/master/models

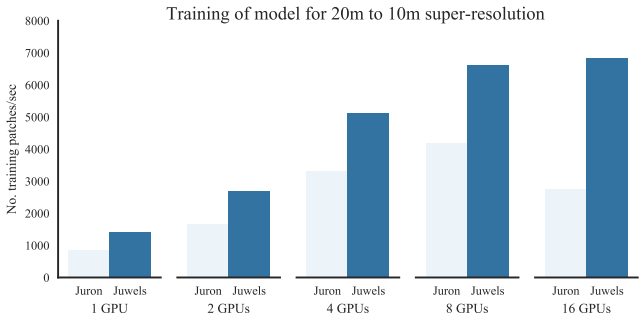


Fig. 3: Data throughput when training $\mathcal{S}_{2\times}$

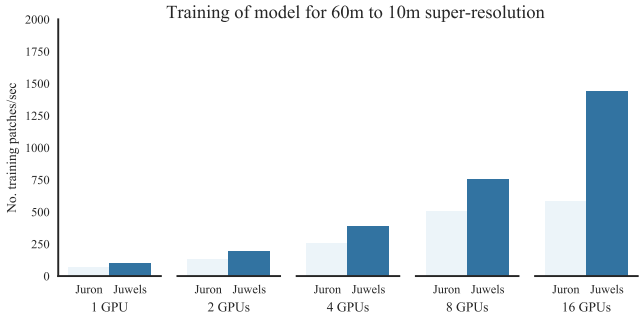


Fig. 4: Data throughput when training $\mathcal{S}_{6\times}$

5. CONCLUSIONS

This paper proposed a novel super-resolution deep learning model based on self-attention mechanism and distributing training via Horovod. The state-of-the-art performance for Sentinel-2 tiles super-resolution is achieved with a significant reduction of the training time from several days to several hours. With the publicly available code in the repository[4], community can train their own super-resolution models with significantly increased speed.

6. REFERENCES

[1] N. Brodu, "Super-Resolving Multiresolution Images with Band-Independent Geometry of Multispectral Pixels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4610–4617, 2017.

[2] J. M. Haut, R. Fernandez-Beltran, M. E. Paoletti, J. Plaza, and A. Plaza, "Remote Sensing Image Superresolution Using Deep Residual Channel Attention," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9277–9289, 2019.

[3] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019.

[4] C. Lanaras, J. Bioucas-Dias, S. Galliani, E. Baltsavias, and K. Schindler, "Super-Resolution of Sentinel-2 Images: Learning a Globally Applicable Deep Neural Network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 305–319, 2018.

[5] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort *et al.*, "Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services," *Remote sensing of Environment*, vol. 120, pp. 25–36, 2012.

[6] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv:1805.08318*, 2018.

[7] Y. Cao, G. W. Ding, K. Y.-C. Lui, and R. Huang, "Improving GAN Training via Binarized Representation Entropy (BRE) Regularization," *arXiv:1805.03644*, 2018.

[8] A. Sergeev and M. D. Balso, "Horovod: Fast and Easy Distributed Deep Learning in TensorFlow," *arXiv:1802.05799*, 2018.

[9] P. Patarasuk and X. Yuan, "Bandwidth Optimal All-Reduce Algorithms for Clusters of Workstations," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 117–124, 2009.

[10] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, Large Minibatch SGD: Training Imagenet in 1 Hour," *arXiv:1706.02677*, 2017.

[11] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of Satellite Images of Different Spatial Resolutions: Assessing the Quality of Resulting Images," *Photogrammetric Engineering and Remote Sensing*, vol. 63, pp. 691–699, 11 1997.

[12] "Adam: A Method for Stochastic Optimization, author=Kingma, Diederik P and Ba, Jimmy," *arXiv:1412.6980*, 2014.

[13] Jülich Supercomputing Centre, "JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre," *Journal of large-scale research facilities*, vol. 5, no. A135, 2019.

[14] L. Wald, *Data Fusion: Definitions and Architectures: Fusion of Images of Different Spatial Resolutions*. Presses des MINES, 2002.

[4] https://gitlab.version.fz-juelich.de/CST_DL/projects/remote_sensing/gan_superresolution