S^6: SEMI-SUPERVISED SELF-SUPERVISED SEMANTIC SEGMENTATION

Moamen Soliman, Charles Lehman and Ghassan AlRegib

OLIVES at the Center for Signal and Information Processing (CSIP)
School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta, GA, 30332-0250
{soliman, charlie.k.lehman, alregib}@gatech.edu

ABSTRACT

Semi-supervised learning provides a means to leverage unlabeled data when labels are expensive to obtain. In this work, we propose a constrained framework that better learns from unlabeled data. The proposed algorithm adds a self-supervised task, image reconstruction, to the target segmentation task. The extra reconstruction task improves the model's geometric reasoning about different textures in an image. It happens that this improvement is transferable from reconstruction to segmentation since they both share some common parts of the architecture. Such extra task allows the trained model to have richer representations and better geometric understanding. Our results show that the proposed constrained framework achieves an improvement in mean Intersection over Union by 18% over unconstrained one, using only 2% of labeled examples. The performance gain and reduction in amounts of labeled data is crucial for applications in which obtaining labels is expensive and labor intensive such as Biomedical Imaging and Seismic Interpretation.

Index Terms— Semantic Segmentation, Semi-supervised Learning, Self-supervised Learning, Unsupervised Learning, Constrained Representation, Image Reconstruction.

1. INTRODUCTION

Deep neural networks are known for their superior performance over traditional machine learning algorithms. In fact, in some applications deep neural networks can achieve super-human-level performance. However, such supremacy comes at the cost of requirement of huge amounts of labeled examples. In many domain-specific applications, obtaining sufficiently large amounts of labeled data can be expensive, time consuming and impractical. To address the problem of scarcity of data, semi-supervised learning (SSL) techniques are employed. In contrast to fully-supervised learning algorithms, which require all the examples to be labeled, SSL uses unlabeled examples to learn structure about the data, thus reducing the dependency on labeled examples.

The authors in [1] proposed a theoretical formalism for Probably Approximately Correct (PAC) framework of SSL. They also introduced the notion of compatibility function that can quantify how well learning models cluster from unlabeled examples. The SSL problem was framed as a combination of fully-supervised learning problem, where a learning model is trained to minimize a loss function using fully labeled examples, and an self-supervised/unsupervised learning problem, where a learning model is trained using a compatibility function to better cluster unlabeled examples. SSL techniques leverage the assumption that if two points are projected into the same dense neighborhoods then they are likely to also share a common label (smoothness assumption [2]). Thus, unlabeled data, when transformed to feature space manifold, can be matched to the nearest label in that space. This means that unlabeled data can act as a regularizer for deep networks, thus improving their generalization abilities. The deep SSL literature is divided into two schools. The first, adds unsupervised loss term called regularizer to the loss function used to train the deep network. This term is applied either to all the data points or only to the unlabeled ones [3, 4, 5, 6]. While the other, assigns pseudo-labels to the unlabeled examples and then treats them as labeled ones [7, 8].

In image processing and computer vision applications, there is often additional information available in the form of unlabeled data, which is often much cheaper and more plentiful than labeled data. This motivated us to revisit the self-supervised portion of SSL algorithms. Some self-supervised learning techniques can improve the deep network geometrical reasoning and understanding. Such techniques can play the role of the SSL compatibility function. An example of was proposed in [9], where the authors combined fully-supervised learning with self-supervised learning. Self-supervised learning is a framework of learning an alternative (pretext) task using unlabeled data. The surrogate task is designed in a way that solving it requires some geometrical understanding and representation of images. Examples of pretext tasks include: predict relative locations in randomly sampled non-overlapping image patches [10, 11, 12], colorize grayscale versions of the images [13], and learn a representation that is invariant to significant image modification for data augmentation [14, 15].

The main contribution of this work is the *study of image reconstruction as form of self-supervision for image segmentation as a main task.* Up to our knowledge, this study was not conducted before. The proposed model learns image segmentation in fully-supervised fashion and image reconstruction as a self-supervised auxiliary task. The system encoder-decoder architecture, shown in Figure 1(b), is constrained by the reconstruction branch to learn a representation that can perform segmentation as well as reconstruction. Our hypothesis is that *reconstruction branch constraints the deep network representation to be richer* since it segments and reconstructs the original image instead of segmenting only. Such arrangement hands over most of the learning from decoders to the encoder since the same encoder representation is now serving two different geometric tasks instead of one.

2. METHOD

2.1. Architectures

We use a popular encoder-decoder architecture for semantic segmentation, DeepLab V3+ [16], which utilizes atrous separable convolution to better condense the features and speed up computations. ResNet 18 is used as a backbone, while the decoder is kept simple by reducing the number of trainable parameters.

The unconstrained model shown in Figure 1(a) has one decoder branch for segmentation that is trained to minimize Softmax Cross Entropy Loss given by

$$L\left(\hat{Y}_i, Y_i\right) = -\sum_{j=1}^{c} Y_{ij} \times \log(p_{ij}), \tag{1}$$

where $\hat{Y}_i$ is the predicted label for data pixel number i, y_i is one hot encoded label vector, c is the number of classes, Y_{ij} is binary label indicating whether pixel i belongs to class j or not, and p_{ij} predicted probability that label i belongs to class j. For the constrained model in Figure 1(b), a modified version of DeepLab v3+ that uses two decoders was implemented. The two decoders: one for segmentation and trained using (1) only, another for simultaneous reconstruction of the input with the following loss function

$$L\left(\hat{X}_i, X_i\right) = ||\hat{X}_i - X_i||^2 \tag{2}$$

where, $\hat{X}_i$ is the reconstructed image and X_i is the original one. We use separate decoders for different tasks to allow model batch norm layers in each decoder to capture more specialized statistics for its specific task [17].

2.2. Data Splits

To test our hypothesis, we assume that the model can access all available training data, but only a fraction of the corresponding labels. Let X, Y denote the sets of all training data points and their corresponding labels, respectively. We uniformly sampled to split training data and labels into two sets denoted by X_n, Y_n and X_{1-n}, Y_{1-n}, respectively. Where $n \in [0, 1]$ denote the split of data and labels. The Sampling process is performed without replacement, so $X = X_n \cup X_{1-n}, Y = Y_n \cup Y_{1-n}$ and $X_n \cap X_{1-n} = \phi, Y_n \cap Y_{1-n} = \phi$. The unconstrained model in Figure 1(a) is trained using X_n, Y_n for prediction, while the constrained model in Figure 1(b) is trained using X for reconstruction and X_n, Y_n for segmentation. The aforementioned training procedure ensures that both models have access to the exact same labels, thus any improvement in performance for constrained model over unconstrained is to be attributed to the extra reconstruction branch.

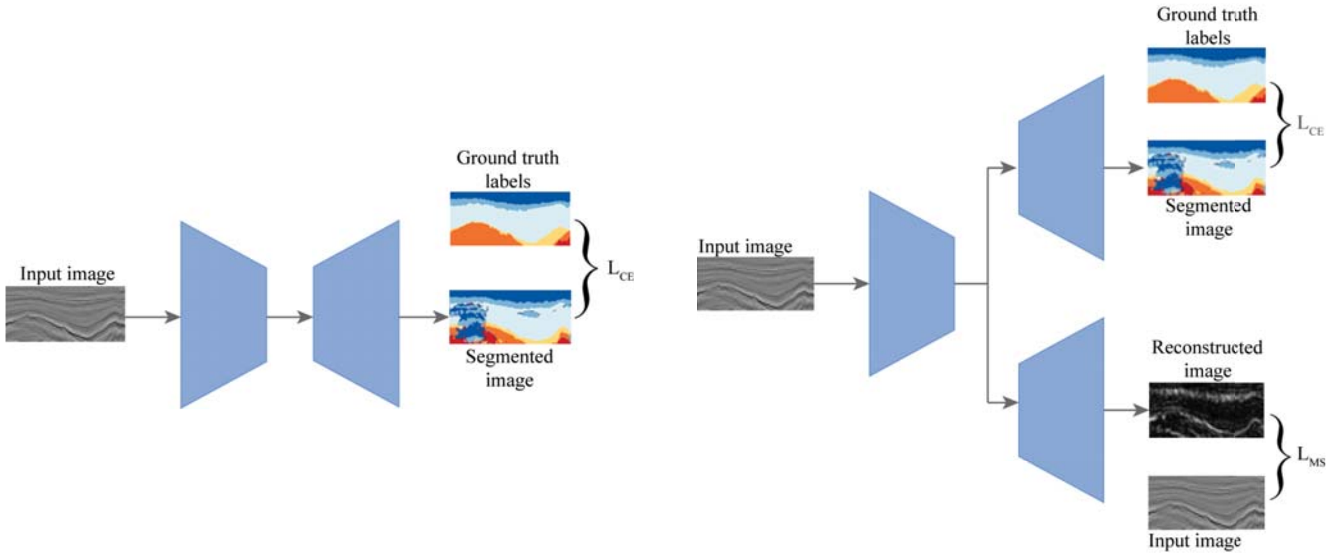
3. EXPERIMENTS

3.1. Dataset

To use our algorithm where it is most useful, we chose a dataset from a domain where obtaining fully labeled examples is expensive and time consuming. Both architectures are trained and tested using the Facies Classification Benchmark dataset [18]. This dataset is comprised of a collection of segmented seismic images of different rock layers and their corresponding geological facies (classes) as labeles. It is an essential part in the oil and gas exploration process to identify different facies (classes) of different rock layers. With six different lithostratigraphic classes, the F3 block facies from shallowest to deepest include: Upper, middle, lower North Sea, Rijnland/Chalk, Scruff and Zechistien classes. The intersection of Inlines from 300 to 700 with Crosslines from 300 to 1000, North West region of Figure 3, are used for training and validation as full 2D images. While the intersection of Inlines from 100 to 299 with Crosslines 300 to 1000, South West region of Figure 3, form test set 1 2D images. Finally, Inlines from 100 to 700 intersecting with Crosslines 1001 to 1200, East region of Figure 3, form test set 2 2D images.

3.2. Implementation Details

In semi-supervised training, Adam optimizer is used to train both networks from without pretraining. The learning rate is set to 5×10^{-4} and weight decay is set to 0. Training and testing are done using 2 RTX Titans with total GPU memory of 48 GBs. With a batch size of 50 it takes approximately



(a): Unconstrained Model

(b): Constrained Model

Fig. 1: Unconstrained model (a) is classical encoder-decoder architecture with only segmentation branch and is only trained using Cross Entropy loss. While the constrained model (b) has an extra decoder branch for reconstruction and is trained using Cross Entropy and Mean Squared Error losses for segmentation and reconstruction respectively.

n%	**Unconstrained**							**Constrained**						
	IoU	**U**	**M**	**L**	**R**	**S**	**Z**	**IoU**	**U**	**M**	**L**	**R**	**S**	**Z**
8	0.519 ± 0.06	0.982 ± 0.00	0.878 ± 0.02	0.964 ± 0.01	0.614 ± 0.31	0.209 ± 0.26	0.077 ± 0.15	**0.563** ± 0.05	0.964 ± 0.02	0.908 ± 0.02	0.926 ± 0.06	0.734 ± 0.11	0.364 ± 0.07	0.161 ± 0.18
6	0.509 ± 0.09	0.964 ± 0.01	0.909 ± 0.02	0.968 ± 0.00	0.606 ± 0.30	0.194 ± 0.25	0.087 ± 0.17	**0.624** ± 0.05	0.977 ± 0.01	0.853 ± 0.06	0.960 ± 0.02	0.734 ± 0.04	0.500 ± 0.14	0.319 ± 0.21
4	0.425 ± 0.10	0.986 ± 0.01	0.538 ± 0.44	0.770 ± 0.38	0.517 ± 0.29	0.441 ± 0.27	0.092 ± 0.18	**0.596** ± 0.07	0.902 ± 0.08	0.868 ± 0.03	0.930 ± 0.04	0.770 ± 0.02	0.548 ± 0.23	0.311 ± 0.10
2	0.414 ± 0.08	0.981 ± 0.02	0.687 ± 0.34	0.943 ± 0.02	0.339 ± 0.28	0.170 ± 0.21	0.008 ± 0.02	**0.608** ± 0.05	0.926 ± 0.05	0.880 ± 0.02	0.954 ± 0.01	0.703 ± 0.08	0.613 ± 0.08	0.203 ± 0.18

Table 1: Ablation study comparing constrained versus unconstrained models in terms of IoU and class accuracies at different percentages of labeled training data. Columns are labeled using class name initials which are: **U**pper North Sea, **M**iddle North Sea, **L**ower North Sea, **R**ijnland, **S**cruff, and **Z**echstein.

2 hours to train and test the encoder-decoder architecture described in section 2.1 using algorithms described in 2.2 and dataset described in 3.1. We do not use any data augmentation or post-processing (e.g. CRF) in these experiments. We use 10% of the training data for validation, and we train the models until convergence. To evaluate the proposed algorithm, we report mean Intersection over union (IoU) averaged over different classes. We also report individual class accuracy. These metrics are reported on test data in Table 1 while qualitative image comparison is provided in Figure 4.

3.3. Impact of Reconstruction

To better appreciate the added value of the reconstruction branch, we sweep over different n values and compare unconstrained versus constrained models. Figure 4 shows some samples of the test results of segmented facies. The original seismic image is shown in Figure 4(a) and its segmented version is shown in Figure 4(b). It is important to notice that for the dataset, some classes are always adjacent while others are always none. This is because different textures in 4(a) correspond to different earth layers which are formed during different geological eras. These geological constraints are consistent across the whole dataset. The rest of the left column shows unconstrained model results, while the rest of the right column are for constrained model. At

$n = 4\%$, Figure 4(g), unconstrained model confuses Upper North Sea class with Lower North Sea, such classes are geologically non-adjacent which implies that unconstrained models *did not learn adjacency constrains* of different classes from labeled data. Similarly, in Figure 4(e) at $n = 6\%$, Upper North Sea is confused with Scruff. However, the corresponding segmentation of constrained models, Figures 4(f) and 4(d), suffer much less from non-adjacent class confusion. This shows that constrained models have better *geometric reasoning about texture constraints* for different values of n. The performance improvement is due to the extra constraints imposed on the encoder by the reconstruction branch to ensure that the encoder representation is rich enough to reconstruct the original image.

3.4. Ablation Study

To quantitatively validate our hypothesis, we perform an ablation study and report the results in Table 1 where we sweep over n. For each value of n we perform each experiment 5 times and report the average value of each metric as well as the standard deviation across the 5 runs. From Table 1, it is evident that constrained models outperform unconstrained ones for most n. The performance improvement in IoU due to the reconstruction branch can go up to approximately 18% for $n = 2\%$. As we increase the value of n the performance gap decreases which is intuitive as both models get trained on more labeled data. But for our application, were obtaining labeled data is expensive in real life, it is cheaper to use less labeled data without much sacrificing the performance by building the geometric reasoning of the model using reconstruction branch during training.

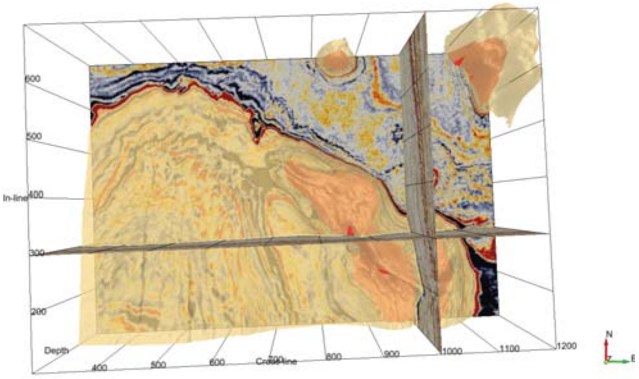
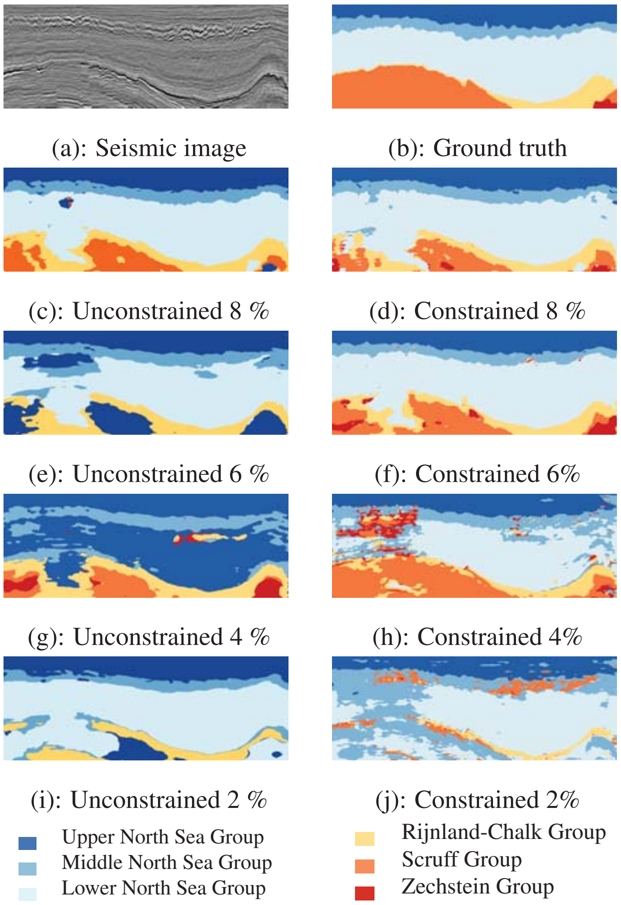


Fig. 3: A plan view of the F3 block with Zechstein group in red, and Chalk group in beige. Inline 300 and crossline 1000 split the survey into different regions for training and testing.



(a): Seismic image

(b): Ground truth

(c): Unconstrained 8 %

(d): Constrained 8 %

(e): Unconstrained 6 %

(f): Constrained 6%

(g): Unconstrained 4 %

(h): Constrained 4%

(i): Unconstrained 2 %

(j): Constrained 2%

Fig. 4: Segmentation results for unconstrained model versus constrained one at different percentages of training labeled data.

4. CONCLUSIONS

In this paper, we combined fully-supervised with self-supervised learning to train deep networks for better segmentation with less labels. The reconstruction branch helps improving the geometric reasoning of the network by forcing a richer encoder representation. In fact, for facies segmentation, the amount of labeled data can go down to 1% of the total training data without much sacrifice in performance. The reported results, using DeepLab V3+, quantitatively and qualitatively validates our hypothesis.

5. REFERENCES

[1] Maria-Florina Balcan and Avrim Blum, "A discriminative model for semi-supervised learning," *Journal of the ACM (JACM)*, vol. 57, no. 3, pp. 1–46, 2010.

[2] Olivier Chapelle, Bernhard Scholkopf, and Alexander

Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[3] Yves Grandvalet and Yoshua Bengio, "Semi-supervised learning by entropy minimization," in *Advances in neural information processing systems*, 2005, pp. 529–536.

[4] Samuli Laine and Timo Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[5] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen, "Mutual exclusivity loss for semi-supervised deep learning," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1908–1912.

[6] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[7] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng, "Transductive semi-supervised deep learning using min-max features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 299–315.

[8] Dong-Hyun Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3, p. 2.

[9] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 1476–1485.

[10] Carl Doersch, Abhinav Gupta, and Alexei A Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[11] Mehdi Noroozi and Paolo Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*. Springer, 2016, pp. 69–84.

[12] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9359–9367.

[13] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[15] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Advances in neural information processing systems*, 2014, pp. 766–774.

[16] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[17] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7354–7362.

[18] Yazeed Alaudah, Patrycja Michalowicz, Motaz Alfarraj, and Ghassan AlRegib, "A machine learning benchmark for facies classification," *arXiv preprint arXiv:1901.07659*, 2019.