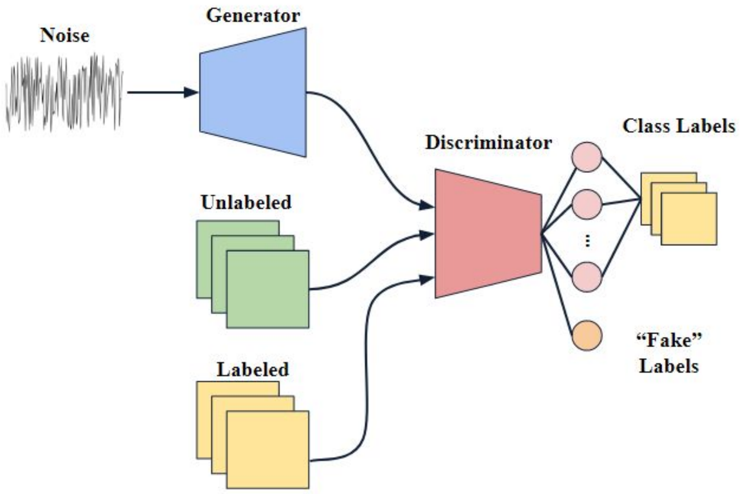
Master Thesis: Semi-supervised Classification in Earth Observation

Deep Learning (DL) methods such as Convolutional Neural Networks (CNNs) can deliver highly accurate classification results when provided with large enough datasets and respective labels. However, using CNNs along with limited labeled data can be problematic in Supervised Learning (SL) settings , as overfitting tends to occur and the final performance of the model (e.g., classification accuracy) is usually sub-optimal. This is perhaps even more true in the earth observation domain where satellite images are collected daily but the labeling process usually requires domain expertise and can be time-consuming.



Semi-Supervised Learning Model with GANs

In this thesis project, we aim to study the following questions regarding Semi-supervised Learning (SSL) methods in the Earth Observation domain:

1. How many labels are needed for an SSL model to reach the same, or at least comparable, performance as an SL model trained on the same dataset given enough labels?
2. How do the properties of the training dataset (e.g., dimension of each image patch, the content, and complexity of each image patch) affects the number of labels needed to train an SSL model to a predefined accuracy level?
3. How do the findings of the previous two questions hold for different types of SSL methods (e.g., GAN-based SSL[1] [2], MixMatch[3]?

---

[1] Salimans, Tim, et al. "Improved techniques for training gans." *Advances in neural information processing systems*. 2016.
[2] Odena, Augustus. "Semi-supervised learning with generative adversarial networks." *arXiv preprint arXiv:1606.01583* (2016).
[3] Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *Advances in Neural Information Processing Systems*. 2019.

If you are interested please send an email with your CV and latest version of transcript to tianzew@kth.se .

Academic supervisor: Tianze Wang (tianzew@kth.se)
Examiner: Prof. Vladimir Vlassov (vladv@kth.se)
Department of Computer Science, SCS division @ Kista

Master Thesis: On the Effect of Batch Size for Data Parallelization

While Deep Neural Networks (DNNs) have been successfully applied to many tasks, the training process of DNNs requires an increasing amount of computation resources. One common way to fasten the training process is to perform data parallelization using multiple GPUs (workers) where each worker holds a full copy of the DNN which is to be trained synchronously with the copies on other workers. Since parameters need to be synchronized regularly after each epoch of training, data transfer across different workers could be a potential bottleneck for speeding up the training process.

In this thesis project, we aim to answer the following questions:

1. In a single GPU training setting, how does the choice of batch size affect the training process? For example, is the training more stable using a smaller batch size? Would a larger batch size lead to lower performance after conversion?
2. How does the finding changes in the multi-GPU setting? Given a DNN model, a dataset, and a fixed number of GPUs, is there an optimal (or at least close to optimal) number for batch size in terms of training time and the performance of the final model?
3. Can we make any suggestions on batch size selection in practical settings?

Training Strategy	**Batch size**	**Training time - first epoch** (seconds)	**Training time - second epoch** (seconds)
Single GPU	32	348s	300s
Four GPUs with data parallelism	32	306s	224s
	4*32	197s	111s

An example on how batch size affects the training time

The thesis will be carried out through experimental studies and tested out on different DNN benchmark dataset. The results of different studies can be compared through measurements, e.g., classification accuracy, of different networks.

If you are interested please send an email with your CV and latest version of transcript to tianzew@kth.se .

Academic supervisor: Tianze Wang (tianzew@kth.se)
Examiner: Prof. Vladimir Vlassov (vladv@kth.se)
Department of Computer Science, SCS division @ Kista