University of St.Gallen

# Machine Learning - Coding Challenge - Spring 2023

Fabian Gubler | Sven Schnydrig

# Our team
Tensor Gone Wild

Fabian Gubler

Sven Schnydrig

# Agenda

The Challenge

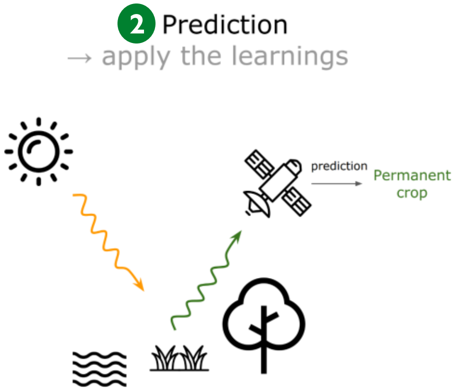
Preprocessing and our Approach

Network Details and Results

Learnings and Outlook

# The Challenge – Domain Shift from L1 to L2

## Remote Sensing Setup


Channels (= features)
- RGB → 3 channels
- Multispectral (e.g. 13 channels)
- Hyperspectral (e.g. 200 channels)

## Goal

**Training Set**
- 13 Bands
- GeoTiff

1 train model that learns **correlations** of Level 1C Training set that

**Test Set**
- 12 Bands
- npy files

2 … can then **generalize** its training to the Level 2A Test set

## Unique Challenges

**Doman Adaptation -** …
…

**Preprocessing -** …
…

# Preprocessing and Hyperparameter Configuration

## Preprocessing & Normalization

- Delete B1 & B9
- Reorder train bands in same order as test bands
- Add NDBI & NDMI index as additional bands
- Standard scaling (except for NDBI & NDMI)

## Other things we tried

- Various other normalization techniques
- Correlation Alignment (CORAL) & Maximum Mean Discrepancy for domain adaptation
- Use only RGB channels with same preprocessing as ImageNet
- Try to convert L1C to L2A data
- Engineer additional features & band indices

## Hyperparameter Tuning

| Parameter | Range | Final |
|---|---|---|
| Dense Layers | 1-4 | 1 |
| Neurons | 256-4'096 | 1024 |
| Dropout rate | 0-0.5 | 0.5 |
| L2 regularization | 0.001-0.2 | 0.1 |
| Scheduler | None, Decay, 1Cycle | Decay |
| Optimizer | SGD, RMSprop, Adagrad, Adam | SGD |
| Learning rate | 0.0001-0.1 | 0.003 |
| Batch size | 16-512 | 128 |

# Our Approaches

| | Learning from Scratch | Transfer Learning | (Pseudo) Self-Supervised |
|---|---|---|---|
| **Description** | 1. Network structure with **randomly initialized** weights<br>2. Training of **complete** network on training dataset | 1. Network structure with **pretrained** weights<br>2. Training of selected layers on the training dataset (= fine tuning) | 1. Network initially trained on labeled data<br>2. Model **uses its own predictions** for further self-supervised training |
| **Characteristics** | • **slow** convergence<br>• **lots** of training data necessary<br>• easy implemented for **every kind** of input data | • much **faster** convergence<br>• less training data necessary<br>• mostly implemented for **RGB**<br>• modifications for MS necessary | • Iteratively improves performance<br>• risks **reinforcing wrong bias**<br>• requires careful monitoring |

# Learning from scratch - Network Configuration


1 **ResNet Architecture**

ResNet-50 model, adapted to our task and input dimensions

2 **Configuration**

Regularization and data augmentation applied

3 **Randomized Weights**

We start the ResNet-50 model with randomly initialized weights, learning dataset specific features

# Outcome Analysis: From-Scratch approach

## What we tried …
Validation Accuracy: > 98 %

### Current Outcome
- Achieved high validation accuracy (~98%
- good fit on our training data.

### Problem Encountered
- Significant domain shift and lack of generalization
- bad fit on our testing data

## Why it didn't work out …
Submission Accuracy: 58 %

- Spent too much time on **overengineering** current approach
- Experienced tunnel vision, overlooking the **big picture**

**Realization:** Continuing current approach will not yield better results

**Decision:** Time to rethink our approach and explore other strategies

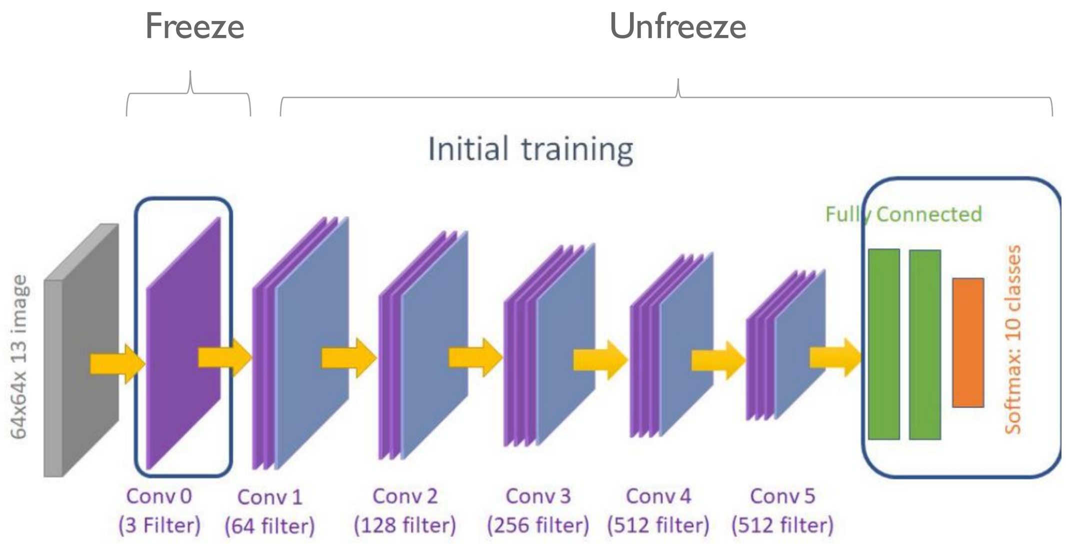
# Adapt Transfer learning for MS Data

*What?*
**Transfer Learning**

- Fine-tune preexisting models to mitigate domain shift.

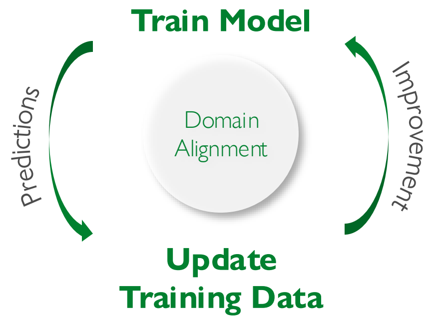
*How?*
**ImageNet**

- Utilize suitable pre-trained weights for the EuroSAT challenge



# Decreasing Domain Shift even further…

## Pseudo Self-Supervised Learning




- Counteract Domain Shift through **Pseudo-labeling**
- Unique solution when dealing with **unlabeled test set**

**Pro**

**Test Domain Familiarity:** The model learns from the test data, improving its generalization.

**Leverages Unlabeled Data:** Exploits abundant unlabeled data, extending training data.

**Iterative Refinement:** Model can progressively improve pseudo labels, boosting performance.

**Contra**

**Reinforces Errors:** Incorrect pseudo labels propagate errors, reinforcing biases.

**Overfitting Risk:** Model can overfit to its own generated labels.

Result accuracy gain of **2%**

# Overview of Results

| Model | Train Accuracy | Val Accuracy | Kaggle Accuracy |
|---|---|---|---|
| ResNet-50 training from scratch | 98.2% | 97.6% | 55.8% |
| ResNet-50 training from scratch & SVL[1] with PL[2] | 90.0% | 92.7% | 57.8% |
| ResNet-50V2 with transfer learning | 98.3% | 97.2% | 60.9% |
| ResNet-50V2 with transfer learning & SVL with PL | 90.8% | 92.2% | 63.1% |

[1] SVL: Self-supervised learning
[2] PL: Pseudo labels

# Key Takeaways

## Learnings

**Scientist Before Engineer:** Prioritize understanding over complex solutions. Simplicity can often outperform complexity.

**Address Domain Shift:** Aim for models that generalize well to the test set. Overfitting to training data is a pitfall.

**Embrace Creativity:** Unconventional methods like pseudo labeling can offer unexpected insights and breakthroughs.

## Outlook

- Expand on the theme of **simplicity**, by experimenting with the **simplest effective architecture**
- Exploring more **domain adaptation techniques** and **self-supervised learning**
- Test applicability of pseudo-labelling to other domains, **evaluate risk of overfitting** to the test set.
- Try other models like **vision transformers** or other approaches like training a neural network to **transform data from L1A to L2C**
- **Cleaner development process** and model log