Universität St.Gallen

ML Mavericks

Machine Learning - Coding Challenge - Spring 2023

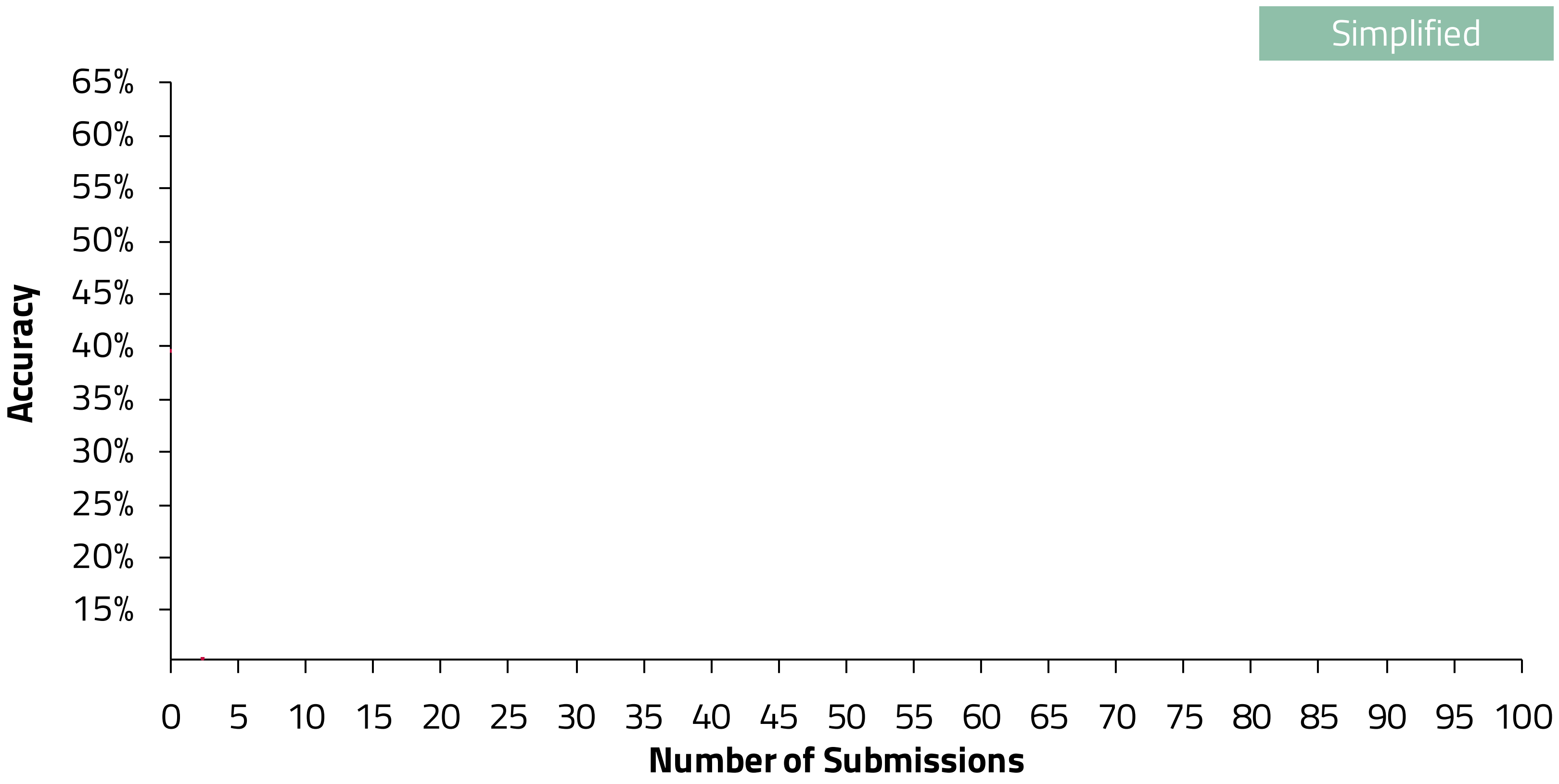
Florence Pfammatter, Kaan Aydin

Team Introduction

Florence Pfammatter

Kaan Aydin

Our Journey in the Coding Challenge



ResNet & ResNeXt

Models

ResNet50: default pretrained weights from PyTorch

ResNeXt: as above, architecture ResNeXt101_64X4D

Transformations

- Re-sizing to 92x92 pixels
- Random rotation of 5°
- Random crop to 112 pixels and padding of 10pixels
- Normalization ([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])

Experiments

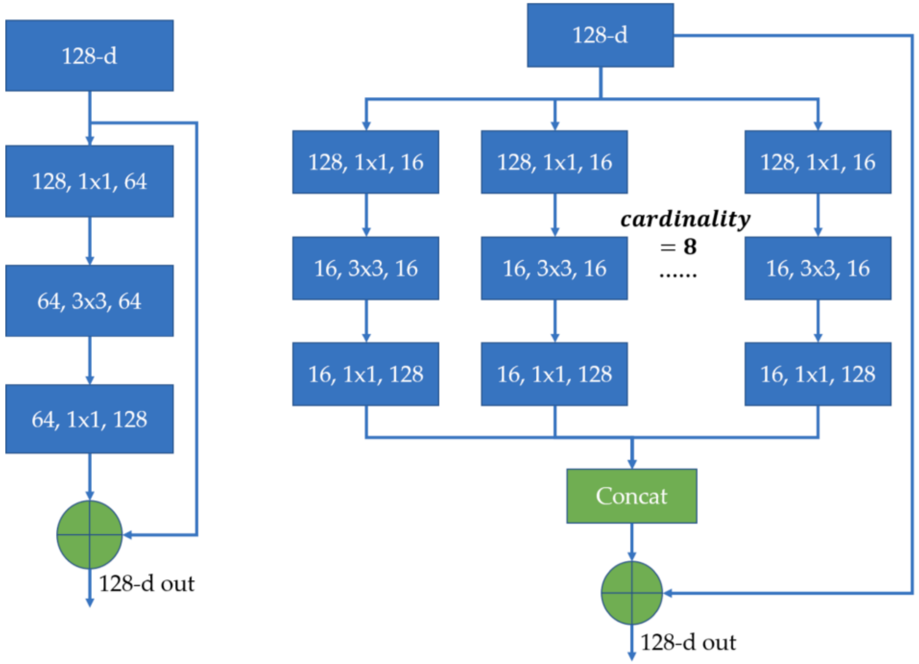
To reduce overfitting, we tried:

Regularization L1/L2, further transformations (Flippling, ColorJitter etc.), freezing layers of pretrained model, Dropout layer (with probability 0.2 and 0.5)

ResNeXt for better generalizability:

Increased capacity of network to capture broader feature variations through cardinality dimension.

Image from Wu et al. (2020)



Best Score: **ResNet50** 0.585 **ResNeXt** 0.617

Aggregated Residual Transformations for Deep Neural Network (Xie et al., 2017)

Vision Transformer

Models

Pretrained weights from PyTorch, VIT_B_16 (best performance on ImageNet1K with this image size)

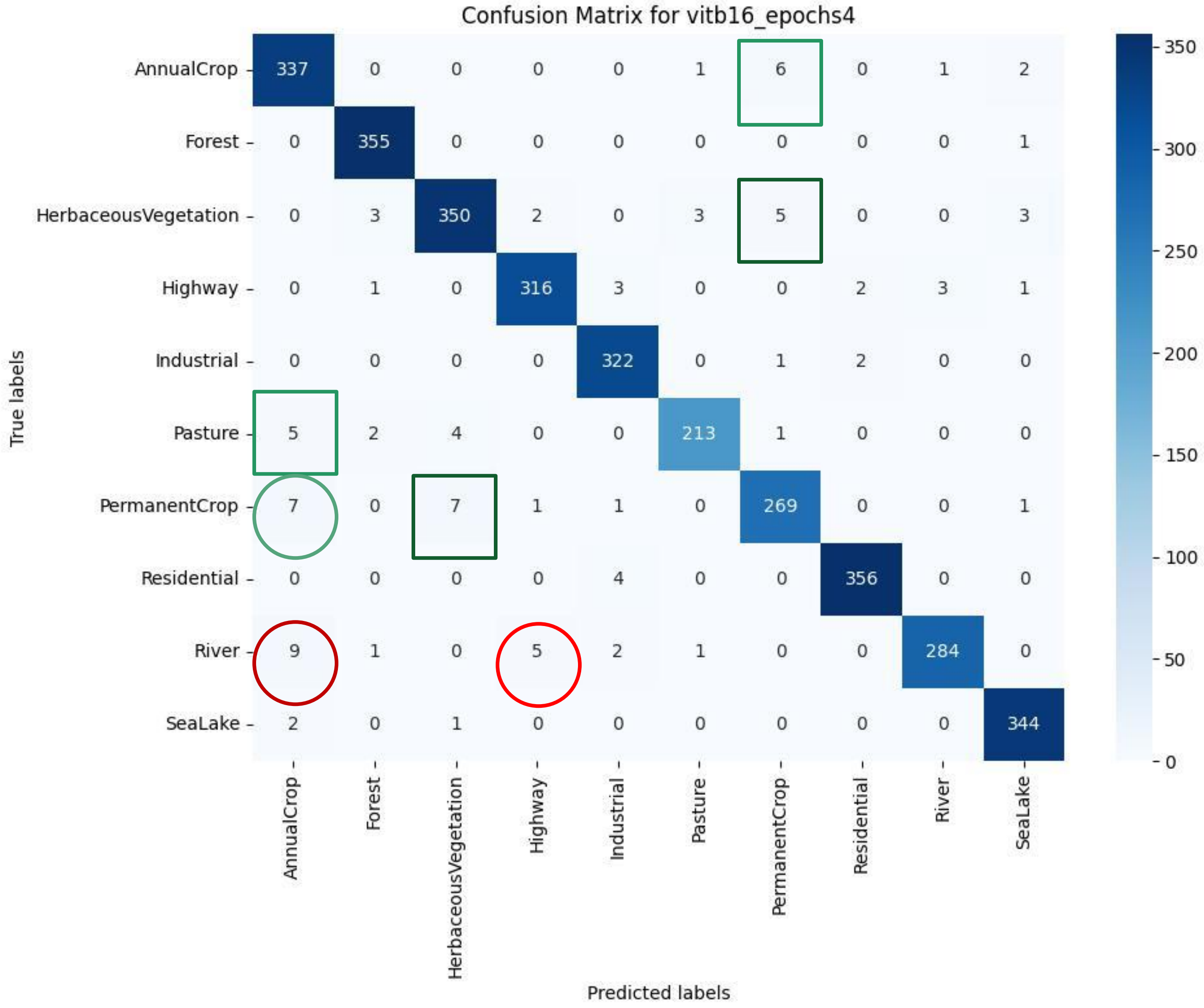
Transformations

- Re-sizing to 204x204 pixels
- Random rotation of 5°
- Random crop to 224 pixels and padding of 10pixels
- Normalization ([0.5, 0.5, 0.5], [0.5, 0.5, 0. 5])

Experiments

Image Resizing, DeiT model for low RAM consumption, Dropout Layer (with probability 0.3 and 0.5)

Best Score: **0.62564**

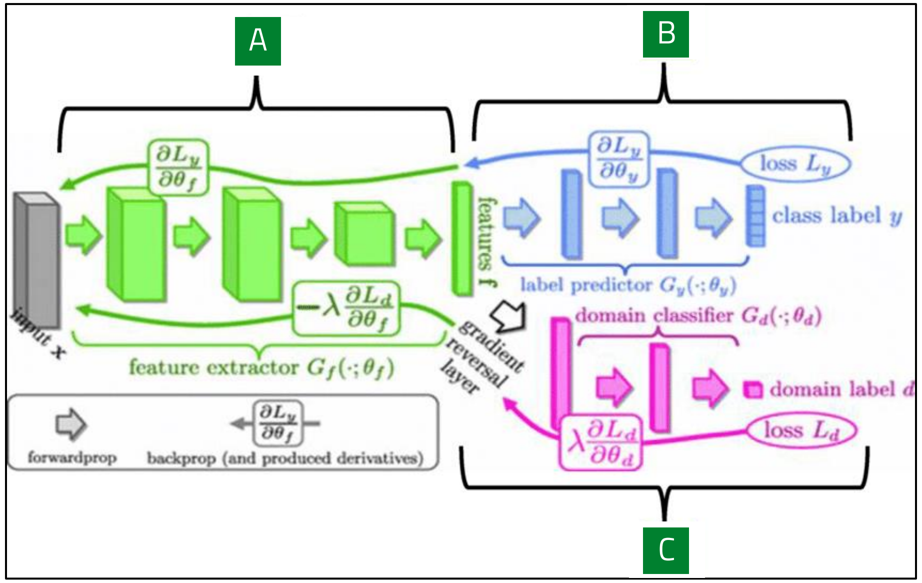


Confusion Matrix for vitb16_epochs4

True labels \ Predicted labels	AnnualCrop	Forest	HerbaceousVegetation	Highway	Industrial	Pasture	PermanentCrop	Residential	River	SeaLake
AnnualCrop	337	0	0	0	0	1	6	0	1	2
Forest	0	355	0	0	0	0	0	0	0	1
HerbaceousVegetation	0	3	350	2	0	3	5	0	0	3
Highway	0	1	0	316	3	0	0	2	3	1
Industrial	0	0	0	0	322	0	1	2	0	0
Pasture	5	2	4	0	0	213	1	0	0	0
PermanentCrop	7	0	7	1	1	0	269	0	0	1
Residential	0	0	0	0	4	0	0	356	0	0
River	9	1	0	5	2	1	0	0	284	0
SeaLake	2	0	1	0	0	0	0	0	0	344

Domain Adaptation (1/2) - Setup

Approach of DANN


A
B
$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features f
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input x
domain classifier $G_d(\cdot;\theta_d)$
gradient reversal layer
feature extractor $G_f(\cdot;\theta_f)$
domain label d
forwardprop
backprop (and produced derivatives)
$\lambda \frac{\partial L_d}{\partial \theta_d}$
loss L_d
C

- Based on the approach as outlined by Ganin et al., 2015
- Same feature extractor, two different heads: one for predicting label, the other for classifying domain
- Loss for classifying domain backprogates through reversal layer to maximize loss of domain classifier
- Goal: Increase loss for domain & decrease for label classifier

Deep Neural Network details

A

Feature Extractor

- Based on the feature extractor layers from the ResNet architecture (i.e., ResNet18 and ResNet50)
- Using default pre-trained weights from PyTorch

B

Label classifier

- 4-layer head on top of feature extractor, consisting of: one linear transformation (to number of classes), batch norm, ReLU activation function & SoftMax

C

Domain classifier

- 8-layer head on top of feature extractor, consisting of various linear mapping and ReLU activation functions and a SoftMax at the end

Experimental details

- 20 epochs, with learning rate of 1e-3 & momentum of 0.9
- Standard transformation (i.e., rotation, padding)
- For every second epoch, we only trained the domain classifier to generate better domain invariant features
- Keeping the λ constant at 0.2 for every iteration & epoch

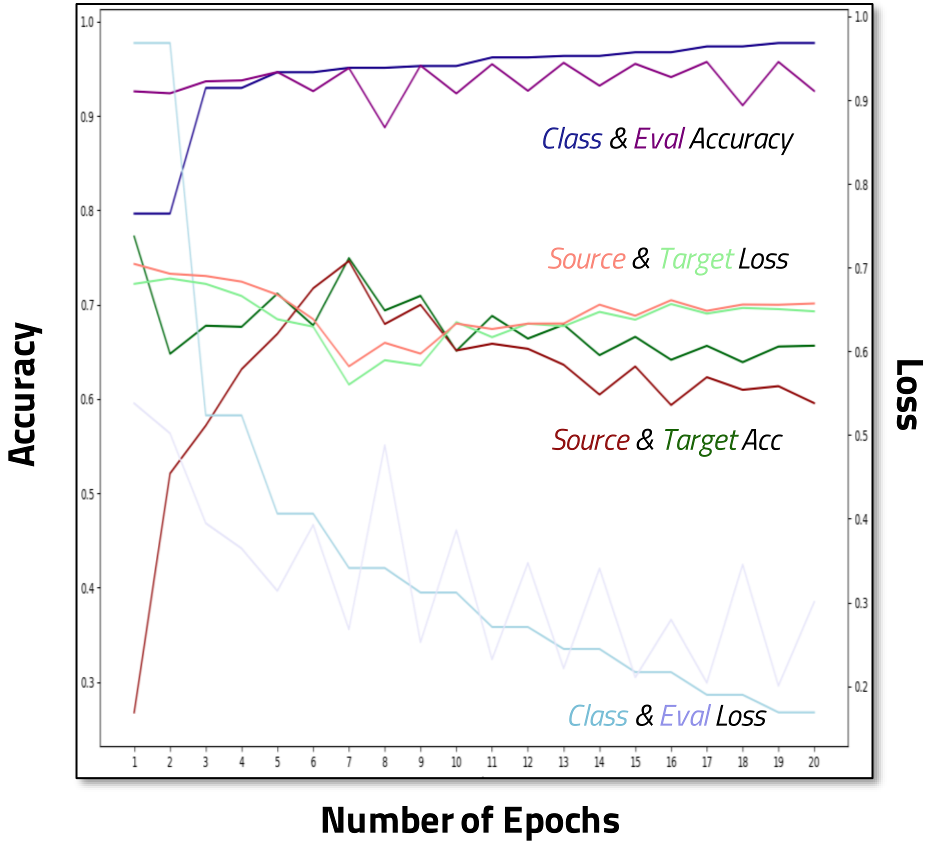
Domain Adaptation (2/2) - Result

High-level results

Feature Extractor	Result
ResNet50 *(w/ DA)*	50.4%
ResNet18 *(w/ DA)*	57.4%
ResNet18 (w/o DA)	52.6%

- All results are based on the same data transformations
- DA results are based on based on the same setup as previously shown
- For ResNet18, we show the epoch with the highest performance on eval

Accuracy & Loss curve



Best result achieved with ResNet18 (vs. ResNet50 despite same setup)

ResNet18 with DA performed better than without – indicating that DA has had some impact

Varying accuracy / loss of the domain classifier across epochs – little convergence

DA is promising given some initial results, but needs further deep-dive, e.g.:

- Implementing lower-layer feature extracors
- Adjusting the λ in the GRL

Universität St.Gallen

Learnings from the Challenge

Overfitting as key issue

Overfitting is a key factor to be considered - even more so when dealing with different domains

Target vs. Source domain

Little changes in augmentation can have a big impact on performance in target domain (vs. source)

Exploration vs. Exploitation

Good balance between trying new things out vs. pursuing existing approaches is important

Domain Adaptation

Domain Adaptation is difficult – has many considerations and factors to be tested

Development setup

A well-built development and model pipeline is key for speed, reproducibility and tracabiility

Outlook

Inclusion of NDVI features during training

Deep-dive in Domain Adaptation

Implementation of Hyperparameter Tuning

Training with cross-validation to reduce overfitting

Ensemble learning by combining various models

References

Wu, P., Cui, Z., Gan, Z., & Liu, F. (2020). Residual Group Channel and Space Attention Network for Hyperspectral Image Classification. *Remote Sensing*, *12*(12), 2035. https://doi.org/10.3390/rs12122035

Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. https://doi.org/10.1109/cvpr.2017.634

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, & Victor Lempitsky. (2016). Domain-Adversarial Training of Neural Networks.

THANK YOU! - ANY QUESTIONS?