Deep Semi-supervised learning (SSL)

On aerial datasets

UNIVERSITAT JAUME I

PyBasel
http://pybasel.ch

About me

- Itzá Hernández
- PhD. Computer Science - Computer Vision
- University Jaume I, Castellón, Spain
- @itzahs

Learning outcomes of this workshop

Get data

Create a class dataset with a transformation pipeline

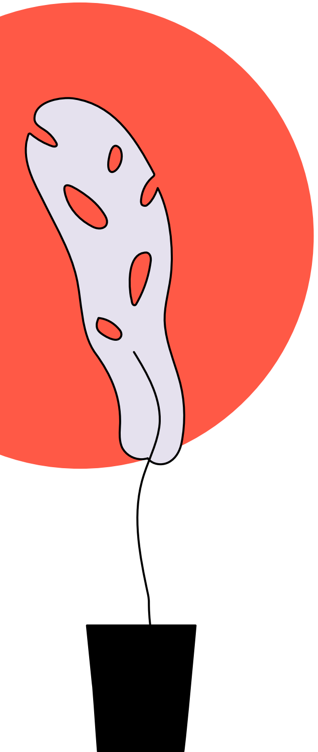
Train model

Train SSL model from a Google Colab environment

Clone repositories

Use software built by others and adapt it

Content of this workshop



01 Introduction to DSSL

14:00 - 14:20 (20 min)

02 Data & Code

14:20 - 14:50 (30 min)

03 Pause & Questions

14:50 - 15:00 (10 min)

04 Training model

15:00 - 15:30 (30 min)

Questions welcome anytime

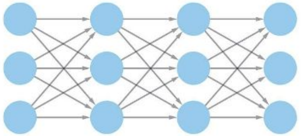
01

Introduction

Deep Semi Supervised Learning
14:00 - 14:20

Machine Learning & Deep Learning

ML

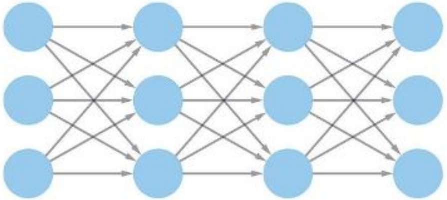


Feature extraction

Classification

Class probability

DL



Feature extraction + Classification

Class probability

Supervised

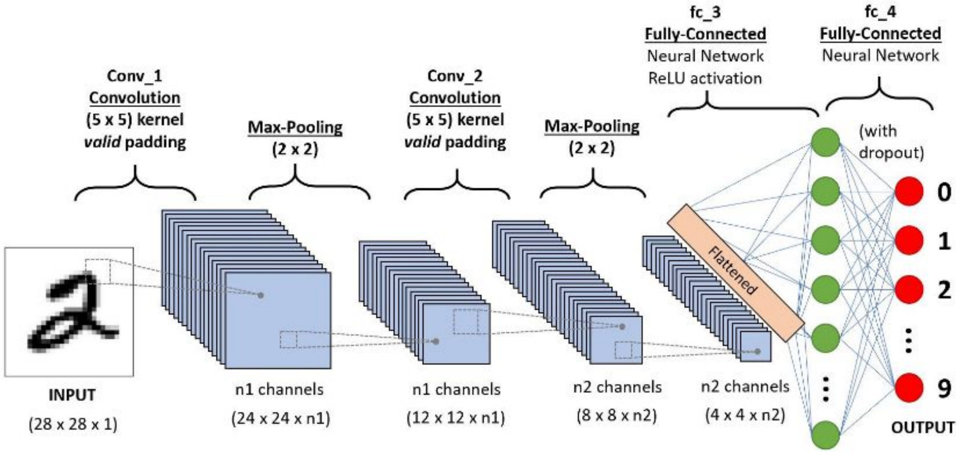
Semi-supervised

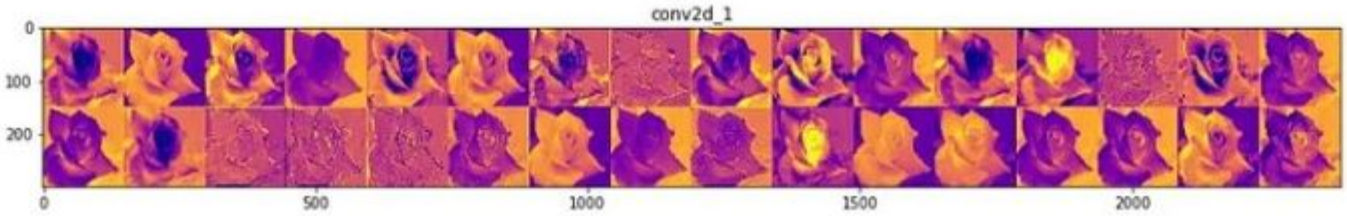
Unsupervised

[1] Bhavsar, K. (2018, September 14). Understanding your Convolution Network with Visualizations. Towards Data Science. https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b

Convolutional Neural Network (CNN)

Original Image

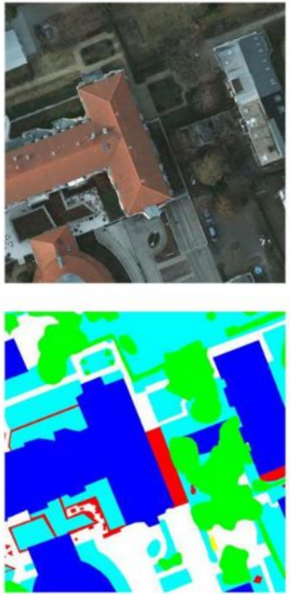




[1] Bhavsar, K. (2018, September 14). Understanding your Convolution Network with Visualizations. Towards Data Science. https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b

DL in Aerial Image Classification

Pixel-level

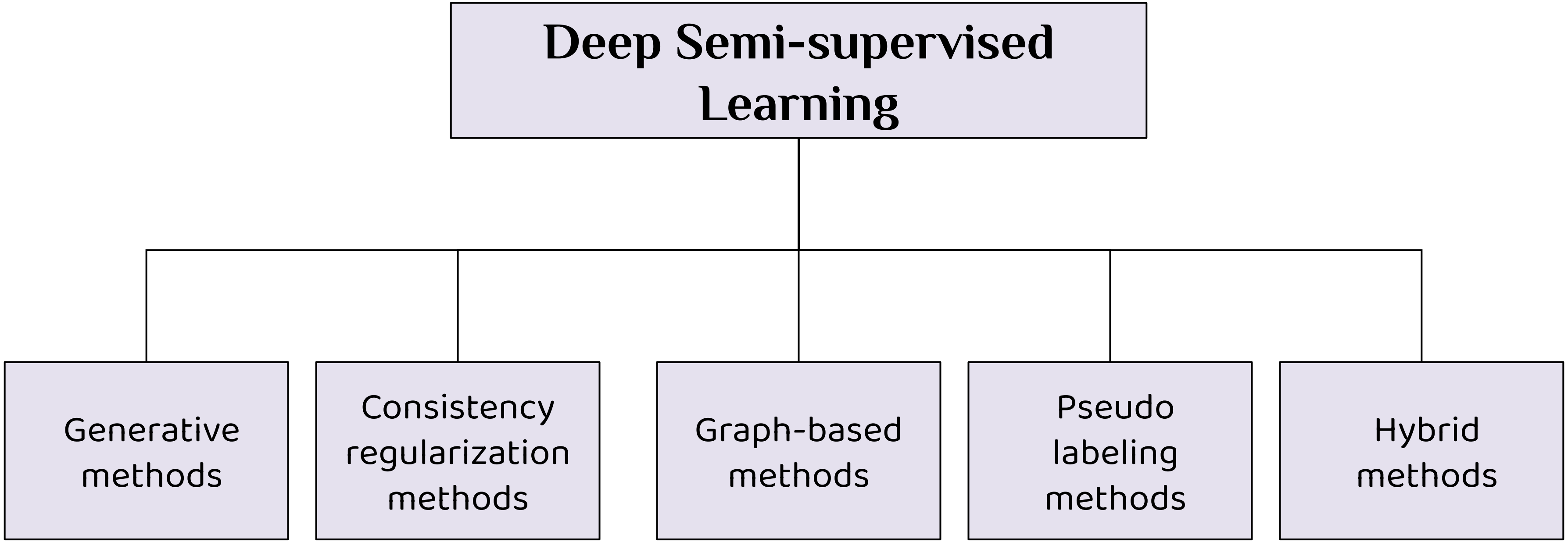


Object-level

Scene-level

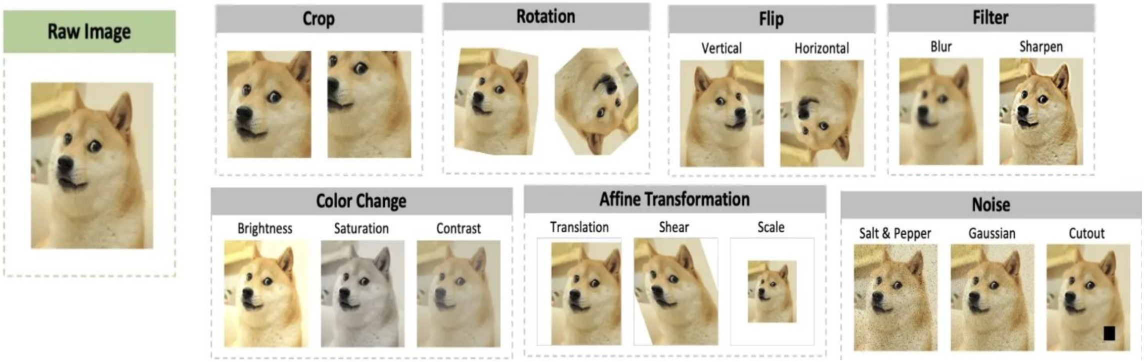
[2] G. Cheng, X. Xie, J. Han, L. Guo and G. -S. Xia, "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 3735-3756, 2020, doi: 10.1109/JSTARS.2020.3005403 .

Taxonomy



[3] X. Yang, Z. Song, I. King and Z. Xu, "A Survey on Deep Semi-Supervised Learning," in IEEE Transactions on Knowledge and Data Engineering, 2022, doi: 10.1109/TKDE.2022.3220219.

Image data augmentation for DL



- **Data Warping**: transforms existing images such as the label is preserved.

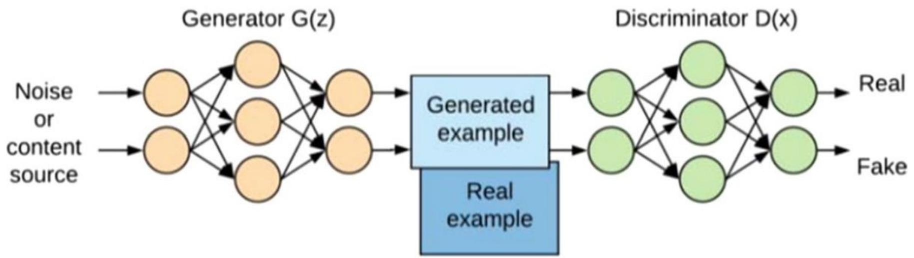
Synthetic data - CycleGANS - emotion classification

- **Oversampling**: creates synthetic instances and add them to the training set.

[4] Sharma, A. (2019, June 12). Complete Guide to Data Augmentation for Computer Vision. Towards Data Science. https://towardsdatascience.com/complete-guide-to-data-augmentation-for-computer-vision-1abe4063ad07.

[5] Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 (2019). https://doi.org/10.1186/s40537-019-0197-0

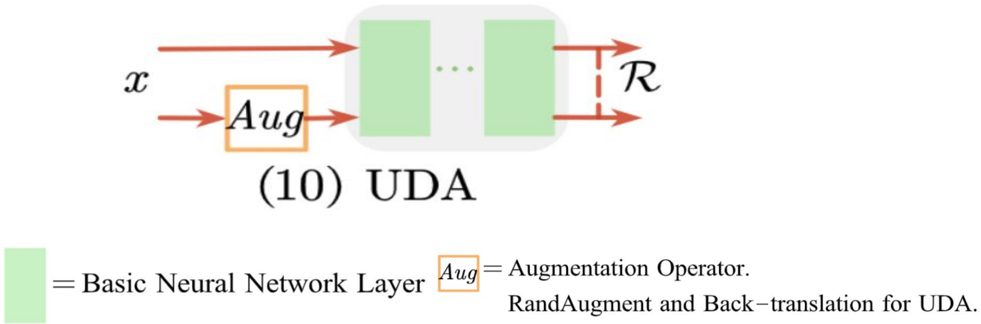
Generative methods



Generative modeling refers to the practice of **creating artificial instances** from a dataset such that they retain similar characteristics to the original dataset.

e.g. Generative Adversarial Networks (**GAN**) and Variational AutoEncoder (**VAE**).

Consistency regularization



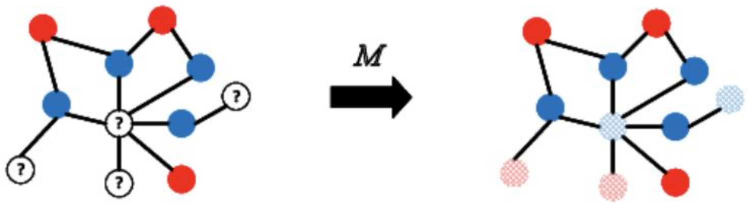
A consistency regularization term is applied to the **final loss function**. e.g. **Cross Entropy for example H(F(x), Tx).**

A realistic perturbation in the training data should not change the output of the model.

e.g. Unsupervised Data Augmentation (**UDA**).

[3] X. Yang, Z. Song, I. King and Z. Xu, "A Survey on Deep Semi-Supervised Learning," in IEEE Transactions on Knowledge and Data Engineering, 2022, doi: 10.1109/TKDE.2022.3220219.

Graph-based

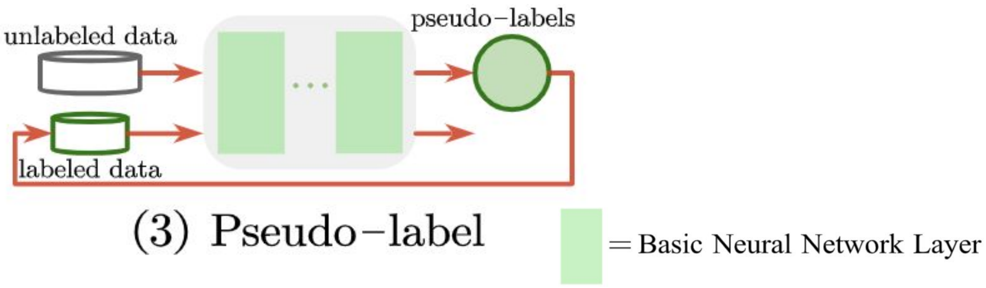


The nodes/vertices are **representations of the training samples** and the edges encode the relationships between the nodes.

The goal is to encode the nodes as small-scale vectors at first and then how each node belongs within the context in the graph.

Deep embedding methods are **AutoEncoders** and Graph Neural Networks (GNN).

Pseudo-labelling



Pseudo-labeling methods rely on the **high-confidence of pseudo-labels**, which can be added to the training data set as labeled data.

Self-training: leverage model's own confident predictions to produce the pseudo-labels for **unlabeled data**.

[3] X. Yang, Z. Song, I. King and Z. Xu, "A Survey on Deep Semi-Supervised Learning," in IEEE Transactions on Knowledge and Data Engineering, 2022, doi: 10.1109/TKDE.2022.3220219.

Hybrid: Fixmatch - Consistency Regularization and Pseudo Labeling

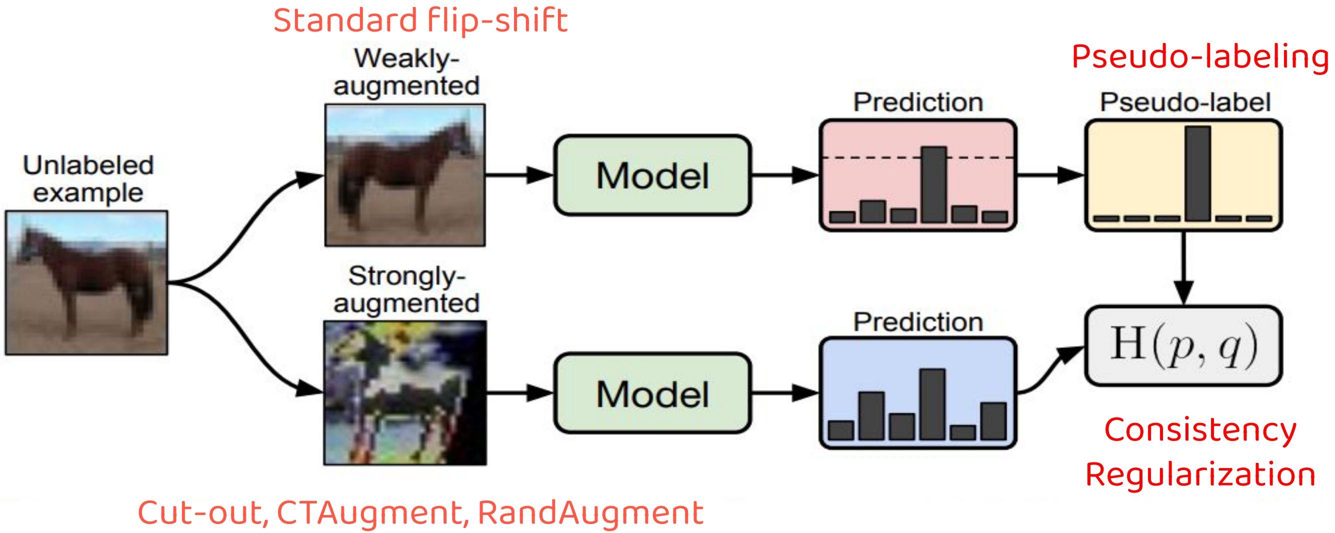


Figure 1: Diagram of FixMatch. A weakly-augmented image (top) is fed into the model to obtain predictions (red box). When the model assigns a probability to any class which is above a threshold (dotted line), the prediction is converted to a one-hot pseudo-label. Then, we compute the model's prediction for a strong augmentation of the same image (bottom). The model is trained to make its prediction on the strongly-augmented version match the pseudo-label via a cross-entropy loss.

[6] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. NeurIPS, 33, 2020.

02
Data & Code

Cloning the GitHub repositories
and downloading the datasets
14:20 - 14:50

Dataset and sampling stategy

UCM Dataset [download]:

- 21 clases
- 100 images per class
- 256 x 256 pixels (0.3 m)

Sampling strategy:

- 50 % training and 50% testing (Cheng, G. et al, 2020) [2].
- When using supervised 1,050 samples
- When using semi-supervised 84 samples or 4 labels per class (Sohn, K. et al, 2020) [6].

[7] Yi Yang and Shawn Newsam, "Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification," ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010.

Semi-supervised learning toolbox

Class-Aware Contrastive Semi-Supervised Learning

Publisher: IEEE Cite This PDF

Fan Yang ; Kai Wu ; Shuyi Zhang ; Guannan Jiang ; Yong Liu ; Feng Zheng ; Wei Zhang ; Chengjie Wang ; Long Zeng **All Authors**

Abstract:
Pseudo-label-based semi-supervised learning (SSL) has achieved great success on raw data utilization. However, its training procedure suffers from confirmation bias due to the noise contained in self-generated artificial labels. Moreover, the model's judgment becomes noisier in real-world applications with extensive out-of-distribution data. To address this issue, we propose a general method named Class-aware Contrastive Semi-Supervised Learning (CCSSL), which is a drop-in helper to improve the pseudo-label quality and enhance the model's robustness in the real-world setting. Rather than treating real-world data as a union set, our method separately handles reliable in-distribution data with class-wise clustering for blending into downstream tasks and noisy out-of-distribution data with image-wise contrastive for better generalization. Furthermore, by applying target reweighting, we successfully emphasize clean label learning and simultaneously reduce noisy label learning. Despite its simplicity, our proposed CCSSL has significant performance improvements over the state-of-the-art SSL methods on the standard datasets CIFAR100 [18] and STL10 [8]. On the real-world dataset Semi-iNat 2021 [27], we improve FixMatch [25] by 9.80% and CoMatch [19] by 3.18%. Code is available https://github.com/TencentYoutuResearch/Classification-SemiCLS.

Published in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Date of Conference: 18-24 June 2022

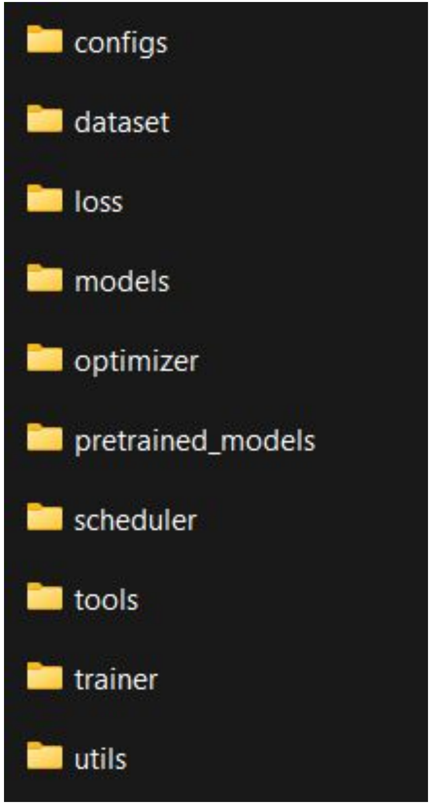
INSPEC Accession Number: 22095223

Date Added to IEEE *Xplore*: 27 September 2022

DOI: 10.1109/CVPR52688.2022.01402

[8] F. Yang et al., "Class-Aware Contrastive Semi-Supervised Learning," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 14401-14410, doi: 10.1109/CVPR52688.2022.01402.

Semi-supervised learning toolbox



https://github.com/TencentYoutuResearch/Classification-SemiCLS

[8] F. Yang et al., "Class-Aware Contrastive Semi-Supervised Learning," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 14401-14410, doi: 10.1109/CVPR52688.2022.01402.

Codes for this workshop

@itzahs github

GeoPythonWorkshop2023_SSL Public

This repository contains the codes for the GeoPython Workshop held in Basel, Switzerland on March 2023.

Jupyter Notebook

https://github.com/itzahs/GeoPythonWorkshop2023_SSL

Colab 1: UCM download

Colab 2: Clone repository

Space requirements

1. Google Account with 5GB space on Drive available
2. Optional: Google Drive Desktop and Visual Studio Code

UNIVERSITAT JAUME I

PyBasel

International Women's Day

Pause & Questions

8th March - International Women's Day

14:50 - 15:00

04
Model training

Modify the configuration files and launch the training
15:00 - 15:30

Modifications: configuration file

```python
train = dict(eval_step=1024,
             total_steps=2**20, #1024*20,
             trainer=dict(type="FixMatch",
                          threshold=0.95,
                          T=1.,
                          lambda_u=1.,
                          loss_x=dict(
                              type="cross_entropy",
                              reduction="mean"),
                          loss_u=dict(
                              type="cross_entropy",
                              reduction="none"),
                          ))
num_classes = 10 #21
```

1. How often model performance is evaluated.
2. Overall duration of the training process.

Modifications: configuration file

```
model = dict(
    type="wideresnet",
    depth=28,
    widen_factor=2,
    dropout=0,
    num_classes=num_classes,
)
```

WideResNet - Number of filters in the ResNet increased by a value of 2.

```
cifar10_mean = (0.4914, 0.4822, 0.4465)
cifar10_std = (0.2471, 0.2435, 0.2616)

#ucm_mean = (0.485, 0.456, 0.406)
#ucm_std = (0.229, 0.224, 0.225)
```

ImageNet mean and std of the pixel intensities for each color channel (RGB).

UNIVERSITAT JAUME I

PyBasel http://pybasel.ch

[9] S. Zagoruyko and N. Komodakis, "Wide residual networks," in Proceedings of the British Machine Vision Conference (BMVC), 2017, pp. 87.1-87.12, doi: 10.22024/UniKent/01.02.69550, arXiv:1605.07146.

Modifications: configuration file

```
data = dict(
    # CIFAR10SSL, CIFAR100SSL
    type="CIFAR10SSL", #"MyDataset",
    num_workers=4, #0,
    num_labeled=250, #84,
    num_classes=num_classes,
    batch_size=64, #8,
    expand_labels=False,
    mu=7,

    root="./data/CIFAR",
    #root="./UCMerced_LandUse/Images",
    #labeled_names_file="./UCMerced_LandUse/Images/UCM_train.txt",
    #test_names_file="./UCMerced_LandUse/Images/UCM_test.txt",
```

4 labeled examples per class.

Path to images

Modifications: configuration file

```
lpipelines=[[
    # 50% chances that the image is horizontally flipped
    dict(type="RandomHorizontalFlip"),
    # RandomCrop crops a fixed size whereas RandomResizedCrop crops and then resizes.
    dict(type="RandomCrop",
         size=32,
         padding=int(32 * 0.125),
         padding_mode='reflect'),
    #dict(type="RandomResizedCrop", size=224, scale=(0.2, 1.0)),
    dict(type="ToTensor"),
    dict(type="Normalize", mean=cifar10_mean, std=cifar10_std)
    #dict(type="Normalize", mean=ucm_mean, std=ucm_std)
]],
```

```
upipelinse=[[
    dict(type="RandomHorizontalFlip"),
    dict(type="RandomCrop",
         size=32,
         padding=int(32 * 0.125),
         padding_mode='reflect'),
    #dict(type="Resize", size=256),
    #dict(type="CenterCrop", size=224),
    dict(type="ToTensor"),
    dict(type="Normalize", mean=cifar10_mean, std=cifar10_std)
    ],
    [
    dict(type="RandomHorizontalFlip"),
    dict(type="RandomCrop",
         size=32,
         padding=int(32 * 0.125),
         padding_mode='reflect'),
    #dict(type="RandomResizedCrop", size=224, scale=(0.2, 1.0)),
    dict(type="RandAugmentMC", n=2, m=10),
    dict(type="ToTensor"),
    dict(type="Normalize", mean=cifar10_mean, std=cifar10_std)
    #dict(type="Normalize", mean=ucm_mean, std=ucm_std)
]],
```

Modifications: configuration file

Augmentation strategy

- **Weak:**
  - Flip
  - Crop
- **Strong1:**
  - RandomAugment
  - CutOut
- **Strong2:**
  - Random Color Jittering
  - Grayscale Conversion

UNIVERSITAT JAUME I

PyBasel

Modifications: configuration file

```
vpipeline=[
    #dict(type="Resize", size=256),
    dict(type="ToTensor"),
    dict(type="Normalize", mean=cifar10_mean, std=cifar10_std)
    #dict(type="Normalize", mean=ucm_mean, std=ucm_std)
])
```

Some UCM samples are have inconsistent spatial sizes, whiles most have a shape of 256*256*3 some are 253*256*3.

Modifications: dataset builder file

```
else:

    # check if .ipynb_checkpoints is in root, and exclude it
    root = os.path.join(cfg.root, '')
    if os.path.isdir(os.path.join(root, ".ipynb_checkpoints")):
        exclude_dir = os.path.join(root, ".ipynb_checkpoints")
    else:
        exclude_dir = None
```

Google colab creates temporary .ipynb inside the image folders and the dataset class gives error when other than .tif extensions are found.

Modifications: training file

```
            data_x = labeled_iter.next()
            #data_x = next(labeled_iter)
        except Exception:
            if args.world_size > 1:
                labeled_epoch += 1
                labeled_trainloader.sampler.set_epoch(labeled_epoch)
            labeled_iter = iter(labeled_trainloader)
            data_x = labeled_iter.next()
            #data_x = next(labeled_iter)

        try:
            data_u = unlabeled_iter.next()
            #data_u = next(unlabeled_iter)
        except Exception:
            if args.world_size > 1:
                unlabeled_epoch += 1
                unlabeled_trainloader.sampler.set_epoch(unlabeled_epoch)
            unlabeled_iter = iter(unlabeled_trainloader)
            data_u = unlabeled_iter.next()
            #data_u = next(unlabeled_iter)
```

Thanks!

Do you have any questions?

isequeir@uji.es

@itzahs

References

Bibliography:

1. Bhavsar, K. (2018, September 14). Understanding your Convolution Network with Visualizations. Towards Data Science. https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b
2. G. Cheng, X. Xie, J. Han, L. Guo and G. -S. Xia, "Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 3735-3756, 2020, doi: 10.1109/JSTARS.2020.3005403 .
3. X. Yang, Z. Song, I. King and Z. Xu, "A Survey on Deep Semi-Supervised Learning," in IEEE Transactions on Knowledge and Data Engineering, 2022, doi: 10.1109/TKDE.2022.3220219.
4. Sharma, A. (2019, June 12). Complete Guide to Data Augmentation for Computer Vision. Towards Data Science. https://towardsdatascience.com/complete-guide-to-data-augmentation-for-computer-vision-1abe4063ad07.
5. Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 (2019). https://doi.org/10.1186/s40537-019-0197-0
6. Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. NeurIPS, 33, 2020.
7. Yi Yang and Shawn Newsam, "Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification," ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010.
8. F. Yang et al., "Class-Aware Contrastive Semi-Supervised Learning," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 14401-14410, doi: 10.1109/CVPR52688.2022.01402.
9. S. Zagoruyko and N. Komodakis, "Wide residual networks," in Proceedings of the British Machine Vision Conference (BMVC), 2017, pp. 87.1-87.12, doi: 10.22024/UniKent/01.02.69550, arXiv:1605.07146.

Presentation credits:

1. Template by **Slidesgo**
2. Icons by **Flaticon**
3. Infographics and images by **Freepik**

UNIVERSITAT JAUME I

PyBasel