A Survey of Un-, Weakly-, and Semi-Supervised Learning Methods for Noisy, Missing and Partial Labels in Industrial Vision Applications

SDS 2021

Talk by Niclas Simmler & Pascal Sager

June 9th 2021

Authors

Niclas Simmler
HSLU Luzern
niclas.simmler@stud.hslu.ch

Pascal Sager
ZHAW Winterthur
sage@zhaw.ch

Dr. Ricardo Chavarriaga
ZHAW Winterthur
chav@zhaw.ch

Dr. Frank-Peter Schilling
ZHAW Winterthur
scik@zhaw.ch

Philipp Andermatt
KITRO SA
philipp.andermatt@kitro.ch

Prof. Dr. Matthias Rosenthal
ZHAW Winterthur
rosn@zhaw.ch

Prof. Dr. Thilo Stadelmann
ZHAW Winterthur
stdm@zhaw.ch

Table of Contents

Use Cases & Terminology

Root Causes & Goals

Noisy Labels

Missing Labels

Partially Labeled Data

Discussion

Questions

Use Cases

Use Case 1: Classification

Input

Target

Horse

Use Case 2: Semantic Segmentation

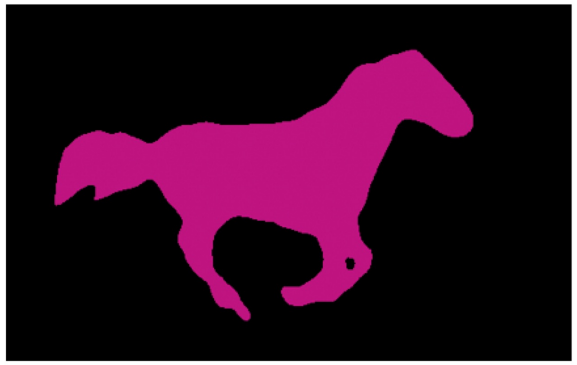
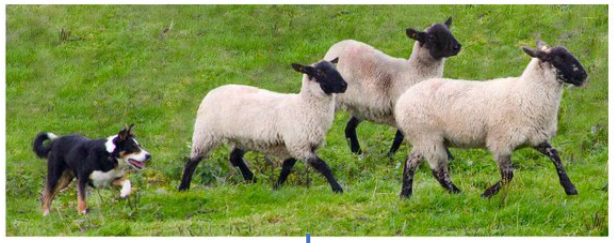


Image Source: https://www.mdpi.com/1424-8220/19/24/5361/htm

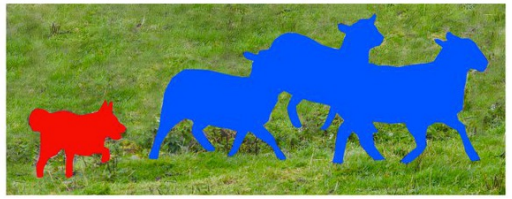
Missing, Noisy and Partial Labels

Given Data



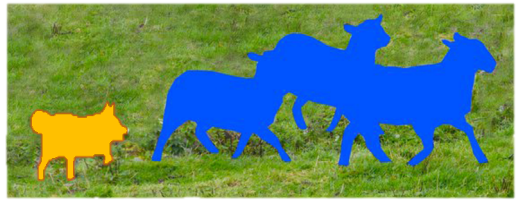
Correct

Dog, Sheeps



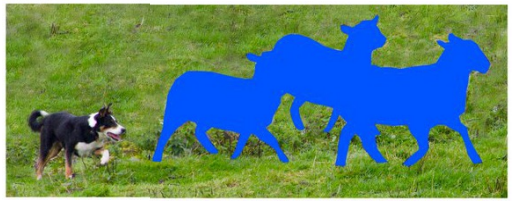
Noisy Label

Cat, Sheeps



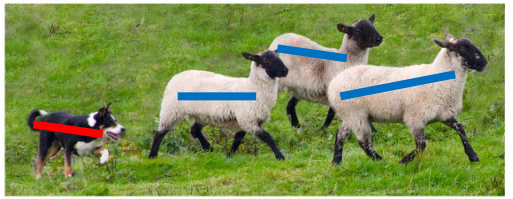
Missing Label

Sheeps



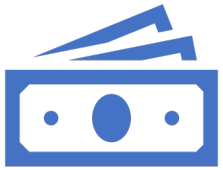
Partial Label

-



Example Image Source: https://pypi.org/project/SemTorch/

Root Causes



Costs

No quality check

Time constraint

Perception Issues

Inter-observer variability

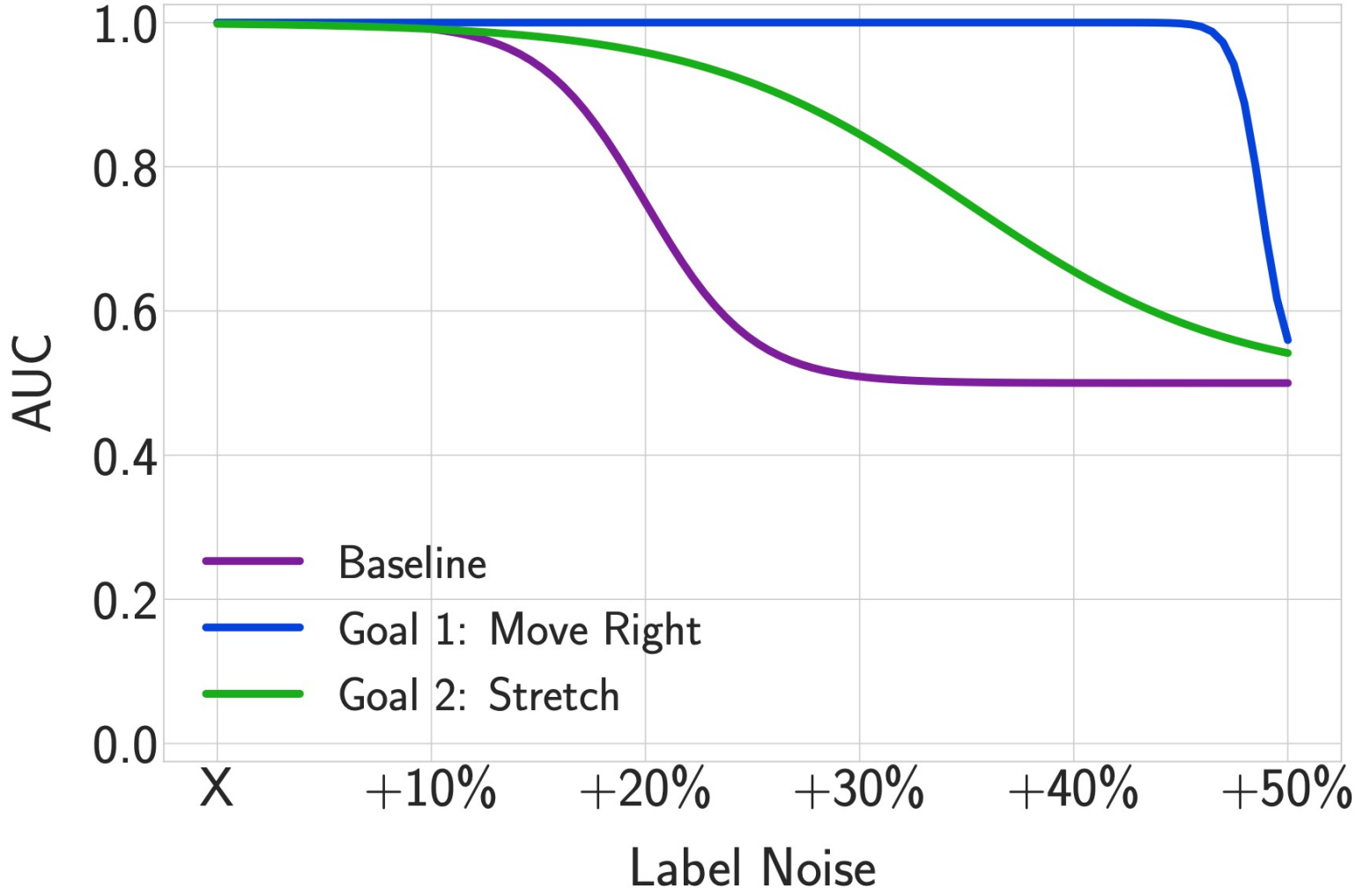
Lack of expertise

Errors in Labeling Process

Human error

Incorrect sensor readings

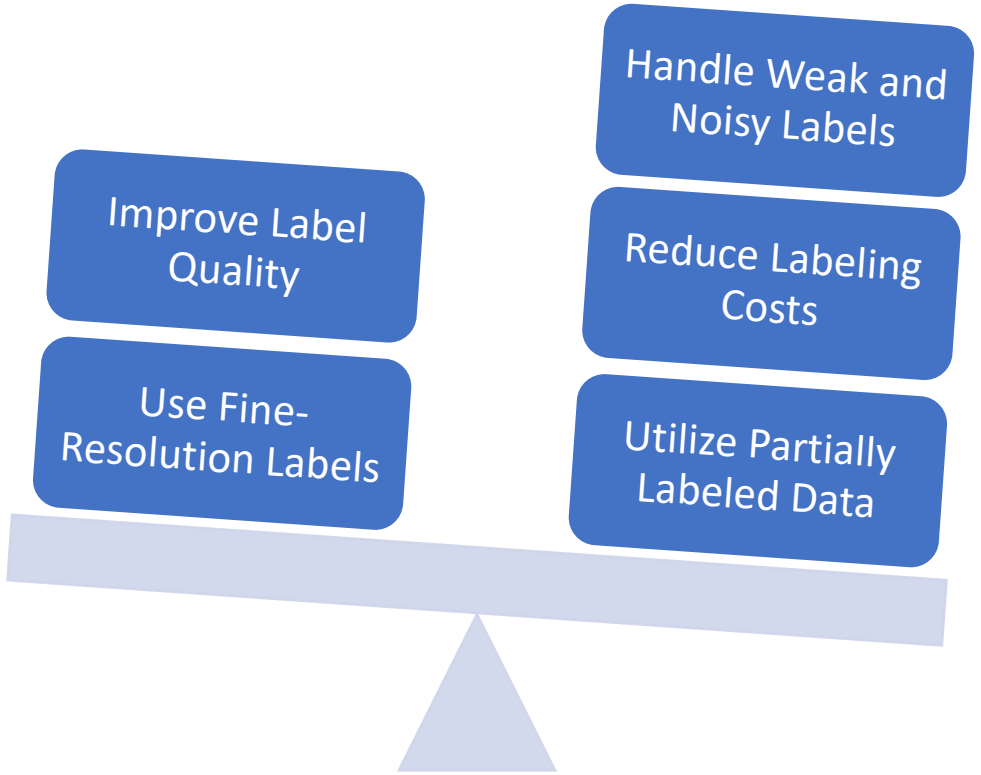
Goal


1.0
0.8
0.6
0.4
0.2
0.0
AUC
Baseline
Goal 1: Move Right
Goal 2: Stretch
X
+10%
+20%
+30%
+40%
+50%
Label Noise

Methods & Benefits

Solve on Data-Side

Solve on Model-Side



Noisy Labels: DivideMix [1]

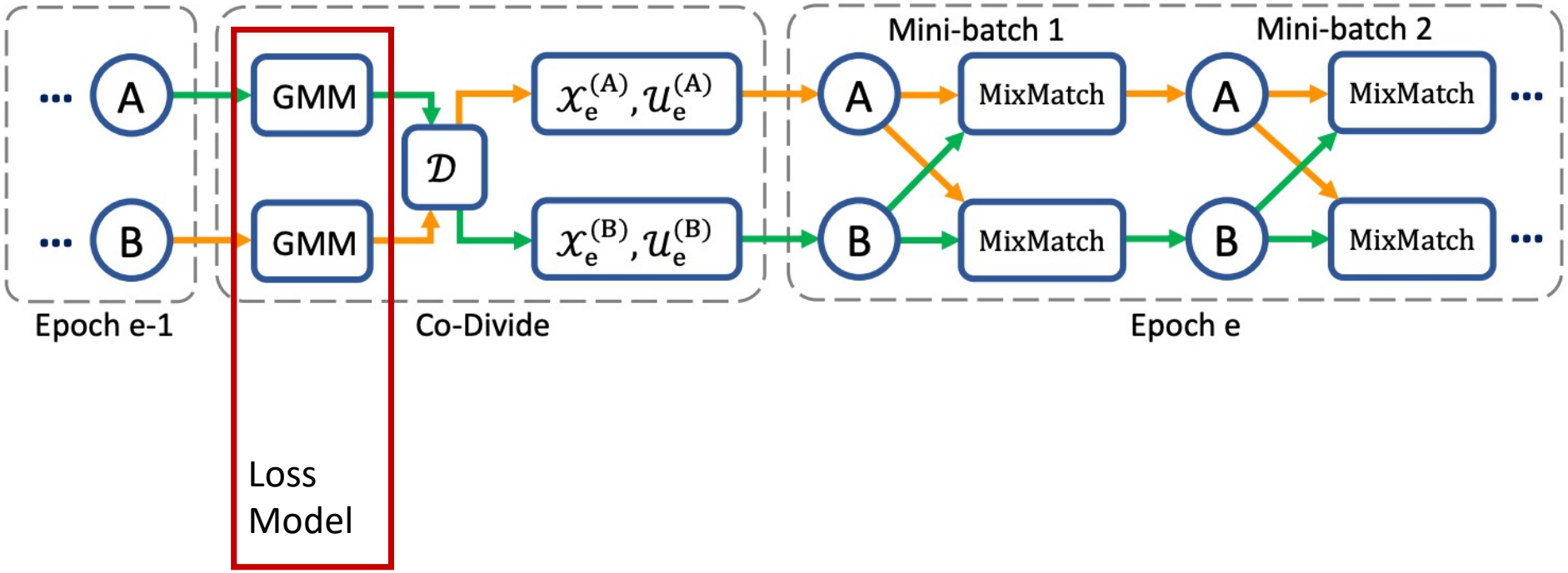
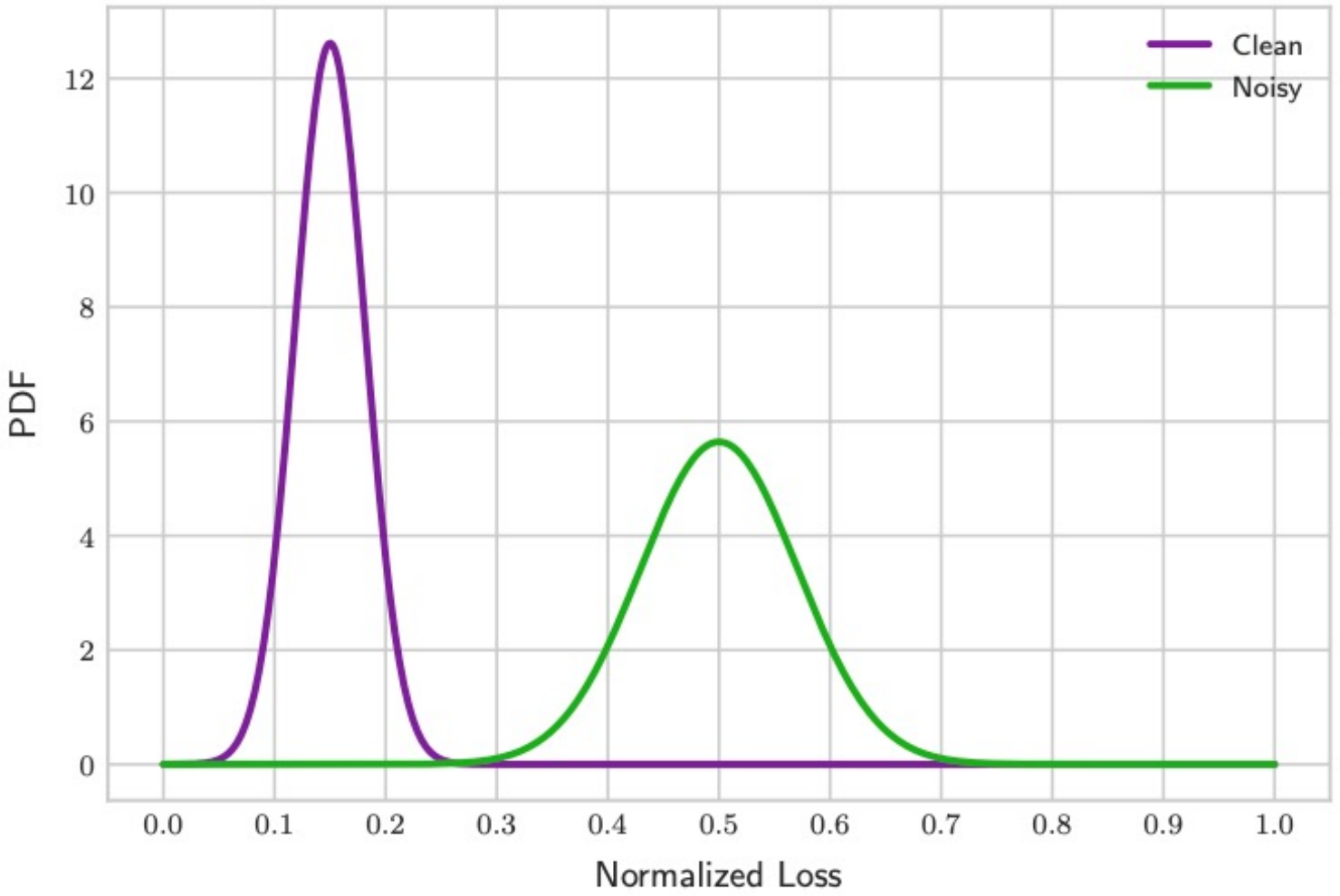
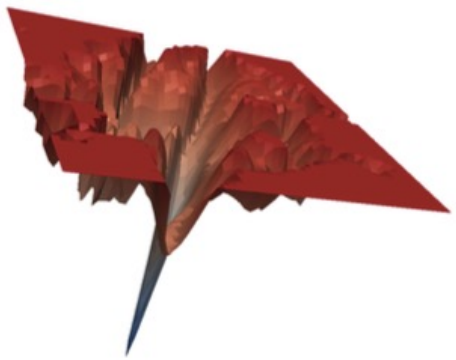


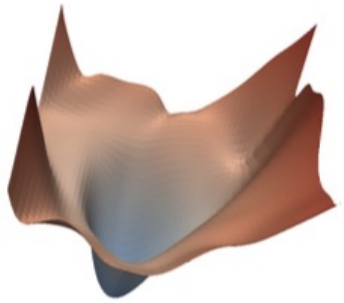
Image Source: https://arxiv.org/pdf/2002.07394.pdf

Noisy Labels: DivideMix [1]



Noisy Labels: Sharpness-Aware Minimization [2]





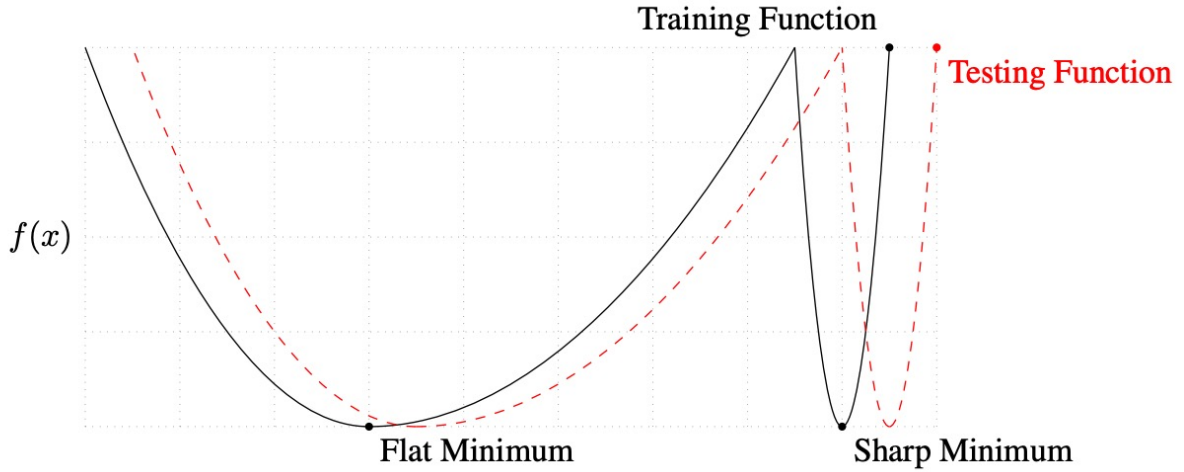
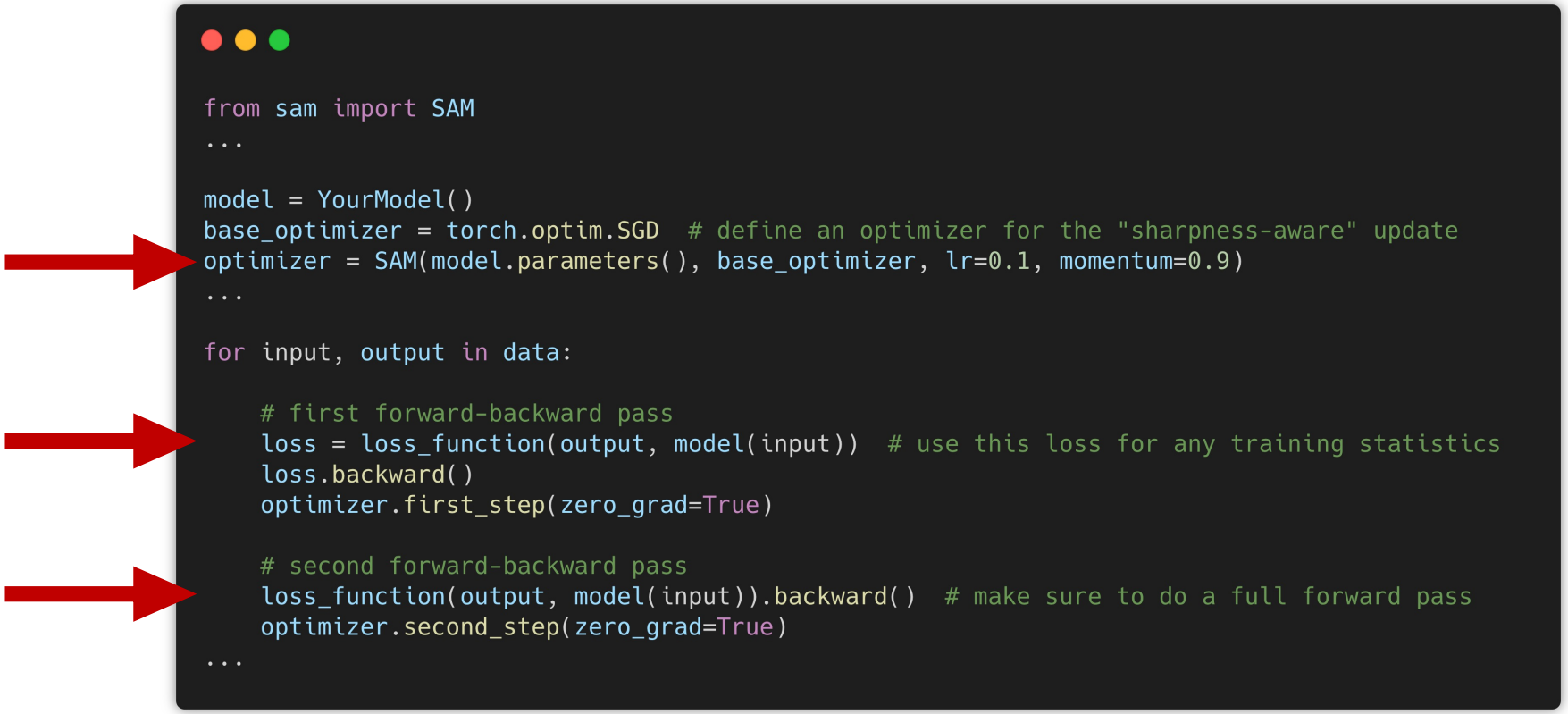


Image Source: https://arxiv.org/pdf/2010.01412.pdf

Image Source: https://arxiv.org/pdf/1609.04836.pdf

Noisy Labels: Sharpness-Aware Minimization [2]



```
from sam import SAM
...

model = YourModel()
base_optimizer = torch.optim.SGD  # define an optimizer for the "sharpness-aware" update
optimizer = SAM(model.parameters(), base_optimizer, lr=0.1, momentum=0.9)
...

for input, output in data:

    # first forward-backward pass
    loss = loss_function(output, model(input))  # use this loss for any training statistics
    loss.backward()
    optimizer.first_step(zero_grad=True)

    # second forward-backward pass
    loss_function(output, model(input)).backward()  # make sure to do a full forward pass
    optimizer.second_step(zero_grad=True)
...
```

Code Source: https://github.com/davda54/sam

Missing Labels: Label Propagation [3]

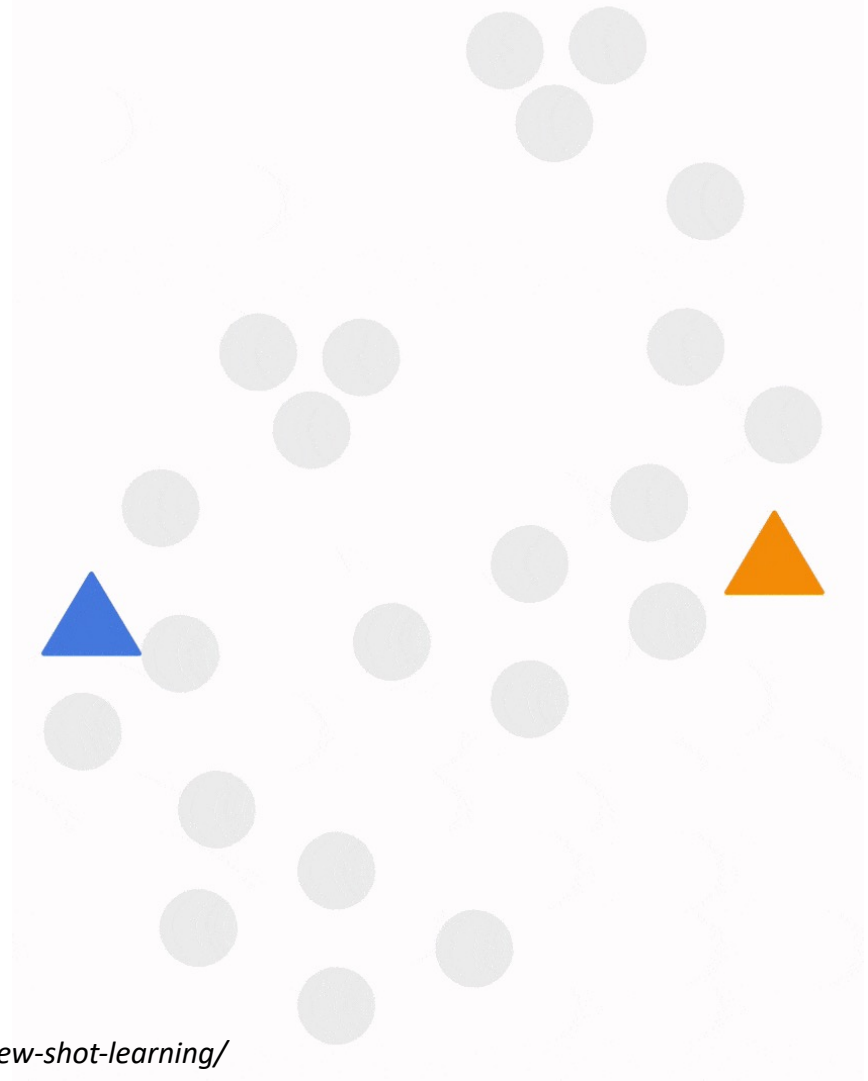


Image Source: https://opencv.org/understanding-transductive-few-shot-learning/

Missing Labels: Label Propagation [3]

Transductive Propagation Network [4]

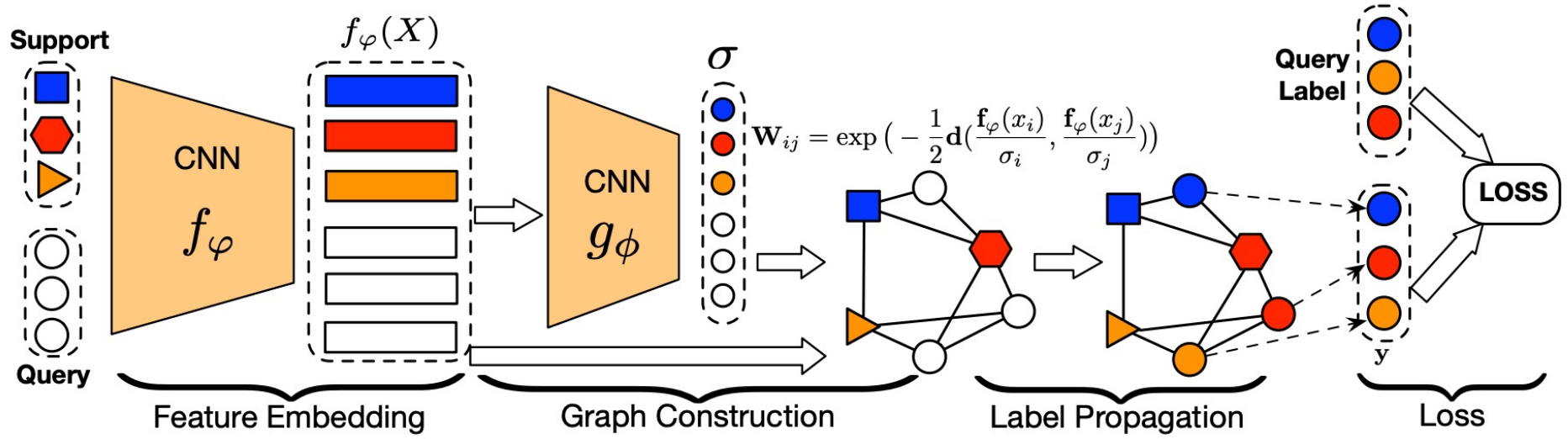
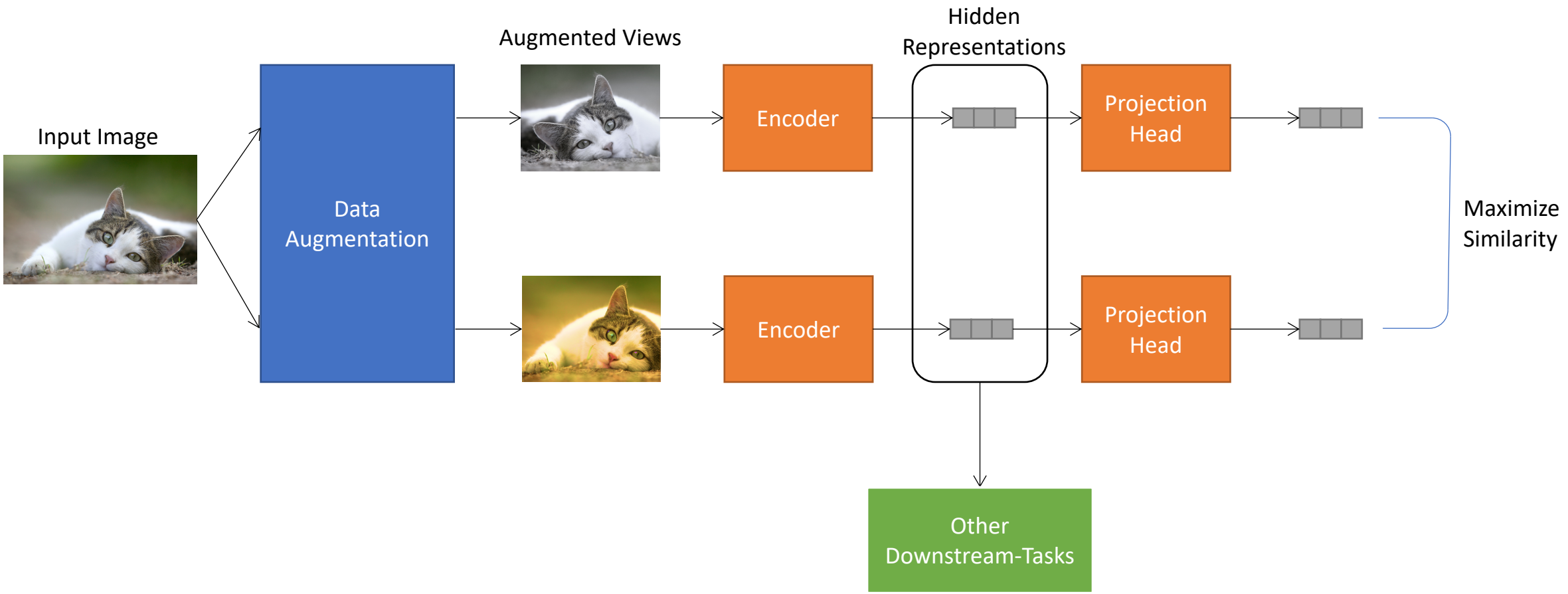


Image Source: https://arxiv.org/pdf/1805.10002.pdf

Missing Labels: Contrastive Learning [5], [6]



Partially Labeled Data [7]

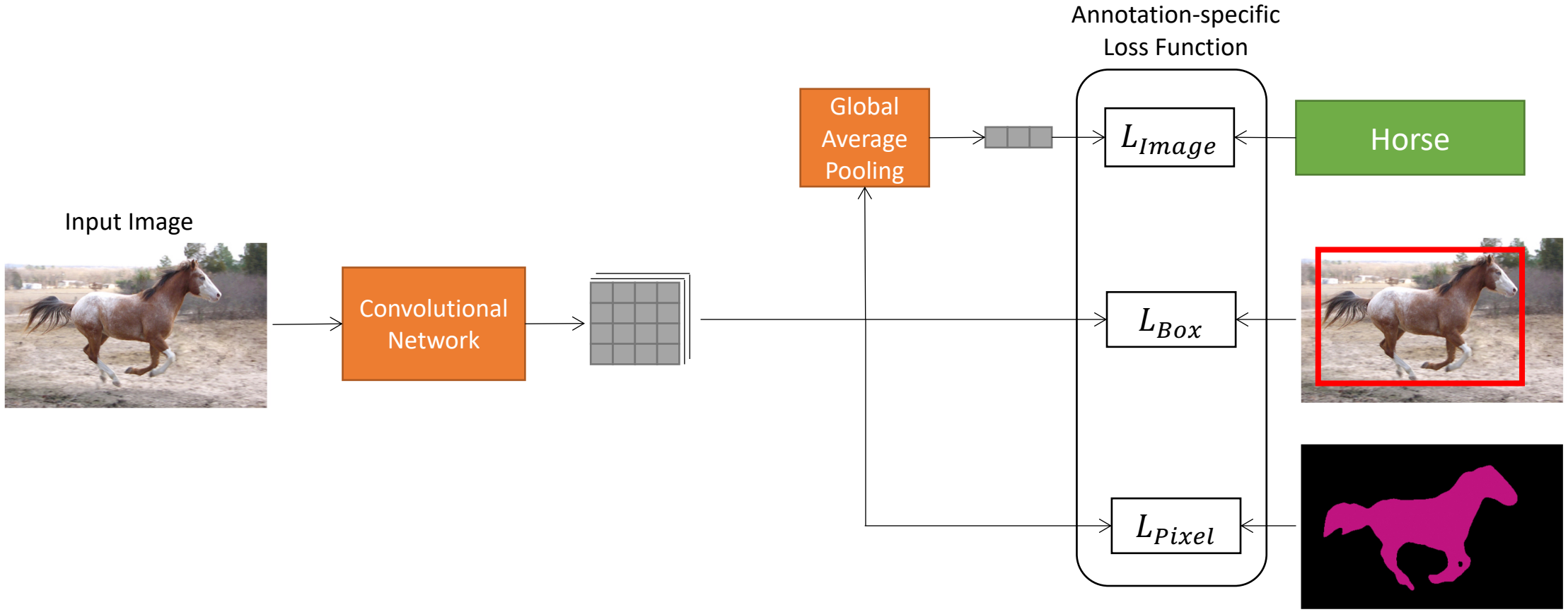
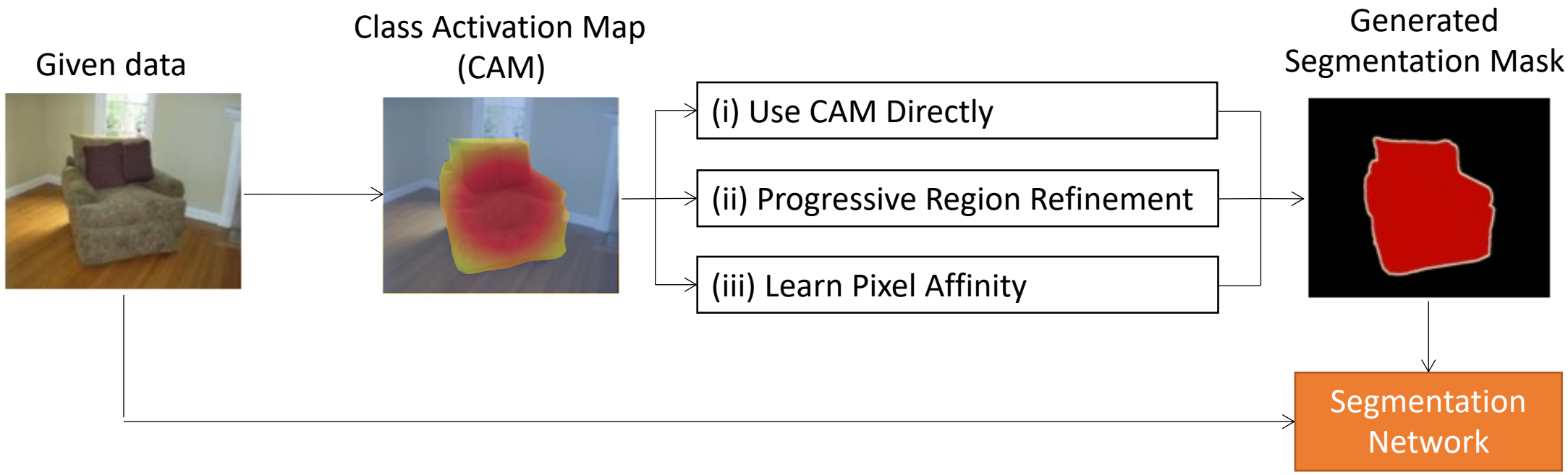


Image Source: https://www.mdpi.com/1424-8220/19/24/5361/htm

Partially Labeled Data [8]-[10]

1. Generate object-based labels from class activation maps
2. Train semantic segmentation networks based on those maps



Example Image Source: https://arxiv.org/pdf/1509.03150.pdf

Discussion

Shift towards real-world benchmarks

Missing labels sufficiently addressed

Allowing label noise might pay off

Pixel-level information potentially not necessary

Questions?

Thank you for your attention!

References

[1] J. Li, R. Socher, and S. C. H. Hoi, “DivideMix: Learning with Noisy Labels as Semi-supervised Learning,” *arXiv:2002.07394*, 2020.

[2] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” *arXiv preprint arXiv:2010.01412*, 2020.

[3] Z. Xiaojin and G. Zoubin, “Learning from labeled and unlabeled data with label propagation,” Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University, 2002.

[4] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, “Learning to propagate labels: Transductive propagation network for few-shot learning,” *arXiv:1805.10002*, 2018.

[5] T.Chen,S.Kornblith,M.Norouzi,andG.Hinton,“Asimpleframework for contrastive learning of visual representations,” in Proc. ICML, 2020, pp. 1597–1607.

References

[6] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assign- ments,” arXiv:2006.09882, 2021.

[7] L. Ye, Z. Liu, and Y. Wang, “Learning semantic segmentation with diverse supervision,” 2018.

[8] A. Kolesnikov and C. H. Lampert, “Seed, expand and constrain: Three principles for weakly-supervised image segmentation,” in European conference on computer vision. Springer, 2016, pp. 695–711.

[9] Y. Wei, X. Liang, Y. Chen, X. Shen, M. Cheng, J. Feng, Y. Zhao, and S. Yan, “Stc: A simple to complex framework for weakly-supervised semantic segmentation,” IEEE TPAMI, vol. 39, no. 11, pp. 2314–2320, 2017.

[10] X. Wang, S. Liu, H. Ma, and M.-H. Yang, “Weakly-supervised semantic segmentation by iterative affinity learning,” Int J Comput Vis, Jun. 2020. [Online]. Available: https://doi.org/10.1007/s11263- 020- 01293-3