Master degree in Computer Engineering
Academic Year: 2023/2024
Course: 01URWOV - Advanced Machine Learning
Professor: Tatiana Tommasi
TA: Leonardo Iurada

DOMAIN ADAPTATION VIA ACTIVATION SHAPING

https://github.com/Truvella99/Activation-Shaping-AML

Presented by:
Freni Davide Giovanni (s305571)
Gagliardo Domenico (s310454)
Gaetano Roberto (s318989)

Politecnico di Torino

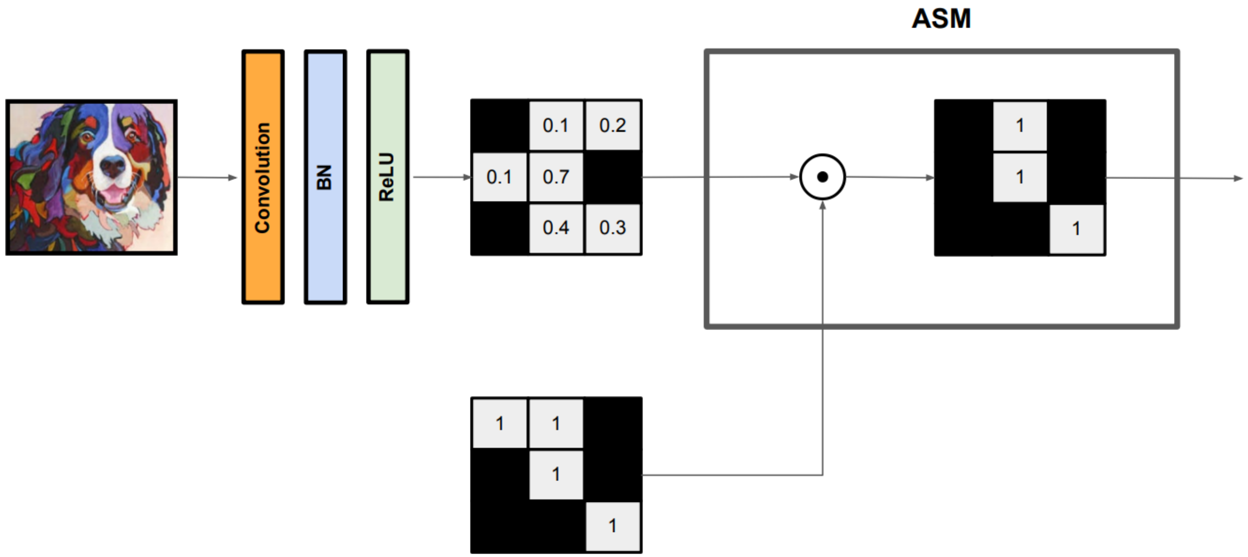
Problem Introduction

- Domain Shift problem.
- Different data distribution can be encountered at test time.
- Formalized with the setting of Unsupervised Domain Adaptation.

PACS DATASET

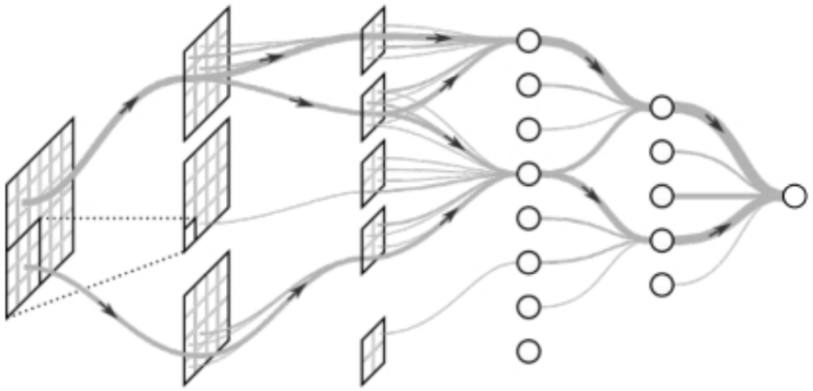
Activation Shaping Module (ASM)

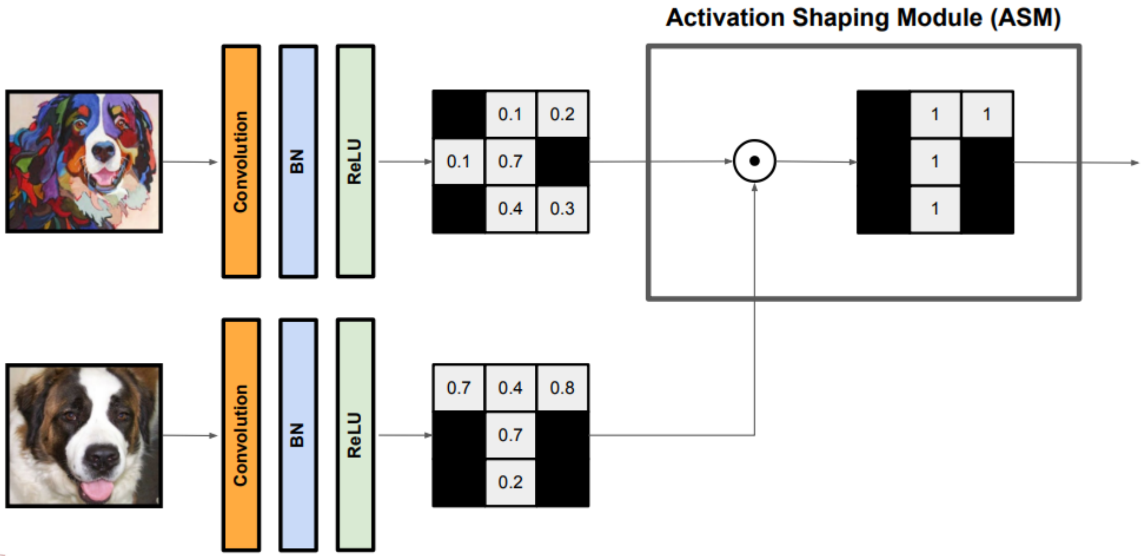
- The goal is to obtain a model that works well on the target distribution.
- Through the ASM it is possible to modify the activation maps.
- Both user defined and learned rules can be exploited.



ASM: The Idea

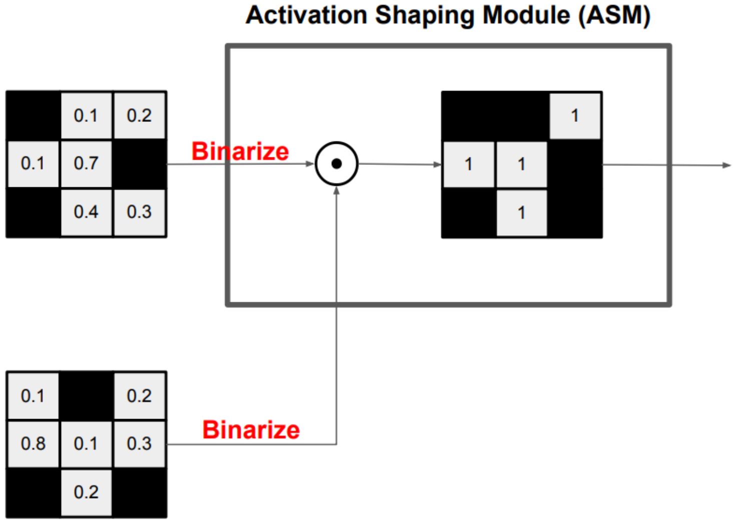
- In the network there could be content-specific and style-specific paths.
- Question: can we discard style-specific paths in order to retain only content-specific paths, which are the ones that result useful to us to accomplish correctly (and with a higher accuracy) our classification task?

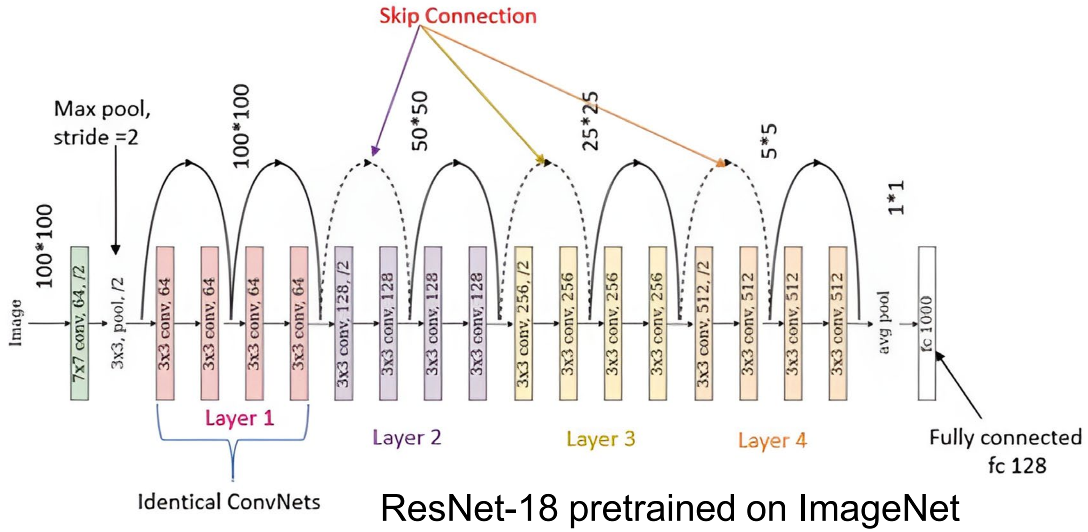




Approach

- Custom ASM designed to receive two activation maps as input.
- ASM initially integrated on the macrolayers and then on the inner layers.
- We tried multiple combinations (all layers, every 2, every 3, etc.) with poor results. The best ones are those with ASH applied on the single layers.



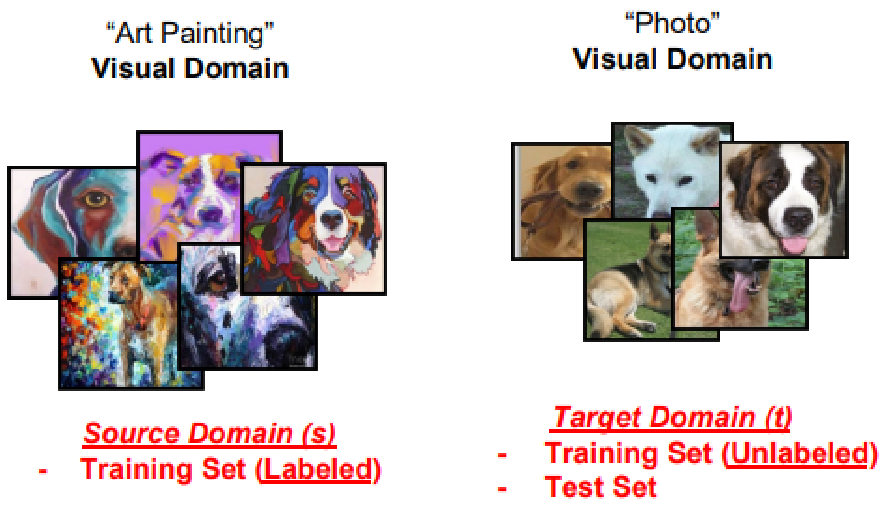


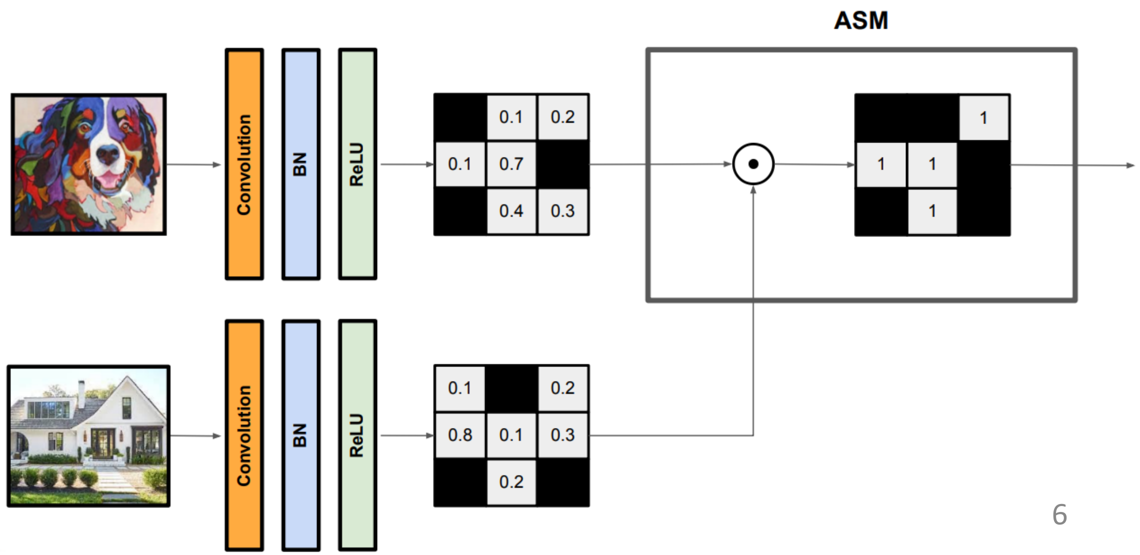
ResNet-18 pretrained on ImageNet

Unsupervised Domain Adaptation (UDA)

- In UDA, there is a labeled Source domain (Training Set) and an unlabeled Target domain (Test Set).

- Goal: training a model on the source domain to achieve good performance on data of the target domain.

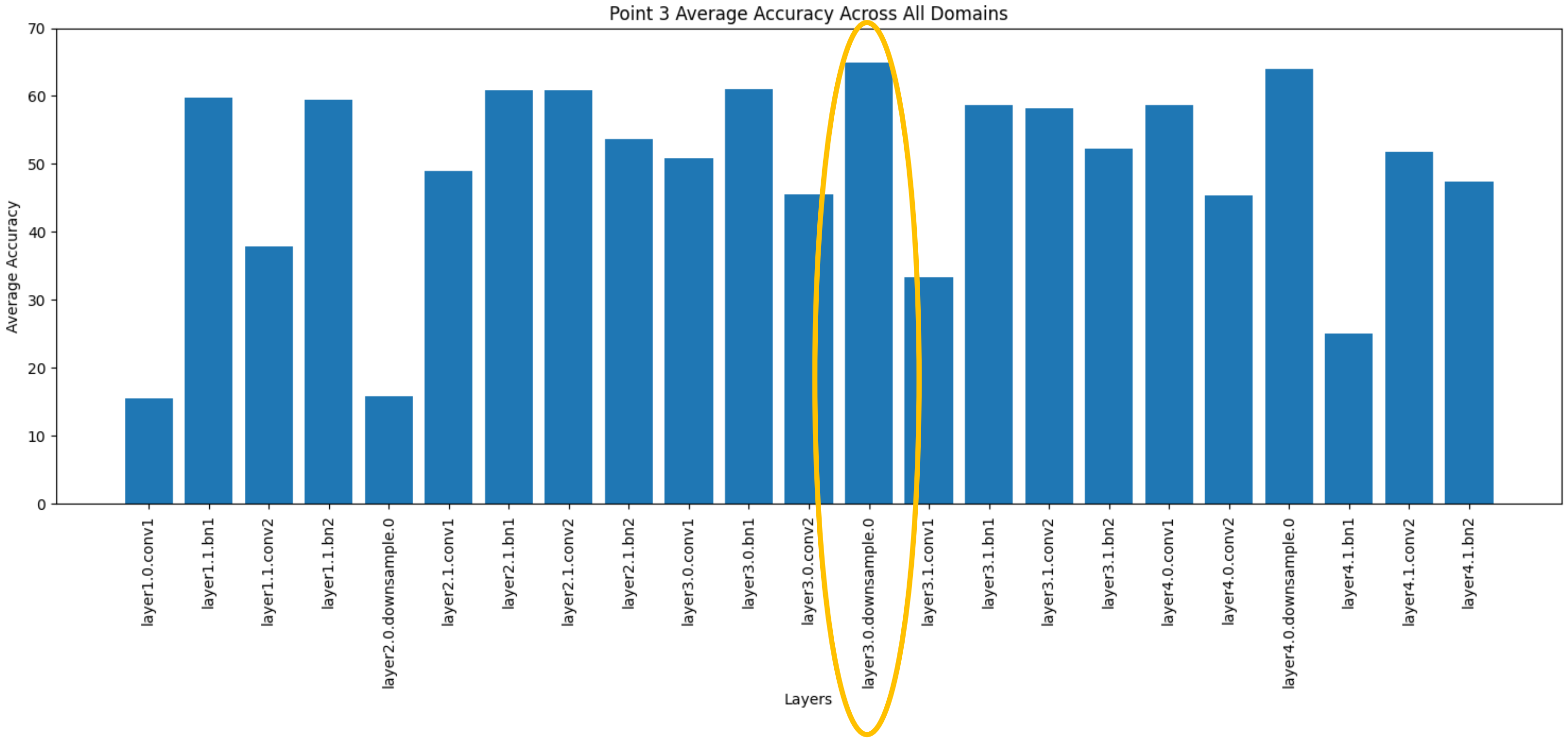




Unsupervised Domain Adaptation (UDA)

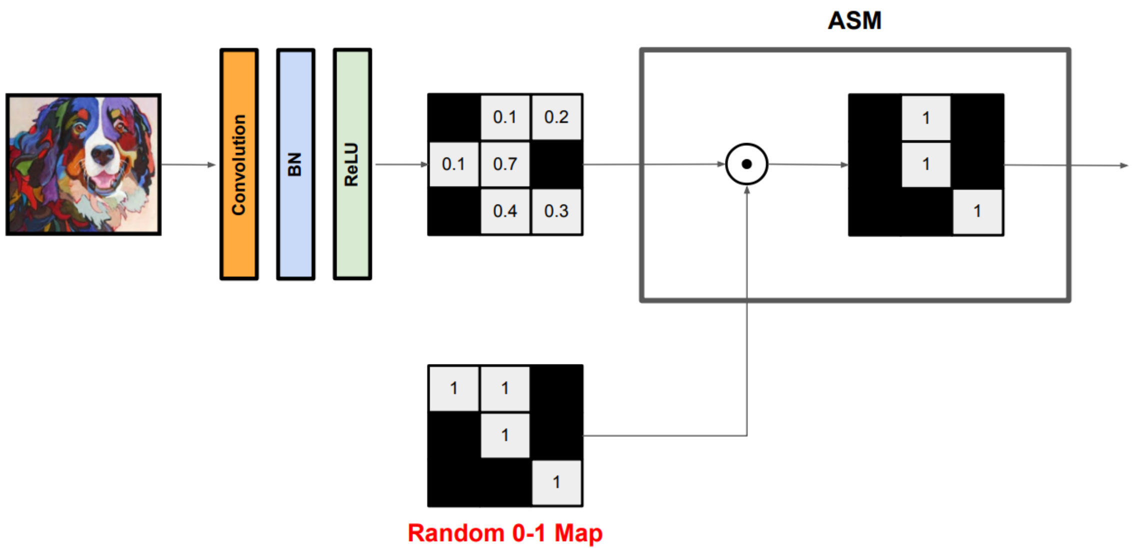
Baseline	Average Accuracy Across All Domains
	63.63%

- Inserting the ASM in the middle seems to lead to improvements (64.85 as average accuracy).

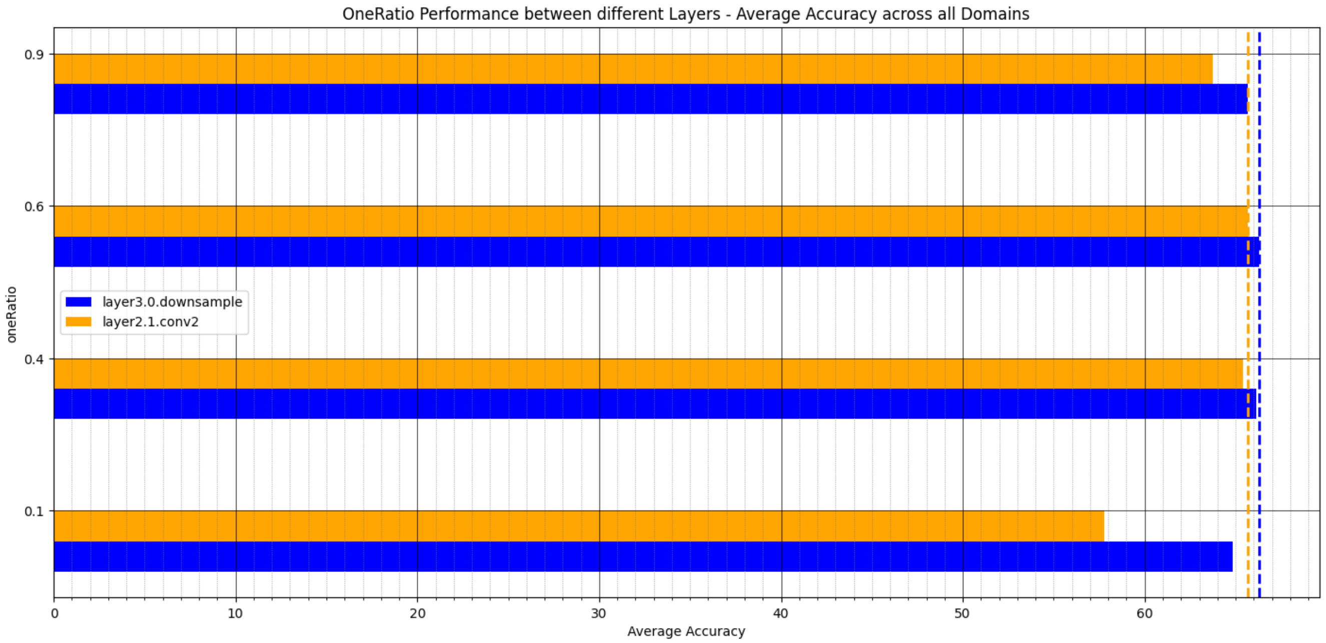


Random Activation Maps (RAM)

- The idea is to make so that each matrix M is a random mask of zeros and ones.
- Matrix A still comes from our Source Domain.
- Goal: deactivate some of the elements of the main activations.



Random Activation Maps (RAM)

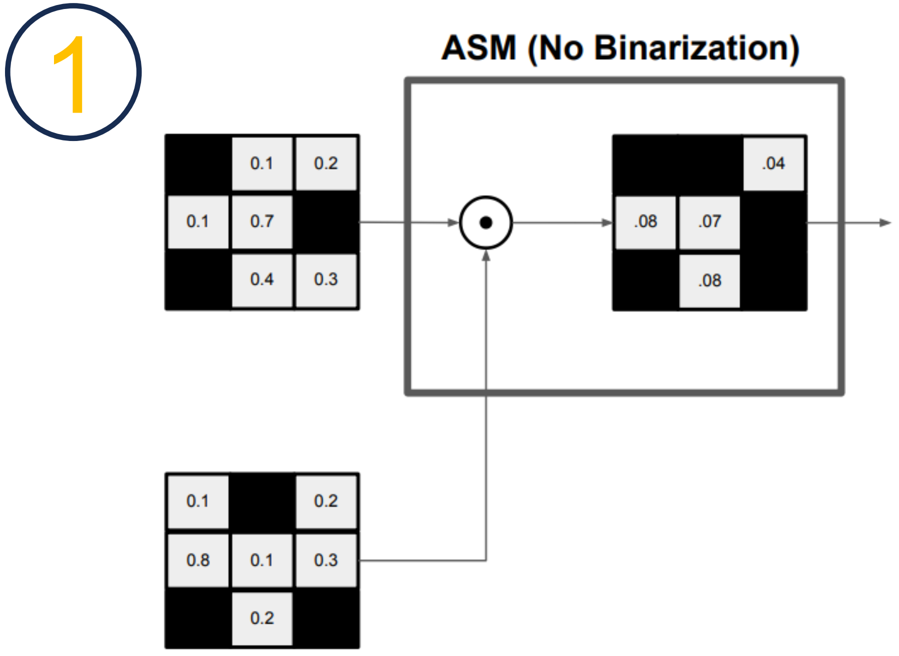


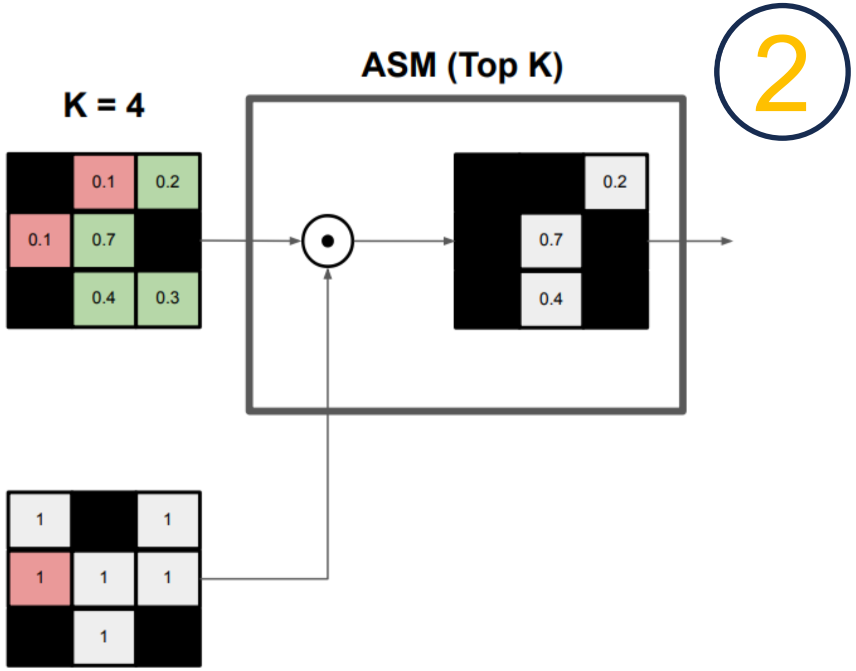
- Good results have been obtained with layer3.0.downsample.0 (66.30 of average accuracy across the three Target Domains with one ratio set to 0.6).
- Performances get slightly worse when we prune too much or too little.

Binarization Ablations

- It may be that binarization distorts the feature representation too much and alter the network behaviour in an undesired way.

- UDA and Random Activation Maps have been repeated with 2 variations of the ASH.

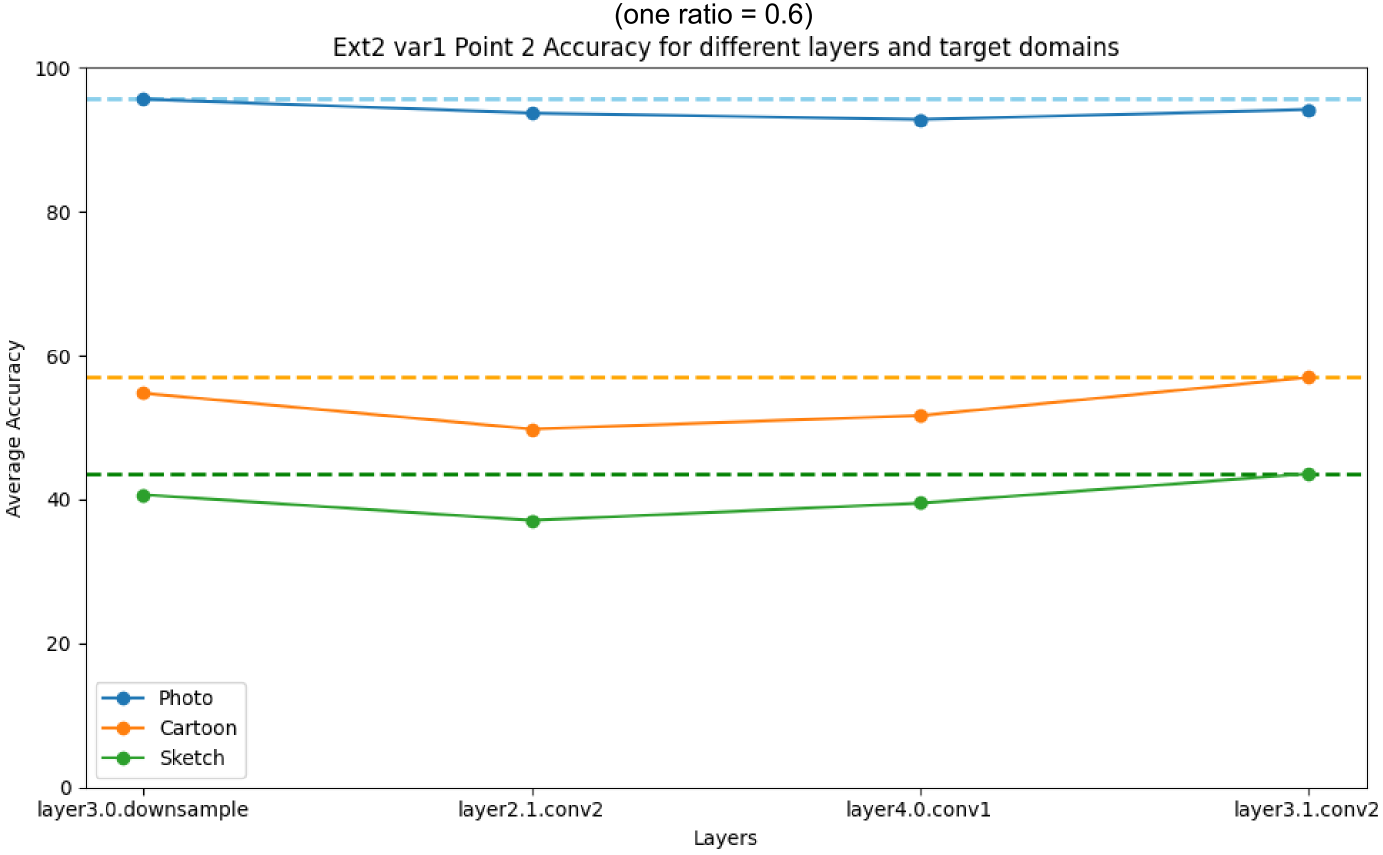




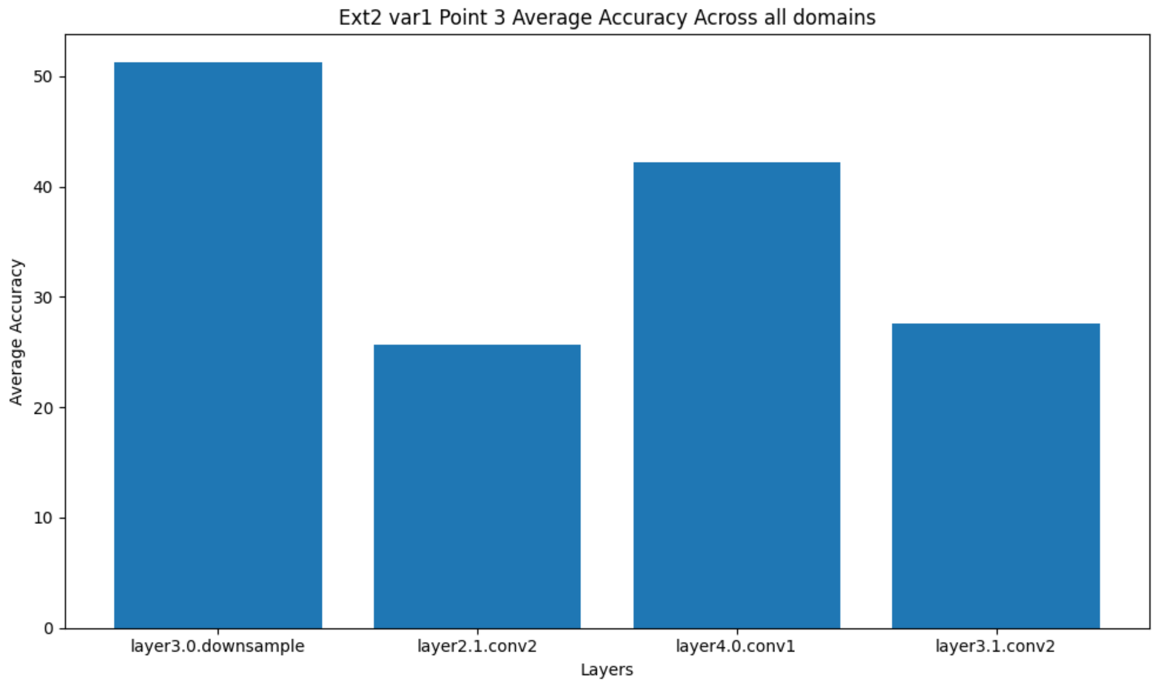
Binarization Ablations (1st variant - RAM)

- A slight improvement with respect to the baseline has been obtained with one ratio = 0.6 on layer3.1.conv2, which gave us 64.90% of accuracy.

- Still a worst result than the one obtained with binarization for layer3.0.downsample.0 with a one ratio equal to 0.6 that performed an accuracy of 66.30 %



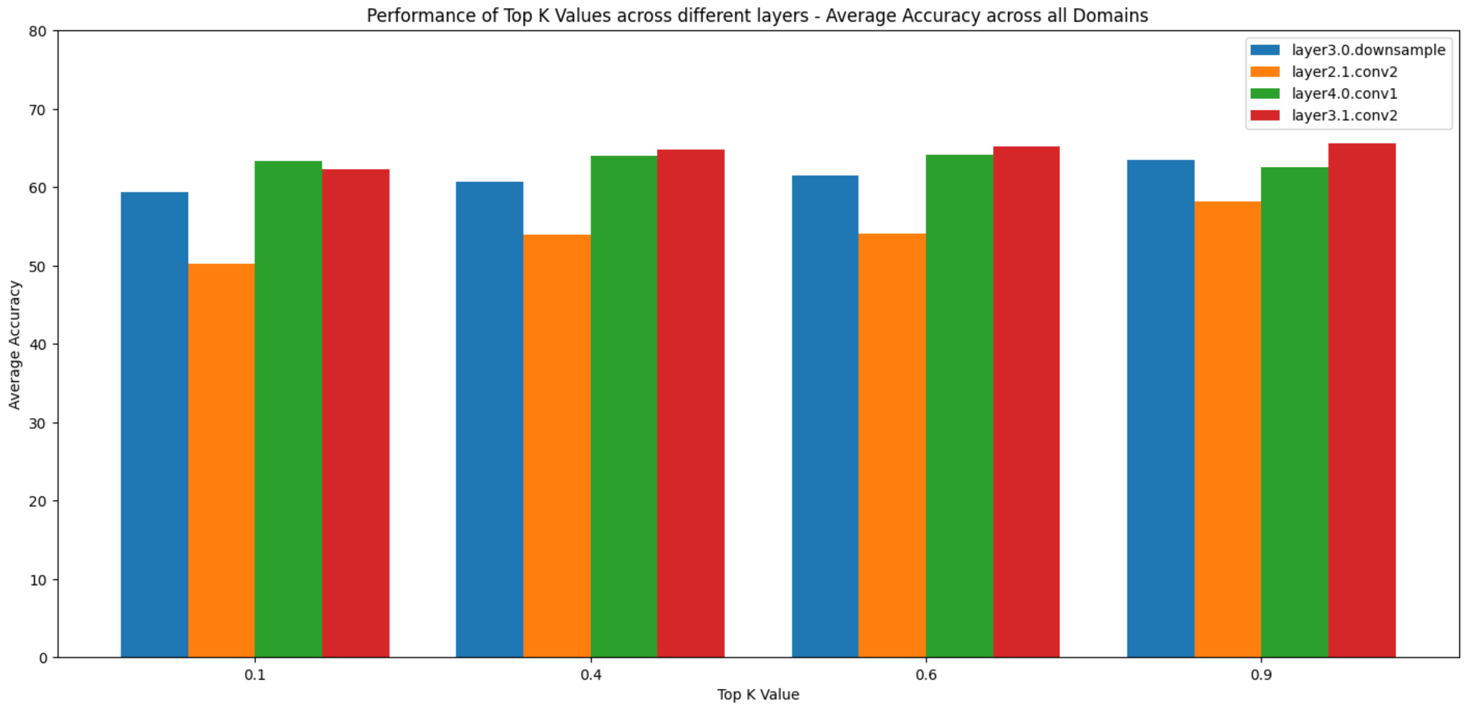
Binarization Ablations (1st variant - UDA)



- In UDA, instead, not binarizing led to a significant accuracy worsening for all the three target domains.
- So, binarization is in some way helping the network.

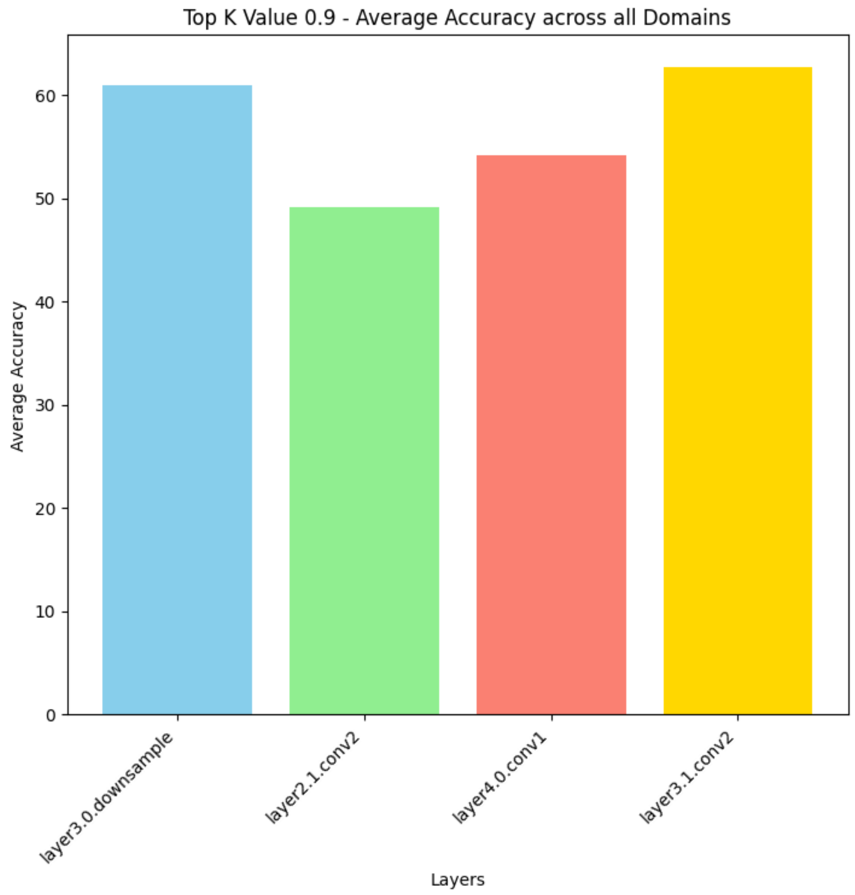
Binarization Ablations (2nd variant - UDA)

- Masking the main activation map retaining only the top K values.
- K is an hyperparameter that we tuned opportunely.
- K is expressed as a percentage and indicates how many of the highest values of the map are not turned off.
- Testing several several values of K (0.1, 0.4, 0.6, and 0.9) across various layers



Binarization Ablations (2nd variant - RAM)

- Higher values of K generally behaves better, with the best results obtained for K = 0.9 for both RAM and UDA.
- The layer that worked better has been layer3.1.conv2, followed by layer3.0.downsample.0.
- Still not able to overcome layer3.0.downsample.0 on vanilla RAM with an accuracy of 66.30 %



To sum up (RAM)

- In general, binarization could be beneficial, but things may change according to the layer on which we are reasoning on.

- Better improvements obtained by applying binarization.

- Ignoring few elements using top-K can lead to good results.

Baseline	Average Accuracy Across All Domains
	63.63%

	ONE RATIO = 60%		
Source Domain	**Average Accuracy Across All Domains**		
Art Painting	**Pt. 2**	**Ext.2 Var.1 Pt.2**	**Ext.2 Var.2 Pt.2 (TOP-K = 90%)**
layer3.0.downsample.0	66.30 %	63.69 %	63.44 %
layer2.1.conv2	65.67 %	60.20 %	58.23 %
layer4.0.conv1	48.32 %	61.32 %	62.14 %
layer3.1.conv2	53.38 %	64.90 %	65.57 %

To sum up (UDA)

Baseline	Average Accuracy Across All Domains
	63.63%

Source Domain	Average Accuracy Across All Domains		
Art Painting	Pt. 3	Ext.2 Var.1 Pt.3	Ext.2 Var.2 Pt.3
			TOP-K = 90%
layer3.0.downsample.0	64.85 %	51.23 %	60.96 %
layer2.1.conv2	60.93 %	25.70 %	49.15 %
layer4.0.conv1	58.62 %	42.21 %	54.22 %
layer3.1.conv2	58.27 %	27.63 %	62.67 %

- Binarization even more helpful.
- Top-k does not produce satisfying results.
- Also in this case, best result obtained for layer3.0.downsample.0.

Conclusions

- UDA leads to good results:
  some style-specific paths have been discarded.

- RAM performs slightly better: randomness prevents overfitting.

- Binarization helps the network in giving more importance to features that do not discriminate between source and target domains.

References

- [1] Github project repository. https://github.com/iurada/Activation-Shaping-AML
- [2] Stanford university website. https://web.stanford.edu/~nanbhas/blog/forward-hooks-pytorch/
- [3] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for outof-distribution detection. ICLR, 2023. https://arxiv.org/pdf/1710.03077.pdf
- [4] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Francois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. JMLR, 17(1):2096–2030, 2016. https://arxiv.org/pdf/1505.07818.pdf
- [5] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. CoRR, 2017. https://arxiv.org/pdf/2209.09858.pdf

Master degree in Computer Engineering
Academic Year: 2023/2024
Course: 01URWOV - Advanced Machine Learning
Professor: Tatiana Tommasi
TA: Leonardo Iurada

Thank you for your attention!

Presented by:
Freni Davide Giovanni (s305571)
Gagliardo Domenico (s310454)
Gaetano Roberto (s318989)

Politecnico di Torino
1859