Open-Set Domain Adaptation through Self-Supervision

Protopapa Andrea, Quarta Matteo, Ruggeri Giuseppe, Versace Alessandro
Politecnico di Torino
Italy
{s286302,s292477,s292459,s292477}@studenti.polito.it <- Riordinare

Abstract

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodi consequatur. Duis aute irure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1. Notes

1.1. Language

All manuscripts must be in English.

1.2. Paper length

Papers, excluding the references section, must be no longer than eight pages in length. The references section will not be included in the page count, and there is no limit on the length of the references section.

1.3. Math

Please number all of your sections and displayed equations as in these examples:

$$E = m \cdot c^2 \tag{1}$$

and

$$v = a \cdot t. \tag{2}$$

All authors will benefit from reading Mermin's description of how to write mathematics: http://www.pamitc.org/documents/mermin.pdf.

1.4. Blind View

Blind review means that you do not use the words "my" or "our" when citing previous work.

Saying "this builds on the work of Lucy Smith [1]" does not say that you are Lucy Smith; it says that you are building on her work. If you are Smith and Jones, do not say "as we show in [7]", say "as Smith and Jones show in [7]" and at the end of the paper, include reference 7 as you would any other cited work.

An example of a bad paper just asking to be rejected:

> An analysis of the frobnicatable foo filter.
>
> In this paper we present a performance analysis of our previous paper [1], and show it to be inferior to all previously known methods. Why the previous paper was accepted without this analysis is beyond me.
>
> [1] Removed for blind review

An example of an acceptable paper:

> An analysis of the frobnicatable foo filter.
>
> In this paper we present a performance analysis of the paper of Smith *et al.* [1], and show it to be inferior to all previously known methods. Why the previous paper was accepted without this analysis is beyond me.
>
> [1] Smith, L and Jones, C. "The frobnicatable foo filter, a fundamental contribution to human knowledge". Nature 381(12), 1-213.

Finally, you may feel you need to tell the reader that more details can be found elsewhere, and refer them to a technical report. For conference submissions, the paper must stand on its own, and not *require* the reviewer to go to a tech report for further details. Thus, you may say in the body of the paper "further details may be found in [**?**]". Then submit the tech report as supplemental material. Again, you may not assume the reviewers will read this material.

1.5. Caption Example

1.6. Miscellaneous: e.g., et al.

The space after *e.g.*, meaning "for example", should not be a sentence-ending space. So *e.g.* is correct, *e.g.* is not. The provided `\eg` macro takes care of this.

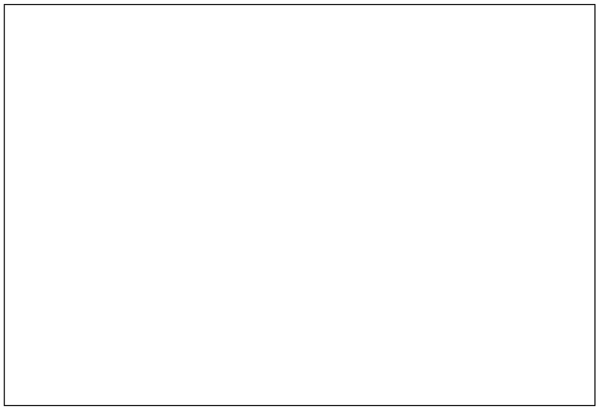


Figure 1. Example of caption. It is set in Roman so that mathematics (always set in Roman: $B \sin A = A \sin B$) may be included without an ugly clash.

When citing a multi-author paper, you may save space by using "et alia", shortened to "*et al.*" (not "*et. al.*" as "*et*" is a complete word). If you use the `\etal` macro provided, then you need not worry about double periods when used at the end of a sentence as in Alpher *et al.* However, use it only when there are three or more authors.

1.7. Sub-Figures examples

1.8. Formatting

All text must be in a two-column format.

1.9. Footnotes

Please use footnotes[1] sparingly. Indeed, try to avoid footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence). If you wish to use a footnote, place it at the bottom of the column on the page on which it is referenced. Use Times 8-point type, single-spaced.

1.10. Cross-references

For the benefit of author(s) and readers, please use the

```
\cref{...}
```

command for cross-referencing to figures, tables, equations, or sections. This will automatically insert the appropriate label alongside the cross-reference as in this example:

> To see how our method outperforms previous work, please see Fig. 1 and Tab. 1. It is also possible to refer to multiple targets as once, *e.g*. to Figs. 1 and 2a. You may also return to **??** or look at Eq. (2).

Method	Frobnability
Theirs	Frumpy
Yours	Frobbly
Ours	Makes one's heart Frob

Table 1. Results. Ours is better.

If you do not wish to abbreviate the label, for example at the beginning of the sentence, you can use the

```
\Cref{...}
```

command. Here is an example:

> Figure 1 is also quite important.

1.11. Caption References

List and number all bibliographical references at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [**?**]. Where appropriate, include page numbers and the name(s) of editors of referenced books. When you cite multiple papers at once, please make sure that you cite them in numerical order like this [**?**, **?**, **?**, **?**, **?**]. If you use the template as advised, this will be taken care of automatically.

1.12. Table Example

1.13. Cenetring Graphics

All graphics should be centered. In LaTeX, avoid using the `center` environment for this purpose, as this adds potentially unwanted whitespace. Instead use

```
\centering
```

at the beginning of your figure.

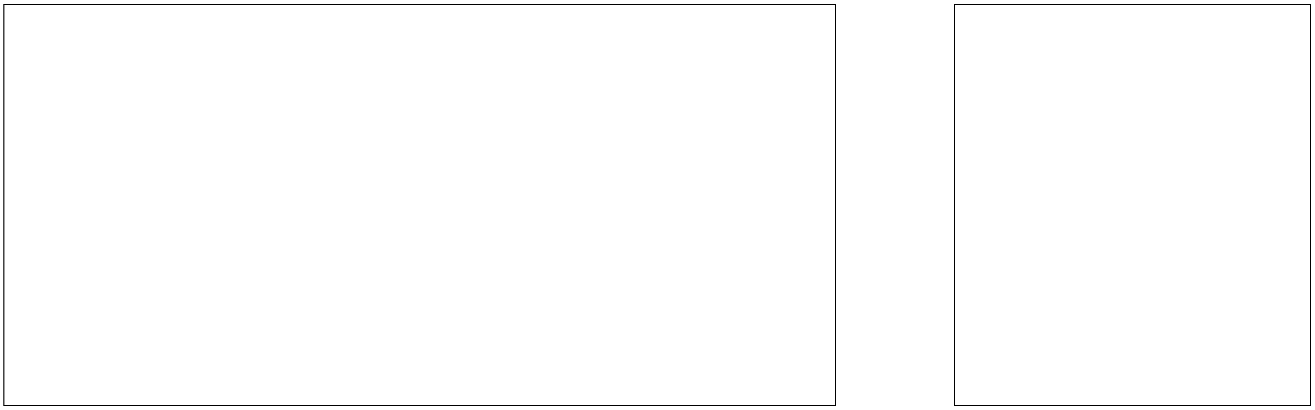
When placing figures in LaTeX, it's almost always best to use `\includegraphics`, and to specify the figure width as a multiple of the line width as in the example below

```
\usepackage{graphicx} ...
\includegraphics[width=0.8\linewidth]
                {myfile.pdf}
```

1.14. Color

If you use color in your plots, please keep in mind that a significant subset of reviewers and readers may have a color vision deficiency; red-green blindness is the most frequent kind. Hence avoid relying only on color as the discriminative feature in plots (such as red *vs*. green lines), but add a second discriminative feature to ease disambiguation.

[1] This is what a footnote looks like. It often distracts the reader from the main flow of the argument.



(a) An example of a subfigure.

(b) Another example of a subfigure.

Figure 2. Example of a short caption, which should be centered.

2. Introduction

Classical Machine Learning in the past years have made some oversimplified assumptions actually detached from the usage of Artificial Intelligent systems in everyday real world and the problems they bring.

The firt assumption is that the training and test sets come from the same distributions. Therefore, a model learned from the labeled training data is expected to perform well on the test data. However, this assumption may not always hold in real-world applications where the training and the test data fall from different distributions and in this case naively applying the trained model on the new dataset may cause degradation in the performance. To solve this problem we can make use of Domain Adaptation, where our goal is to train a neural network on a source dataset for which labels are available, and test a good accuracy on the target dataset, which is related but significantly different from the source dataset, and whose label or annotation is not available. Generally it seeks to learn a model from a source labeled data that can be generalized to a target domain by minimizing the difference between domain distributions and enforcing feature extractor to extract similar features for source and target domain images. As underlined in [5], feature-based adaptation approaches aim to map the source data into the target data by learning a transformation that extracts invariant feature representation across domains, transforming the original features into a new feature space and then minimize the gap between domains in the new representation space in an optimization procedure, while preserving the underlying structure of the original data.

Secondly, in real-world recognition/classification tasks it is usually difficult to collect training samples to exhaust all classes when training a recognizer or classifier. A more realistic scenario is open set recognition (OSR) [6], where incomplete knowledge of the world exists at training time, and unknown classes can be submitted to an algorithm during testing, requiring the classifiers to not only accurately classify the seen classes but also effectively deal with unseen ones, which otherwise drastically weakens the robustness of the methods. On the contrary this system should reject unknown/unseen classes at test time and separate the known and unknown samples. As underlined by [13], existing open-set classifiers rely on deep networks trained in a supervised manner on known classes in the training set; this causes specialization of learned representations to known classes and makes it hard to distinguish unknowns from knowns.

To solve these significant issues this method is focused on a self-supervised task. Self-supervised Learning is an unsupervised learning method where the supervised learning task is created out of the unlabelled input data. This task could be as simple as given the upper-half of the image, predict the lower-half of the same image. Supervised learning requires usually a lot of labelled data and getting good quality labelled data is an expensive and time-consuming task. On the other hand, the unlabelled data is readily available in abundance. The fundamental idea for self-supervised learning is to create some auxiliary pre-text task for the model from the input data itself such that while solving the auxiliary task, the model learns the underlying structure of the data (for instance the high-level knowledge, correlations, metadata embedded in the data). This type of learning was recently used for Domain Adaptation, learning robust cross-domain features and supporting generalization [4, 12], and also for some Open Set problems specialized for anomaly detection and discriminate normal and anomalous data [2, 8].

The approach presented in this paper brings these topics together in the so called Open-Set Domain Adaptation (OSDA) problem. A two-stage method is hence proposed, aiming to identify and isolate unknown class samples in the first stage, before reducing in the second stage the domain shift between the source domain and the knwon target domain to avoid negative transfer. All this is done in both stages using a modified version of the rotation task as self-supervised model, predicting the relative rotation between an image and its rotated version. Finally a classifier is used to predict if each target sample belongs either to one of the knwon classes or to an unknown class, being in the latter case rejected.

The method was evaluated on the Office-Home benchmark [**?**] with a specific OSDA metric.

ADD HERE RESULTS AND A BRIEF OF CONCLUSIVE IDEAS (ALSO POSSIBLE FUTURE WORKS)!!

3. Related Work

4. Method

4.1. Problem Formulation

Our starting point is the source dataset, defined as $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_S} \sim p_s$, where each element $\mathbf{x}_i^s$ belonging to any y_i^s is a sample from the source domain S. This dataset has a target counterpart, $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t} \sim p_t$ which is unlabeled. In OSDA we have that $p_s \neq p_t$. The source dataset $\mathcal{D}_s$ is associated with a set of known classes, $\mathcal{C}_s$, which can also be found in the target dataset $\mathcal{D}_t$, but is supposedly smaller. Hence we have that $|\mathcal{C}_s| < |\mathcal{C}_t|$ and that $\mathcal{C}_s \subset \mathcal{C}_t$. In a setting of domain adaptation, we further have that $p_t^s \neq p_s$, the target distribution of the known source classes A metric for measuring how different two domain are is the openness betweeen source and target domain [1], defined as $\mathbb{O} = 1 - \dfrac{\mathcal{C}_s}{\mathcal{C}_t}$. When $\mathbb{O} > 0$, we're dealing with an OSDA problem.

4.2. Approach

To tackle the task, we chose to split it in two different steps. First we have to train the model to separate between the known classes ($\mathcal{C}_s$) and the unknown classes ($\mathcal{C}_{t \setminus s}$) in a reliable enough way. This is achieved by using a semi-supervised task, by training to model to both recognize a sample class and its correct orientation. The second step is similar to a classic CSDA problem, where we train the model on a union of both source and target datasets.

4.3. Rotation Recognition

Let's denote with $rot(\mathbf{x}, i)$ the rotation of the sample image $\mathbf{x}$ by $i \times 90$ degrees clockwise. This is the self-supervised part of our proposed model as rotations $i \in [0, 3]$ can be randomly generated and then predicted. To avoid situations where the objective orientation of a sample would be a too complicated task, even for a human being, we also feed in input to the model the un-rotated image features. Alternatively, instead to have the model predict the rotation of a sample for any class, we also try using a different head for each one of the known classes $\mathcal{C}_s$, along with different loss functions.

4.4. Step I: Sample Separation

To separate samples we train the model on an enhanced version of $\mathcal{D}_s$, $\tilde{\mathcal{D}}_s = \{(\mathbf{x}_i^s, \tilde{\mathbf{x}}_i^s, z_i^s)\}_{i=1}^{N_s}$ where $\tilde{\mathbf{x}}_i^s$ is the rotated version of $\mathbf{x}_i^s$ and z_i^s is the rotation index associated to image i. The network is composed by an extractor E and two heads, R_1 and C_1 in its standard form. When using a multi-head rotation predictor, it is composed of $|\mathcal{C}_s| + 1$ heads for the rotation task, and an additional one for the classification task. For the roation index prediction, we both use the single-head and multi-head architecture. The features of both original and rotated image are used to predict the rotation as $\tilde{\mathbf{z}}_s = softmax(R_1([E(\mathbf{x}^s), E(\tilde{\mathbf{x}}^s)]))$ in the single-head case and using the j-th head as $\tilde{\mathbf{z}}_s = softmax(R_{1,j}[E(\mathbf{x}^s), E(\tilde{\mathbf{x}}^s)])$ in the multi-head case. Classes are predicted only using un-rotated features as $\tilde{\mathbf{y}}^s = softmax(C_1(E(\mathbf{x}^s)))$. The model is training by minimizing an objective function defined as $\mathcal{L}_1 = \mathcal{L}_{C_1} + \mathcal{L}_{R_1}$. This is the sum of two cross-entropy loss functions as in:

$$\mathcal{L}_{C_1} = -\sum_{i \in \mathcal{D}_s} \mathbf{y}_i^s \log \tilde{\mathbf{y}}_i^s \tag{3}$$

$$\mathcal{L}_{R_1} = -\alpha_1 \sum_{i \in \tilde{\mathcal{D}}_s} \mathbf{y}_i^s \log \tilde{\mathbf{z}}_i^s \tag{4}$$

Where α_1 is a weight associated to the rotation task. We also try using an extended rotation objective function $\mathcal{L}_{R_1}^*$ also implementing a center loss function [11]:

$$\mathcal{L}_{R_1}^* = -\alpha_1 \sum_{i \in \mathcal{D}_s} \mathbf{y}_i^s \log \tilde{\mathbf{z}}_i^s + \lambda_1 ||\mathbf{v}_i^s - \gamma(\mathbf{z}_i^s)||_2^2 \tag{5}$$

Here v_i is the penultimate layer of the rotation classifier, called *features*, and $\gamma(\mathbf{z}_i)$ gives the center of the features $\mathbf{v}_i$ associated to class i and $|| \cdot ||_2^2$ is the l-2 norm and λ_1 is the weight associated with this extension of the loss function.

When training is completed, we can start separating samples. To do so, we get the normality score $\mathcal{N}(\cdot)$ which is defined as the maximum prediction of the rotation classifier, $\mathcal{N}(\tilde{\mathbf{x}}_i^s) = \max(\tilde{\mathbf{z}}_i^t)$. To tell wheter a sample belongs to the known samples of the target domain $\mathcal{D}_t^{knw}$ or the unknown one $\mathcal{D}_t^{unk}$ requires choosing a threshold $\tilde{\mathcal{N}}$. Then we can simply separate as:

$$\begin{cases} \tilde{\mathbf{x}}_i^t \in \mathcal{D}_t^{knw} & \text{if } \mathcal{N}(\tilde{\mathbf{x}}_i^s) \geq \tilde{\mathcal{N}} \\ \tilde{\mathbf{x}}_i^t \in \mathcal{D}_t^{unk} & \text{if } \mathcal{N}(\tilde{\mathbf{x}}_i^s) < \tilde{\mathcal{N}} \end{cases} \tag{6}$$

When employing a multi-head architecture, we need to choose which one of the $|\mathcal{C}_s|$ heads to use for the classification task. Head $R_{1,j}$ is used by picking j as $j = \arg\max_j \tilde{\mathbf{y}}_i^t$.

4.5. Performance Metric

Evaluating model perfomance requires finding a balance between two metrics: the first one is *OS**, the share of correctly classified samples; the second one is *UNK*, the share of correclty rejected samples. One common problem is a model never confident enough to reject a sample as unknown, thus possibly achieving high *OS** scores but near-zero *UNK*, and the opposite one, a model rejecting every given sample achieving perfect *UNK* but zero *OS** scores. To compare models in a robust manner, we picked an harmonic mean between *OS** and *UNK*, defined as $HOS = 2\frac{OS^* \times UNK}{OS^* + UNK}$. This type of mean is more biased towards the lowest of the two scores, resulting in a more severe evaluation of models.

5. Experiments

5.1. Dataset

Our model is tested on the *Office-Home* dataset [10], which features 65 classes of images over four different domains: Art (A), Clipart (C), Product (P) and Real World (R). We set the first 45 classes to be known while the remaining 20 are unknown. After each epoch performed during the domain alignment phase, a validation run is performed on the entire original target dataset. For each experiment, we report both the best achieved *HOS* as HOS_{Best} and the one achieved by the last epoch as HOS. As separation is crucial for the model effectiveness, we also report the computer AUROC score for the first part.

5.2. Results

All the 12 source-to-target $S \to T$ shift are considered. Results are reported separately for each possible mode of execution, Multi-Head ON/OFF and Center-Loss ON/OFF. Each mode has a specific configuration, please refer to **??**.

Single-Head, CE Loss				
Source	Target	AUC_{ROC}	HOS	HOS_{Best}
S	T	50%	30%	30%
S	T	50%	30%	30%
S	T	50%	30%	30%

Multi-Head, CE Loss				
Source	Target	AUC_{ROC}	HOS	HOS_{Best}
S	T	50%	30%	30%
S	T	50%	30%	30%
S	T	50%	30%	30%

Single-Head, CE+C Loss				
Source	Target	AUC_{ROC}	HOS	HOS_{Best}
S	T	50%	30%	30%
S	T	50%	30%	30%
S	T	50%	30%	30%

Multi-Head, CE+C Loss				
Source	Target	AUC_{ROC}	HOS	HOS_{Best}
S	T	50%	30%	30%
S	T	50%	30%	30%
S	T	50%	30%	30%

Table 2. Test Caption

6. Method - Write name of proposal approach

6.1. Subsection 1

6.2. Subsection 2

7. Experiments

7.1. Subsection 1 - Setup

7.2. Subsection 2 - Implementation Details

7.3. Subsection 3 - Results

8. Conclusions (and Future Work)

A. Appendices

Appendices are material that can be read, and include lemmas, formulas, proofs, and tables that are not critical to the reading and understanding of the paper. Appendices should be **uploaded as supplementary material** when submitting the paper for review. Upon acceptance, the appendices come after the references, as shown here.

LaTeX-specific details: Use `\appendix` before any appendix section to switch the section numbering over to letters.

B. Supplemental Material

Submissions may include [9] [11] [3] [7] [4] non-readable supplementary material used in the work and described in the paper. Any accompanying software and/or data should include licenses and documentation of research review as appropriate. Supplementary material may report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.

Nonetheless, supplementary material should be supplementary (rather than central) to the paper. **Submissions that misuse the supplementary material may be rejected without review.** Supplementary material may include explanations or details of proofs or derivations that do not fit into the paper, lists of features or feature templates, sample inputs and outputs for a system, pseudo-code or source code, and data. (Source code and data should be separate uploads, rather than part of the paper).

The paper should not rely on the supplementary material: while the paper may refer to and cite the supplementary material and the supplementary material will be available to the reviewers, they will not be asked to review the supplementary material.

References

[1] Abhijit Bendale and Terrance Boult. Towards open set deep networks, 2015. 4

[2] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data, 2020. 3

[3] Silvia Bucci, Mohammad Reza Loghmani, and Tatiana Tommasi. On the effectiveness of image rotation for open set domain adaptation, 2020. 5

[4] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles, 2019. 3, 5

[5] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation, 2020. 3

[6] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. 2021. 3

[7] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018. 5

[8] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. 2018. 3

[9] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations, 2018. 5

[10] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation, 2017. 5

[11] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition, 2016. 4, 5

[12] Jiaolong Xu, Liang Xiao, and Antonio M. Lopez. Self-supervised domain adaptation for computer vision tasks. 2019. 3

[13] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition, 2019. 3