Unsupervised Domain Adaptation Through Multi Task Recognition for RGB-D Object Recognition

1st Gaetano Salvatore Falco
Dipartimento di Automatica e Informatica
Politecnico di torino
Turin, Italy
gaetanosalvatore.falco@studenti.polito.it

2nd Kuerxi Gulisidan
Dipartimento di Automatica e Informatica
Politecnico di torino
Turin, Italy
kuerxi.gulisidan@studenti.polito.it

***Abstract*—Domain Adaptation helps to achieve state-of-the-art performance on computer vision tasks, yet it cannot easily be standardized for every domain shift.**

Recently, the emergence of depth sensors and cameras is helping to increase accuracy performance, but Unsupervised Domain Adaptation (DA) must be implemented in order to reduce the distribution shift that arise from using different cameras or different light conditions.

In this paper we propose a novel approach for alignment by learning to solve a self-supervised multi-labelled task using Decimal Encoding that helps the extraction of domain invariant features carrying relative and absolute information.

This method not only helps to improve accuracy on existing methods, but it is also straightforward to implement and easy to optimize.

All code is publicly available at: https://github.com/chowned/rgb-d_uda_multi_task_learning

***Index Terms*—Multiple task learning, Domain adaptation, Self supervised task, RGB-D data**

I. Introduction

Significant progress has been made over the years on Domain Adaptation and Image Recognition, but the emergence of Depth sensors brings completely new information that can and should be used.

It is straightforward to understand why it has been used in robotic systems, as they need to interact with objects and surroundings.

But the depth information finds usage also in real-world scenarios and it is widely available on mobile phones or tablets (i.e. iPhone 12 pro or iPad pro) for taking pictures and for Augmented Reality Applications, and on Virtual Reality Headset that needs to track the movement of the person playing or the surroundings.

But, as a matter of fact, the cost for generating labelled data is an obstacle to the accuracy on target data as it is almost impossible to predict the target data at run time, as it depends on too many variables.

This real-world challenge is why Domain Adaptation is a very naive and popular approach, as it requires no manual intervention for labelling the data that are now treated as coming from two different distributions. Then, alignment is induced on the source and target domains via some self-supervised tasks that have been studied and proposed over the years including colorizing grayscale images [4], image jigsaw puzzles [5], etc.

In this paper, our main idea is a combination of the relative rotation classifier task and the flipping classifier task, which allows us to improve the results over the paper of Loghmani et al [1].

Our architecture then consists of a main task or label classifier and a pretext task which is responsible for helping the emergence of features that carry relative information (the relative rotation between the RGB image and the Depth image) and absolute information, as now the classifier should understand if both the RGB image and the Depth image are flipped independently.

By using this approach, we can also demonstrate that the deep learning architecture does not internally create a boundary to help with the decisions and, thus, we reduce overfitting.

While we do not propose a new pretext task for self-supervised domain adaptation, the key contribution of our paper is a new way of creating labels for data that bring information on multiple domains and a novel insight into features.

Additionally, we report state-of-the-art results when compared to the synthetic benchmark synROD→ ROD.

II. Related work

In this section, we provide a brief look on the recent technique on domain adaptation for RGB-D images and we bridge it with a work by Sun et al 2019 [2].

A. Unsupervised Domain Adaptation for RGB-D Images

Self-supervised tasks through rotation proved to be great tasks for a convolutional neural network to learn features from.

As we can find in the literature, this technique is very easy to implement and based on understanding the rotation of an image given 4 possible choices, 0°-90°-180°-270° [6].

Our approach is mainly based on the paper of Loghmani et al [1] for RGB-D data classification. Their work has been based on a simply but effective task, that is understanding the relative rotation between RGB and Depth of a given pair of images. This solves the problem of generating labels for a target domain and does not require human intervention.

While achieving state-of-the-art results, this approach leaves absolute information about the image and depends entirely on the relative information that is coming from the relative rotation.

In contrast, we implement a new self-supervised task that is able to understand the relative rotation but, in addition, it tries to understand if the RGB and/or Depth images are vertically flipped.

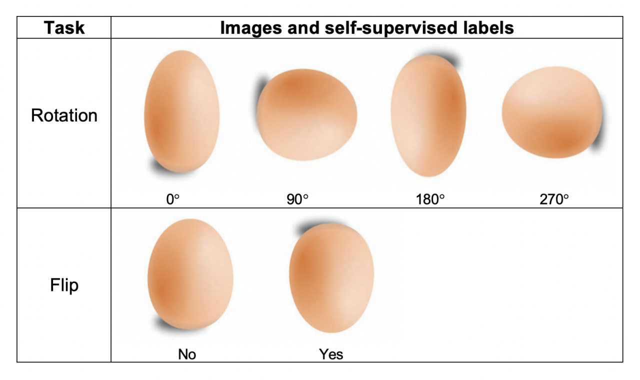

Fig. 1. Q:"By how much should the RGB image rotate with respect to the Depth image? Is the RGB Image flipped? Is the Depth Image flipped?" These questions are describing the decision boundary that the task is implementing. For simplicity, only absolute tasks are shown in the figure.

B. Self Supervised Tasks for Adaptation

Recently in RGB images, Unsupervised Domain Adaptation through self-supervision has been implemented with different tasks both to improve the accuracy of the model or to improve the robustness of the model to resist to attacks.

The concept of learning features for solving these tasks has given great results, but it can easily overfit or provide sub-optimal results if the input images are not suited for the given task.

As an example, we can find image colourization [4] that easily fails if light conditions are not optimal, the rotation task [6] fails if the object we want to classify is in different poses, and so on with the jigsaw puzzle task [5].

In contrast, we propose to jointly train the tasks as a single task, that is more robust to the geometry of the object we want to classify as shown in Figure 1.

C. Multitask Learning

There is no general approach for Domain Adaptation but, over the years, an increasing part of the literature has studied Multi-task Learning.

An idea related to ours is described by Sun et al. [2]. While their goal is quite similar (domain adaptation through separate tasks), the core implementation is different as they design multiple tasks running in parallel with the main task. More recently, model robustness has been addressed in Lawhon et al. [3], which shows that multitask learning is generating features that avoid over-fitting while ensuring robust information. Inspired by these successes, we implemented this approach for RGB-D data while proposing a novel way of reducing the number of necessary networks.

III. Method

In this section, we present our method for RGB-D DA with the following structure:

Section III-A provides a high-level overview of the method;

Section III-B describes the details of the pretext task;

Section III-C describes the Decimal Encoding used for generating labels;

Section III-D specifies the architecture of the CNN;

A. High-Level Overview

The goal of our approach is to induce alignment in the Distributions of the Source Domain and Target Domain. In order to do this, we have to keep in mind that we have labelled data for the source and unlabelled data for the target, thus we need to design an effective auxiliary self-supervised task that will be secondary to the main label predictor task.

While the main task is very straightforward, for the latter we designed a combination of relative and absolute tasks, meaning we predict the relative rotation between a pair of RGB and Depth images previously rotated and a prediction if the RGB and Depth images have been flipped independently.

The ground truth for the target data is then easily generated automatically from the data, we will discuss in section III-C the details of the implementation.

Learning to solve this auxiliary task improves the object class detection by generating domain-invariant features, that are now carrying both absolute information about the pair images and relative information.

B. Pretext task

Predicting image rotation is not only easy to implement and gives good results, as described by Gidaris et al. [6], but is also a robust method with higher performance when compared to other methods as we can find from Xu et al. [9].

However, choosing a single task can lead to sub-optimal results as maybe the data are not well suited. This is why, following the exciting approach of Sun et al.[2], we designed a joint pretext task consisting in:

- Rotating the Depth image by a multiple of 90°

- Rotating the RGB image by a multiple of 90°
- Vertical Flipping the Depth image
- Vertical Flipping the RGB image

In the end, the features must reconstruct all the above information, predicting the relative rotation applied to the input pair and if the RGB and Depth images have been flipped.

Horizontal Flipping was discarded as it is a Data Augmentation technique, typically used for building invariant features for natural scenes.

By applying this task, we overcome the problem if the object that we want to classify is in different poses thanks to the relative rotation, and improve the accuracy if the object appears almost in the same pose.

Algorithm 1 Algorithm for finding labels for multiple tasks

```
Data: Given N pretext tasks with 9 maximum labels, Given
      M classes for task 0;
Result: Calculate a priori the total number of classes for the
        labels;
numberLabels=0;
 digit=1;
 numberLabels+=M;
 while Task.HasNext do
    numberLabels=Task.getCurrentNumberClasses*digit
     digit=digit*10
end
```

C. Generating labels

We implemented a simple algorithm while importing the images to generate the ground truth for the target domain and the auxiliary task, for more details the labelling pseudocode is available in algorithm 1.

The catch is not to use multiple networks for multiple tasks but a single network and this already allows us to reduce the number of networks required (in our case, by 67% as we use 3 different tasks).

Given a pair of RGB and Depth images, the label is generated as follows:

- The relative rotation required to align the two images is calculated, its range is [0°-270°] or [1,4]
- The vertical flip on the RGB image, the range is [False,True] or [0,1]
- The vertical flip in the Depth image, its range is [False,True] or [0,1]

All of our labels are mutually non-exclusive, thanks to this we can borrow the main idea behind One-Hot Encoding to develop our algorithm.

In fact, this process of encoding is that every single state of a Finite State Machine can be labelled as a single bit of a binary number. As here we are working with decimal numbers and we are not required to completely discard our digits, we can implement labels on every digit of our task network classes, thus leading to the following result:

- Digit [1,2,3,4] for the relative rotation information, meaning [0°,90°,180°,270°]
- Digit [0,10] for the RGB flipping information, meaning [False,True]
- Digit [0,100] for the Depth flipping information, meaning [False,True]

This simple algorithm allows easy scalability for multiple tasks and does not require any additional networks.

It is important to note that different implementation can be used in order to reduce the dimension of the label. In this paper we present the easiest implementation, but we can use also an implementation based on the Huffman lossless encryption that is briefly explained here:

- classify tasks by number of labels and by likelihood. In our example, the relative rotation has a high likelihood to be classified and a high number of output classes compared to the flipping classifier
- sort tasks by likelihood and classes
- the next label for a different task is the next digit that doesn't intersect with other labels. In the case of our tasks, we have [1,2,3,4]-[5]-[10] instead of [1,2,3,4]-[10]-[100].

D. Network architecture

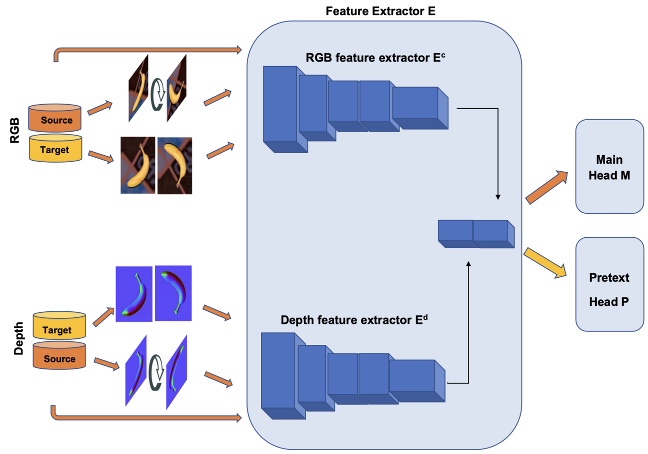

Fig. 2.

The network architecture used in this paper can be found in Figure 2.

Input images from source and target domains are processed and then given as input to the Feature Extractor. The output is then the input for the Label Predictor M and the pretext task P.

Each of these 3 modules is a standard neural network, thus allowing to use standard back-propagation through Stochastic Gradient Descent.

Feature Extractor E: in detail, we have two feature extractors built on the implementation of a Residual Neural Network with very few differences from the original implementation.

Main task M: this network is responsible for finding the ground truth for every feature among the 47 classes available in the dataset. It has been designed with an initial fully connected layer of 1000 neurons, followed by a Batch Normalization, using ReLU as an activation function before finishing with a final fully connected layer of the 47 output neurons.

Pretext task P: this network is responsible for solving the classification problem of predicting the label carrying absolute and relative information. It is defined with 2D convolutional layers, Batch Normalization and ReLU as activation functions.

IV. Experiment and results

In this section, we present the experimental protocol and the evaluation results of our method. More precisely:

Section IV-A describes the adopted dataset and comparison against Loghmani et al. paper results;

Section IV-B presents the implementation details for training the CNN;

Section IV-C shows quantitative and qualitative results on RGB-D DA.

A. Dataset and Comparison

The dataset that was used in this paper was the ROD dataset as it is the reference for RGB-D object recognition in robotic systems. It contains 41,877 pairs of RGB-D images of 300 objects grouped into 51 categories that are commonly found in house or in office.

synROD is a synthetic dataset generated from the models of the previous dataset, in this paper is used as the labelled source domain following the original implementation of our target paper [1].

In the previous work, the result that was accomplished was the new state of the art for this type of data. In the following section, we compared our paper directly with [1] and with the results obtained by disabling the domain adaptation, accomplished by setting the parameters for the rotation weight and entropy weight to 0.

B. Implementations details

The CNN is trained using SGD with momentum 0.9, learning rate 1×10^{-4}, batch size 64. We included entropy minimization with weight 0.1 but, on the contrary of [1], we lowered the weight decay to 0.04. The drop-out probability was left to 0.5 for all the networks that implement this method.

The Residual Networks used for feature extraction are pre-trained on the popular visual database ImageNet [10], thus allowing the initial weights to be initialized from it.

The rest of the network is initialized with Xavier initialization, and all parameters are updated during training with the above mentioned Stochastic Gradient Descent.

Following the procedure of [1], pre-processing is applied to the input images where the depth image follows a colourization with surface normal encoding.

Method	Source Domain	Difference in Percentage
Laghmani et al. [1]	66.72	0
Disabled Domain Adaptation	**68.62**	**+2.84**
Ours	66.63	-0.13

TABLE I

Method	Target Domain	Difference in Percentage
Laghmani et al. [1]	65.96	0
Disabled Domain Adaptation	36.58	-44.54
Ours	**69.14**	**+4.82**

TABLE II

C. Results

Table I and Table II presents the quantitative results of our method when compared to the Domain Adaptation presented in [1] and to the results with training only on source data.

Among the three proposed results, our approach yields significant improvement upon the state-of-the-art approach despite the simplicity of our method.

On the synROD dataset that has been chosen as the source domain, our approach does not give improved results compared to the approach of Laghmani et al. [1].

On our target domain dataset, instead, our work improves the target accuracy with a percentage gap of more than 4%, which we believe to be statistically significant.

Surprisingly, despite the baseline method providing a very high accuracy, our method further improves the accuracy over the target domain. As a matter of fact, while the state-of-the-art method learns to predict the most relevant pixels on the picture, we show that it could still benefit from another task of adaptation through self-supervision.

V. Conclusion

We hope that this paper encourages further research in this area, as there is much additional work in this field to be done due to the limited experiments that we were able to run.

The empirical result suggests that domain adaptation can be achieved with multi-task learning, and we also noted that no computational cost is required as there is no need to add more networks with the proposed method.

One disadvantage of our method, which is important to note, is that it achieves domain adaptation through tasks that bring alignment and not through measuring the gap. We leave this topic for future work, as we were able to implement a domain classifier as an additional task but not a Domain Adversarial Task.

References

[1] Mohammad Reza Loghmani, Luca Robbiano, Mirco Planamente, Kiru Park, Barbara Caputo and Markus Vincze, "Unsupervised Domain Adaptation through Inter-modal Rotation for RGB-D Object Recognition" in arXiv, 2020 (https://arxiv.org/pdf/2004.10016.pdf)

[2] Yu Sun, Eric Tzeng, Trevor Darrell, Alexei A. Efros, "UNSUPERVISED DOMAIN ADAPTATION THROUGH SELF-SUPERVISION" in arXiv, 2019 (https://arxiv.org/pdf/1909.11825.pdf)

[3] Matthew Lawhon, Chengzhi Mao & Junfeng Yan, "USING MULTIPLE SELF-SUPERVISED TASKS IM- PROVES MODEL ROBUSTNESS" in ICLR PAIR2Struct Worksho, 2022

[4] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in ECCV, pp. 649–666, Springer, 2016.

[5] M. Noroozi and P. Favaro, “Unsupervised learning of visual representions by solving jigsaw puzzles,” in ECCV, 2016.

[6] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations” in ICLR, 2018

[7] Chengzhi Mao, Amogh Gupta, Vikram Nitin, Baishakhi Ray, Shuran Song, Junfeng Yang, and Carl Vondrick. Multitask learning strengthens adversarial robustness. In European Conference on Computer Vision, pp. 158–174. Springer, 2020.

[8] Caruana, R. Multitask Learning. Machine Learning 28, 41–75 (1997). (https://doi.org/10.1023/A:1007379606734)

[9] J. Xu, L. Xiao, and A. M. Lpez, “Self-supervised domain adaptation for computer vision tasks,” IEEE Access, vol. 7, pp. 156 694–156 706, 2019.

[10] J. Deng, W. D. R. Socher, L. Li, K. Li, and F. Li, “Imagenet: A largescale hierarchical image database,” in CVPR, 2009, pp. 248–255.