ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

# Unsupervised domain adaptation of deep object detectors

Debjeet Majumdar [1] and Vinay P. Namboodiri[2]

Indian Institute of Technology, Kanpur - Computer Science and Engineering
Kalyanpur, Kanpur, Uttar Pradesh 208016 - India

**Abstract**. Domain adaptation has been understood and adopted in vision. Recently with the advent of deep learning there are a number of techniques that propose methods for deep learning based domain adaptation. However, the methods proposed have been used for adapting object classification techniques. In this paper, we solve for domain adaptation of object detection that is more commonly used. We adapt deep adaptation techniques for the Faster R-CNN framework. The techniques that we adapt are the recent techniques based on Gradient Reversal and Maximum Mean Discrepancy (MMD) reduction based techniques. Among them we show that the MK-MMD based method when used appropriately provides the best results. We analyze our model with standard real world settings by using Pascal VOC as source and MS-COCO as target and show a gain of 2.5 mAP at IoU of 0.5 over a source only trained model. We show that this improvement is statistically significant.

## 1 Introduction

Deep networks perform impressively for computer vision problems, but suffer from dataset bias. Finetuning the networks can reduce the bias but require large amount of labelled target data. Therefore unsupervised domain adaptation techniques are well suited for this task.

Unsupervised domain adaptation in visual domain has been studied mostly in light of object recognition but very less work has been done in terms of object detection. We give a deep adaptation pipeline for object detection in this paper. To fix the dataset bias we look into methods that help deep networks regularize. We formulate our training objective to provide us with domain invariance, i.e., the features are more generic in nature. In this paper we apply methods by [1, 2] to Faster R-CNN[3] in order to regularize the training.

## 2 Related Work

Most of the work on unsupervised domain adaptation can be categorized into shallow and deep domain adaptation methods. A recent report by Gabriela [4] surveys extensively in methods of domain adaptation in visual domain.

Shallow methods rely on pre-extracted features and try to align or re-weight these features for adaptation. It has been shown that without adaptation Deep Convolutional Activation Features (DeCAF) [5] generalize very well and beat these methods based on shallow features by a large margin. The shallow methods have been attempted to be applied on these DeCAF features but the gain

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

achieved is far lower than on shallow features as seen in [5]. So focus has been shifted to more promising deep domain adaptation methods.

Our work is based on [6] which is uses Maximum Mean Discrepancy(MMD). They use the Krizhevsky architecture as base CNN model and find out which layers contribute to maximum discrepancy in the source and target datasets, which is the final fully connected layer (fc7). Then they place a discrepancy loss (MMD loss) with a notion of regularizing the weights such that outliers of source distribution have minimum effect on the learned representation in turn reducing the domain shift. [7] use multi-kernel Maximum Mean Discrepancy (MK-MMD). They use a sum of MK-MMD defined at several layers to minimize the domain shift in the learned representation.

In contrast to above method, [1] augments a domain classifier to the network, which predicts image label as source or target. The domain loss is maximized to achieve feature invariance.[8] also use a adversarial training where they use inverted labels to train the discriminator, as it provides stronger gradients. The major contrast to other methods is [8] which considers separate embeddings for source and target space.

Clearly we see a focus of domain adaptation methods to be applied on object recognition problem. There has been significantly low amount of work in adapting the detectors which are much more useful in real world scenario. We do see [9] , who apply the Subspace alignment on RCNNs, but they do not leverage the full power of deep Convolutional Neural Networks (CNNs).

## 3 Adaptation Method

We try to define briefly the methods used by us to augment the CNNs for adaptation purposes.

**Maximum Mean Discrepancy :** Let $x_s \in X_s$ be source data points defined by distribution $p$ and $x_t \in X_t$ be target data points defined by distribution $q$. MMD metric and its empirical estimate can be defined as

$$MMD[\mathcal{F}, X_s, X_t] = \sup_{f \in \mathcal{F}} \left( \frac{1}{|X_s|} \sum_{x_s \in X_s} f_p(x_s) - \frac{1}{|X_t|} \sum_{x_t \in X_t} f_q(x_t) \right) \qquad (1)$$

If $p = q$, i.e., distributions are same then $MMD[\mathcal{F}, X_s, X_t]$ vanishes.

**Gradient Reversal :** The model can be decomposed into 3 components. The first component *feature extractor* ($G_f$) maps the input $x$ into a feature space $f \in R^D$ . This component can be denoted by $G_f(x; \theta_f)$ where $\theta_f$ are component parameters to be learned. The next component *classifier* predicts the label $y$, it is denoted by $G_y(f; \theta_y)$ where $\theta_y$ are component parameters to be learned. The *domain classifier* predicts the domain label $d$ denoted by $G_d(f; \theta_d)$, parameterized by $\theta_d$. The model is shown in Figure 1.

We want to minimize classification loss on source and make the feature domain invariant at the same time. This is done by minimizing loss of label prediction along with maximizing the loss of domain classifier. The objective function

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

is as follows :

$$\begin{aligned} E(\theta_f, \theta_y, \theta_d) = & \sum_{i=1..N, d_i=0} L_y(G_y(G_f(x_i; \theta_f); \theta_y), y_i) - \\ & \lambda \sum_{i=1..N} L_d(G_d(G_f(x_i; \theta_f); \theta_d), d_i) \\ = & \sum_{i=1..N, d_i=0} L_y^i(\theta_f, \theta_y) - \lambda \sum_{i=1..N, d_i=0} L_d^i(\theta_f, \theta_d) \end{aligned} \tag{2}$$

Here $L_y(.,.)$ and $L_d(.,.)$ are classification and domain loss respectively. Directly maximizing the $\theta_d$ is not possible using SGD, so a gradient reversal layer is added which acts as identity transform for forward pass but multiplies a negative constant at backward pass.

## 4 Experiments

We use two popular datasets used in object detection for purpose of domain adaptation. We mainly use PASCAL Visual Object Classes (VOC) [10] 2007 dataset as source dataset which has 20 object categories. We use microsoft's Common Objects in Context (COCO) [11] dataset as the target dataset. The COCO dataset has 80 categories, with 20 categories common with VOC. We only use the subset of COCO which has common categories of VOC. We call this dataset as minicoco for further reference.

The datasets are similar, where COCO dataset being the harder and larger of the two datasets. The VOC07 has about 9,963 images in the training set and minicoco has 66,843 images in the training set. The minicoco validation set has 32,467 images.

### 4.1 Adaptation Networks

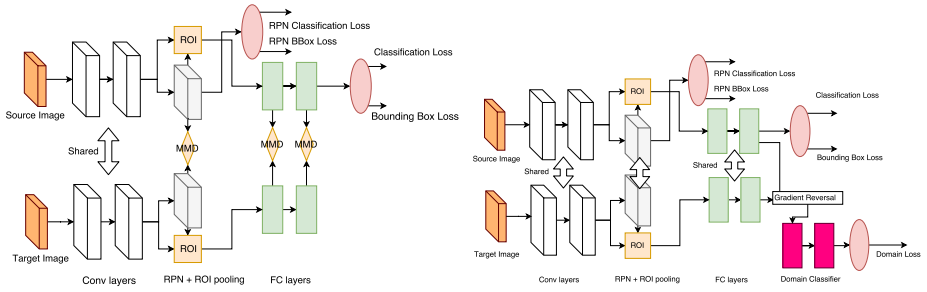


Fig. 1: Left MMD Augmented Faster R-CNN ,Right Gradient Reversal Augmented Faster R-CNN

**CNN architecture :** One of the implementation of *siamese* networks generally work by merging the source and target as a single batch and perform the

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

shared computations, making it computationally efficient. Unlike other object recognition networks, Faster R-CNNs have a batch size of 1 and are unconstrained on image size, which allows scale invariance in Faster R-CNNs. As with domain adaptation the image size of source and target may likely differ. To preserve the property of Faster R-CNNs we go for an architectural choice of two parallel networks with shared weights.

**Gradient Reversal :** We augment the above generic *siamese* CNN architecture with a domain classifier.The domain classifier is a three layered fully connected network with last layer predicting the output domain. Input to the domain classifier is the fixed size feature representations from *fc7* layer of each proposal. For domain loss we use a standard *sigmoid cross-entropy* loss. Like [1] we also suppress the initial noisy predictions of domain classifier using a adaptation term $\lambda_p$ using following formula :

$$\lambda_p = \frac{1}{1 + exp(-\gamma * p)} - 1 \tag{3}$$

Where $\gamma$ is set to 10 for all experiments an $p$ is progress term. The loss weight is set to 0.1 for all experiments. We also observe that the domain loss after a point overweighs the classification objective. To find an optimal point we use a small validation set of the target dataset.

**Maximum Mean Discrepancy :** To generic siamese architecture we add the *mmd_loss_layer* at the fully connected layers. We initialize the network with weights of source only trained model. The MMD layer works in $O((m+n)^2)$ time but also takes $O((m+n)^2)$ space for parallel computation of the kernel function. This limits the number of proposals to be used to train the fast-rcnn part of the network. We go for 300 proposals for each source and target image, which fit the memory. We also look for the choice of *mmd_loss_layer* placement. If applied to the *RPN output layer*, which acts as a feature layer for RPN, we get the best results, In contrast to [6, 7] which state best improvement at *fc6* and *fc7*. For the multi-kernel version all experiments use a combination of five Gaussian kernels. The loss weight is set to 0.1.

## 5 Results

Contrary to Pascal VOC challenge where IoU overlap of .5 is fixed, we use metric specified in MSCOCO challenge where AP is measured at 10 IoU thresholds of [.05:.95]. Averaging over IoUs rewards detectors with better localization. We can see that MMD when done at *rpn features* gives the best results.

**Data Augmentation Strategy**: We follow the data augmentation suggested in [12]. We show the affect of data augmentation strategy on source only trained models as well as other domain adaptation methods. The data augmentation provides a level of color invariance , which can be very useful in providing domain invariant features. Table 1 we can see a average boost of 0.9 mAP at IoU of 0.5 and a boost of 0.5 at COCO metric of IoU at [.5:.95].

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

| **Methods** | **Without Data Augmentation** | | **With Data Augmentation** | |
|---|---|---|---|---|
| | **IoU:.5** | **IoU:[.5:.95]** | **IoU:.5** | **IoU:[.5:.95]** |
| Source Only | 34.95 | 16.10 | 35.37 | 16.44 |
| GRL | 35.38 | 16.52 | 36.30 | 17.10 |
| $MMD_{fc6}$ | 35.51 | 17.01 | 36.36 | 17.45 |
| $MMD_{fc7}$ | 35.44 | 16.99 | 36.17 | 17.36 |
| $MKMMD_{fc6}$ | 35.28 | 16.92 | 36.20 | 17.38 |
| $MKMMD_{fc7}$ | 35.24 | 16.74 | 36.08 | 17.15 |
| $MMD_{RPN}$ | 36.46 | 17.15 | 37.35 | 17.55 |
| Target Only | 46.01 | 22.40 | 46.91 | 22.90 |

Table 1: Summary of mAP of all methods

**Impact on Region Proposal Network** : To test the impact on RPNs we use recall to IoU curve as suggested by [13]. In Figure 2, we show the results of using 2000 region proposals per RPN. We can see that at lower IoU the recall of adaptation methods are higher which decreases at higher IoU.

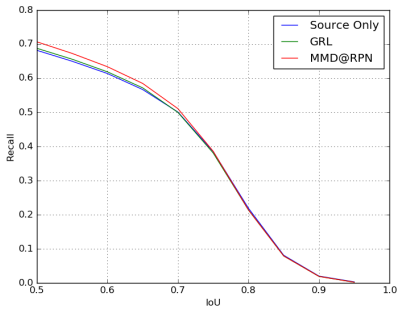


Fig. 2: Recall to IoU on COCO validation set

**Statistical Significance of Results** : We want to establish the statistical significance of our pipeline for domain adaptation on Faster R-CNNs. We determine that whether any domain adaptation done is significant enough over source only trained models. Our analysis is based on the methods proposed in by [14] , specifically using the Friedman test with Nemenyi *post hoc* analysis. It is shown in Figure 3.

## References

[1] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[2] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *International Conference in Computer Vision (ICCV)*, 2015.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

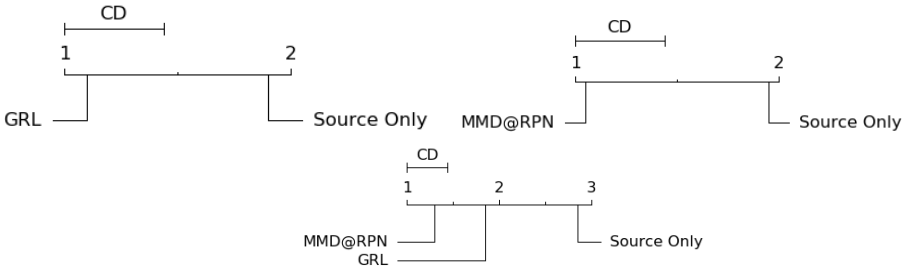


Fig. 3: Analysis of statistically significant difference in the domain adaption methods and Source only trained method , with a significance level of 0.05. The mean rank is plotted on x-axis. The CD calculated as 0.43 and we can see all the methods are way outside the CD , so are statistically significant over source only trained model. We can see MMD is statistically significantly over GRL

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[4] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *CoRR*, abs/1702.05374, 2017.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *CoRR*, abs/1310.1531, 2013.

[6] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[7] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 97–105. JMLR.org, 2015.

[8] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Anant Raj, Vinay P. Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 166.1–166.11. BMVA Press, September 2015.

[10] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[11] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[12] Andrew G. Howard. Some improvements on deep convolutional neural network based image classification. *CoRR*, abs/1312.5402, 2013.

[13] Jan Hendrik Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *CoRR*, abs/1502.05082, 2015.

[14] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.