

000
001 **Unsupervised Deep Domain Adaptation on**
002 **People Detection**

004 Anonymous ECCV submission
005

006 Paper ID ***
007

010 **Abstract.** This paper addresses the problem of unsupervised domain
011 adaptation on the task of people detection in crowded scenes. That is,
012 given a well-trained deep detection model on source domain, we aim to
013 adapt it into the target domain for which no annotations are needed.
014 Firstly, we utilize iterative algorithm to auto-annotate samples on target
015 domain. While auto-annotated samples include noise, the rate of false
016 positive tends to explored. Thus, we mixed into training examples from
017 source domain to suppress the data noise. Furthermore, we transform
018 the inner product layers of our network into two separate layers and
019 proposed a weights regularizer to avoid overfitting. Compared to generic
020 model without any fine tuning on samples on target domain, the pro-
021 posed method increased recall by $xx\%$ with precision drop from $xx\%$ to
022 $xx\%$ on the target scenes. Also, we perform our algorithm on standard
023 domain adaptation benchmarks Office Dataset with both supervised and
024 unsupervised settings and our results are state of art.

024 **Keywords:** Unsupervised Domain Adaptation, Deep Neural Network,
025 People Detection

027 **1 Introduction**

030 Deep neural network has outstanding result on computer vision tasks, howev-
031 er, it require large labelled dataset. In famous challenges like PASCAL VOC
032 and MS COCO, over millions of images with labels are needed for training. In
033 surveillance applications like people detection, the annotation process is even
034 human resource consuming. Till today, there are millions of cameras deployed
035 for surveillance. However, surveillance situations varies in background, lights,
036 viewpoints and so on in real world, which make the labelling work for all surveil-
037 lance applications ever more unpractice. In the task of people detection, tens of
038 thousands of annotated images are needed to train a deep neural network.

039 When labelled data are few and even lack of in target domain, domain adap-
040 tation help to reduce the amount of labelled data needed when given abun-
041 dant labelled data in source domain. Most traditional works [1–5] try to learn
042 a shared representation between source and target domain, or project features
043 into a common subspace. Recently, works are also [6–8] proposed to learn scene-
044 specific detector by deep architectures. In our task of people detection, we try

045 to shift deep detection network well-trained on the source scene to target scenes
 046 on which no annotation are needed.

047 In this paper, we proposed a new approach of unsupervised deep domain
 048 adaptation on people detection. Using generic model as initialization, we firstly
 049 use iterative algorithm to auto-annotate samples on target domain as fake ground
 050 truth. During every iteration, target model are updated and utilized to auto-
 051 annotate samples for next iteration. However, the data noise in auto-annotated
 052 dataset, like lack of true negative instances and false positive instances, will
 053 leads to exploration of false positive rate, as well as impeding further increase of
 054 recall. In order to eliminate the effect of data noise, two methods are proposed to
 055 regularize the training of the deep network. On the one hand, we mixed into auto-
 056 annotated dataset with annotated samples from source domain to compensate
 057 the lack of true negative instances. On the other hand, we separate the operation
 058 of last inner product layer of our network into two sub-operations – element-wise
 059 multiply and sum operations, resulting new element-wise multiply layer and sum
 060 layer. Thus, a weights regularizer on element-wise layer can be added into the
 061 deep network as unsupervised loss to avoid exploration of false positive rate and
 062 boost recall.

063 Also, we further evaluate our method on standard domain adaptation bench-
 064 mark Office Dataset. The results of our adaptation approach outperform pre-
 065 viously published works on both supervised and unsupervised settings, which
 066 demonstrate the feasibility of our adaptation approach on both detection and
 067 classification tasks.

068 The contributions of our work are three folds.

- 069 – We provided a feasible scheme to shift deep detection network well-trained on
 070 the source scene to target scenes on which no annotated data are needed. This
 071 makes easier the widely deploy of deep neural network on various surveillance
 072 applications.
- 073 – As most algorithms focus on the last feature vector (also last inner product
 074 layer), we have one step further and transform the last inner product layer
 075 into element-wise multiply layer and sum layer. A weight regularizer can,
 076 thus, be added on element-wise layer to have better effect on suppressing
 077 exploration of false positive rate and increasing recall.
- 078 – Experiments on standard domain adaptation benchmarks also demonstrate
 079 the effectiveness of our approach on other deep domain adaptation tasks.

080 The remainder of this paper is organized as follows. Section 2 reviews related
 081 works. Section 3 presents the details of our approach. Experimental results are
 082 shown in Section 4. Section 5 concludes the paper.

084 2 Relate Work

087 In many detection works, generic model trained by large amount of samples on
 088 source domain are directly utilized to detect on target domain. They assume
 089 that samples on target domain are subsets of source domain. However, when the

090 distribution of data on target and source domain varies largely, the performance
 091 will drop significantly. Domain adaptation aims to reduce the amount of data
 092 needed for target domain.

093 Many domain adaptation works tried to learn a common representation s-
 094 pace shared between source and target domain. Saenko et al. [1, 2] proposed a
 095 both linear-transform-based technique and kernel-transform-based technique to
 096 minimize domain changes. Gopalan et al. [3] projected features into Grassmann
 097 manifold instead of operating on features of raw data. Alternatively, Mesnil et al.
 098 [9] used transfer learning to obtain good representations. However, these meth-
 099 ods are limited because scene-specific features are not learned to boost accuracy.
 100 The regularizer of our method are inspired these works.

101 Another group of works [4, 5, 10] on domain adaptation is to make the distri-
 102 bution of source and target domain more similar. Among these works, Maximum
 103 Mean Discrepancy (MMD) is used to as a metric to reselect samples from source
 104 domain in order to have similar distribution as target samples. In [11], MMD is
 105 incorporated as regularization to reduce distribution mismatch.

106 There are also works on deep adaptation to construct scene-specific detector.
 107 Wang et al.[6] explored context cues to compute confidence, and [7] learns dis-
 108 tributions of target samples and proposed a cluster layer for scene-specific visual
 109 patterns. These works reweighted auto-annotated samples for their final object
 110 function and additional context cues are needed for reliable performance. Alter-
 111 nately, Hattori el al. [8] learned scene-specific detector by generating a spatially-
 112 varying pedestrian appearance model. And Pishchulin et al. [12] used 3D shape
 113 models to generate training data. However, Synthesis for domain adaptation are
 114 also costly.

115 3 Our Approach

116 In this section, we introduce our unsupervised deep domain adaptation approach
 117 on the task of people detection. We denote the training images of source do-
 118 main as $\mathbf{X}^S = \{x_i^S\}_{i=1}^{N_i^S}$ and that of target domain as $\mathbf{X}^T = \{x_j^T\}_{j=1}^{N_j^T}$. For
 119 each image in source domain, we have corresponding annotations denoted as
 120 $\mathbf{B}_i^S = \{b_{i,k}^S\}_{k=1}^{N_i^S}$ with $b_{i,k}^S = (x, y, w, h) \in R^4$, however, the annotations for im-
 121 ages in target domain are auto-labelled which we denote as $\tilde{\mathbf{B}}_j^T = \{\tilde{b}_{j,k}^T\}_{k=1}^{N_j^T}$
 122 with $\tilde{b}_{j,k}^T = (\tilde{x}, \tilde{y}, \tilde{w}, \tilde{h}) \in R^4$. The annotations for target domain changes for ev-
 123 ery iteration during the adaptation process. Human annotated images on target
 124 domain are used only for evaluation.

125 The adaptation architecture of our approach consists of two streams – source
 126 stream M^S and target stream M^T , as shown in Fig x(!). Source stream takes
 127 samples from source domain as input, and target stream operates on samples
 128 from target domain. These two streams can utilize any end to end deep detection
 129 network as their model. Here we use the below mentioned network in Sec 3.1 in
 130 our experiment. In initialization stage, we firstly use abundant annotated sam-
 131 ples from source domain to train the model of source stream under a supervised

loss function to regress bounding box. After its convergence, the weights of the model of source stream are used to initialize target stream. In adaptation stage, iteration algorithm is used as training method. Target model in target stream is trained and upgraded in this process while source stream stays static.

Both supervised loss and unsupervised regularizer are designed as loss function to train the target stream for learning scene-specific detector as well as avoid overfitting. For supervised loss, the auto-annotated data can be used as training labels. As the auto-annotated samples contains data error, an unsupervised loss are required to regularize the network. We take the source model in source stream as a reference for feature vector distribution. We defined the combination of supervised loss and unsupervised loss as our loss function for adaptation:

$$L(\theta^T | \mathbf{X}^S, \mathbf{B}^S, \mathbf{X}^T, \tilde{\mathbf{B}}^T, \theta^S) = L_S + \alpha * L_U \quad (1)$$

$$L_S = \sum_{j=1}^{N^T} \sum_{k=1}^{N_j^T} (r(\theta^T | x_j^T, \tilde{b}_{j,k}^T) + c(\theta^T | x_j^T, \tilde{b}_{j,k}^T)) \quad (2)$$

$$L_U = L_{MMD}(\theta^T | \mathbf{X}^S, \mathbf{X}^T, \theta^S) \quad (3)$$

where L_S is supervised loss to learn the scene-specific detector and L_U is the unsupervised regularizer part. $r(\cdot)$ is a regression loss for bounding box location, like norm-1 loss, and $c(\cdot)$ is a classification loss for bounding box confidence, such as cross-entropy loss. And $L_{MMD}(\cdot)$ is the MMD-based loss for unsupervised regularization. Coefficient α balance the effect of supervised and unsupervised loss. We set $\alpha = 10$ in our experiments.

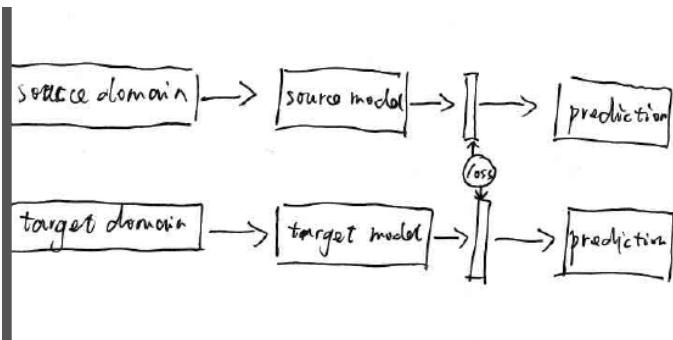
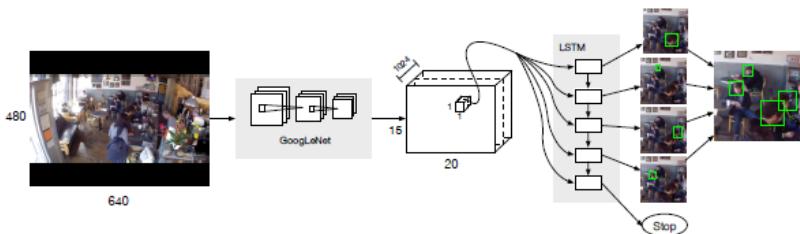


Fig. 1. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in italics, in parentheses, as shown in this sample caption. The last sentence of a figure caption should generally end without a full stop

180 3.1 Detection Network

181 The generic model ¹ used in our adaptation architecture for source and tar-
 182 get stream is an end to end detection network without any precomputed region
 183 proposals needed. It consists of a GoogLeNet [13] for feature extraction and a
 184 RNN-based decoder for output of bounding box and corresponding confidence,
 185 as shown in Fig x(!). Firstly, the GoogLeNet model encode the image into a
 186 feature map (15x20x1024) of which each 1024 dimension vector are data repre-
 187 sentation of its receptive field corresponding to a subregion of the image. Then
 188 the RNN-based layers with batch size 300 will decode the data representation
 189 and sequentially predict 5 possible bounding boxes by the order of its corre-
 190 sponding confidence. Finally, all outputs are summarized to give final detection
 191 results. When trained with abundant samples from source domain, the obtained
 192 model has a high precision on target domain, however, its recall is low. Also, dif-
 193 ferent from other detection network [14, 15], which need precomputed proposals
 194 for classification and fine regression, this generic model directly predict bound-
 195 ing boxes with high confidence. Thus, negative instances may contain people and
 196 non-people predictions, and cannot be employed in adaptation training.



201
 202 **Fig. 2.** One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)
 203 lead to the same summed estimate at x_s . This shows a figure consisting of different
 204 types of lines. Elements of the figure described in the caption should be set in italics,
 205 in parentheses, as shown in this sample caption. The last sentence of a figure caption
 206 should generally end without a full stop

219 3.2 Iterative Algorithm

220 In this section we introduce the iteration algorithm as training method at adap-
 221 tation stage. The auto-annotating tool takes images on target domain with high
 222 confidence as training data for next iteration. To generate the auto-annotated

223 ¹ Proposed by Russel et al.

225 data for the first iteration, we utilize the generic model well-trained on source
 226 domain, which results in training set on the target domain. As mentioned in
 227 Sec 3.1, the training set on target domain auto-annotated by the generic model
 228 in our experiment have low recall and high precision. At subsequent iterations,
 229 training set on target domain are auto-annotated by upgraded model resulting
 230 from training of last iteration. Among every iteration, these auto-annotated data
 231 are used as training data to upgrade the deep network.

232 As the training set on target domain auto-annotated for the training of first
 233 iteration have low recall rate and people and non-people regions mixed in nega-
 234 tive instances, we ignore back propagation of bounding boxes with low confidence
 235 during training. That is, we encourage the network to have more confidence on
 236 positive instances and stay conservative toward negative instances. This policy
 237 will no doubt resulting predictions of many non-people instance and impeding
 238 the further training when many non-people instances are regarded as people in-
 239 stances by the target model. To compensate for lack of true negative instances,
 240 we add annotated samples from source domain into the training data, which
 241 are human annotated and can thus provide true negative samples. In our ex-
 242 periment, when training on additional samples from source domain, we only do
 243 back propagation of bounding boxes with low confidence. At the same time,
 244 the unsupervised loss will also regularize the network. The complete adaptation
 245 process is illustrated in Algorithm 1. After a predetermined iteration limit N^I
 246 is reached, we obtain our final detection model on the target domain.

Algorithm 1 Deep domain adaptation algorithm (to be completed)

```

250 1: procedure DEEP DOMAIN ADAPTATION
251 2: Train source model on source stream with abundant annotated data
252 3: Use well-trained source model on source stream to initialize model on target stream
253   as  $M_0$ 
254 4:   for i = 0: $N^I$  do
255 5:      $M_i$  generate "fake ground truth"  $G_i$  of target domain
256 6:     balbal
257 7:     balbabla
258 8:      $G_i = G_i +$  random samples from source domain
259 9:     Take  $G_i$  as training data to upgrade  $M_i$  into  $M_{i+1}$ 
260 10:   end for
261 11:  $M_{N^I}$ : final model.
262 12: end procedure
  
```

265 3.3 Unsupervised weights regularizer on Element-wise Multiply 266 Layer

267 **Element-wise Multiply Layer** In deep neural network, the last feature vector
 268 layer are taken as an important data representation of input images. However,

in this paper, we take one step further to focus on the last full connected layer which serves as an decoder to decode rich information contained in the last feature vector into final outputs. As source model are trained with abundant labelled data on source domain, the parameters of the last full connected layer are also well converged. We assume that a regularizer on the last full connected layer will achieve better results compared with the last feature vector layer. Firstly, denote the last feature vector, parameters of the last full connected layer and final outputs as $\mathbf{F}_{(N^B \times N^D)}$, $\mathbf{C}_{(N^D \times N^O)}$ and $\mathbf{P}_{(N^B \times N^O)}$, respectively. The operation of full connected layer can be thus formulated as matrix multiply:

$$\mathbf{P} = \mathbf{F} * \mathbf{C} \quad (4)$$

$$P_{b,o} = \sum_d F_{b,d} * K_{d,o} \quad (5)$$

Inspired by this form, we separate the above formula into two sub-operations – element-wise multiply and sum, which can be formulated as:

$$M_{b,o,d} = F_{b,d} * C_{d,o} \quad (6)$$

$$P_{b,o} = \sum_d M_{b,o,d} \quad (7)$$

where $\mathbf{M}_{(N^B \times N^O \times N^D)} = [\mathbf{m}_{b,o}]$ is the parameter tensor of element-wise multiply operations. $\mathbf{m}_{b,o}$ is a vector with N^D dimensions, which will be the basis of unsupervised regularizer. Finally, we can equivalent-transform the last full connected layer between the last feature vector layer and final outputs layer into element-wise multiply layer and sum layer. The transformed element-wise layer is thus the last layer with parameters before output layers. Fig x (!) illustrates the transform.

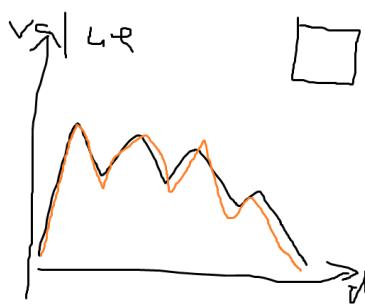


Fig. 3. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s .

Unsupervised weights regularizer on Element-wise Multiply Layer In works [decaf][1], the last feature vector layer are regarded as the final representation of images. (how to introduce beyond sharing weights) In domain adaptation tasks, when generic deep model are trained with abundant data from source domain, the last element-wise layer mentioned in Sec 3.3 also includes rich information leading to the final output. For the nodes in the output layer, inputs from element-wise layers that will contribute to its value are not randomly decided. In this paper, we assume that the distribution of $\mathbf{m}_{b,o}$ of the last element-wise layer on both source and target domain should be similar. In the last element-wise multiply layer, every dimension of $\mathbf{m}_{b,o}$ may contribute to the final output. However, as the old model on source domain are trained with abundant images, the distribution of $\mathbf{m}_{b,o}^T$ should be consistent compared with $\mathbf{m}_{b,o}^S$ of source domain when adapting deep network on target domain. We utilize MMD (maximum mean discrepancy) to encode the similarity of element-wise multiply layer of source domain and target domain:

$$L_{MMD}(\theta^T | \mathbf{X}^S, \mathbf{X}^T, \theta^S) = \frac{1}{N^O} \sum_{o=1}^{N^O} \left\| \frac{1}{N^B} \sum_{b=1}^{N^B} \mathbf{m}_{b,o}^T - \frac{1}{N^B} \sum_{b=1}^{N^B} \mathbf{m}_{b,o}^S \right\|^2 \quad (8)$$

which can also interpreted as the Euclidean distance between the center of $\mathbf{m}_{b,o}^T$ and $\mathbf{m}_{b,o}^S$ across all output nodes. It's unpractical to get the distribution of the whole training set, while too few images cannot obtain a stable center for regularization. In our experiments, the $L_{MMD}(\cdot)$ loss is calculated for every batch. An example comparison of centers of $\mathbf{m}_{b_i,o}^S$ and $\mathbf{m}_{b_j,o}^S$ are shown in Fig x(1).

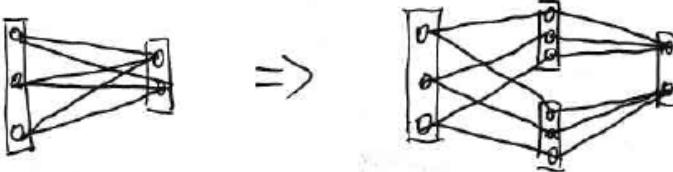
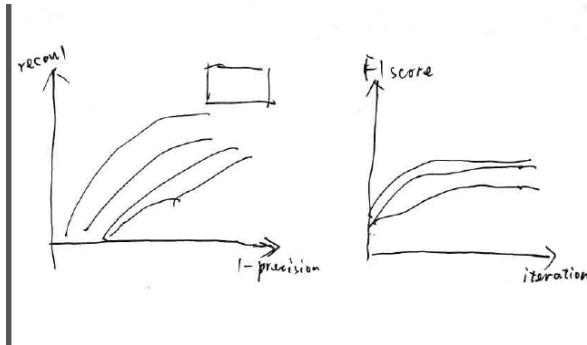


Fig. 4. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s .

360 4 Experiment Results

361
 362 In this section, we introduce experiment results on both surveillance applications
 363 and standard domain adaptation dataset. Our motivation for unsupervised do-
 364 main adaptation method is for easier deployment of We firstly evaluate our
 365 approach on video surveillance. Then we employ our approach to standard do-
 366 main adaptation benchmarks on both supervised and unsupervised settings to
 367 demonstrate the effectiveness of our method.



381 **Fig. 5.** One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)
 382 lead to the same summed estimate at x_s . This shows a figure consisting of different
 383 types of lines. Elements of the figure described in the caption should be set in italics,
 384 in parentheses, as shown in this sample caption. The last sentence of a figure caption
 385 should generally end without a full stop

388 4.1 Domain Adaptation on Crowd Dataset

390 **Dataset and evaluation metrics** To show the effectiveness of our domain
 391 adaptation approach on people detection, we collected a dataset consisting of
 392 3 target scenes for target domain. These three scenes contain 1308, 1213 and
 393 331 un-annotated images with 0000, 0000, 0000 people instances respectively.
 394 For each scene, 100 images are annotated for evaluation. Instead of labelling the
 395 whole body of a person, we labels the head of a person as bounding box during
 396 training. The motivation for labelling only people heads comes from detection of
 397 indoor people or in crowded scenes, where the body of a person may be invisible.
 398 The dataset for source domain are Brainwash Dataset released on [http://d2.mpi-](http://d2.mpi-inf.mpg.de/datasets)
 399 [inf.mpg.de/datasets](http://d2.mpi-inf.mpg.de/datasets). Brainwash Dataset consists of over 11917 images from 3
 400 crowded scenes. Examples of images from source and target domain are shown
 401 in Fig x(!).

402 Our evaluation metrics for detection uses the protocol defined in PASCAL
 403 VOC [16]. To judge a predicted bounding box whether correctly matches a
 404 ground truth bounding box, their intersection over their union must exceed 50%.

405 And Multiple detections of the same ground truth bounding box are regarded as
 406 one correct prediction. We plot the precision-recall curve in Fig x(!). Also, the
 407 F1 score $F1 = 2 * precision * recall / (precision + recall)$ during the adaptation
 408 process are also shown in Fig x(!).

410 **Experimental settings** We use deep learning framework Caffe [17] as the
 411 adaptation architecture of our approach. During the adaptation, we set learning
 412 rate as 0.01 and momentum as 0.5. At initialization stage, GoogLeNet weights are
 413 firstly used to initialize source model of source stream, while parameters in RNN
 414 layers are randomly initialized from a uniform distribution. For each iteration,
 415 100 auto-annotated images from target domain and 1000 annotated images from
 416 source domain are alternatively used for training. The outputs of our detection
 417 network contains bounding box locations and corresponding confidence, thus
 418 there are two full connected layers between the last feature vector layer and
 419 the final outputs. Experiments of unsupervised regularizer on element-wise layer
 420 for bounding box regression have no additional improvement on the performance
 421 when regularizer on element-wise layer for box confidence classification are added
 422 already. Our approach on domain adaptation are executed separately in the 3
 423 target scenes.

424

425 **Comparison with baseline methods** To demonstrate the effectiveness of our
 426 approach, 4 methods are compared with method 4 as our final approach:

- 427 1. Only auto-labeled samples on target domain are used for training, and with-
 428 out any unsupervised regularizer.
2. Only auto-labeled samples on target domain are used for training, with M-
 429 MD regularizer on last element-wise multiply layer as unsupervised weights
 430 regularizer.
3. Both auto-labeled images from target domain and labeled images from source
 431 domain are alternately sampled for training, with MMD regularizer on last
 432 feature vector as unsupervised weights regularizer.
4. Both auto-labeled images from target domain and labeled images from source
 433 domain are alternately sampled for training, with MMD regularizer on last
 434 element-wise multiply layer as unsupervised weights regularizer.

435 Fig x(!) plots the precision-recall curve of the above comparison methods in
 436 target scene 1). Also, the F1 score changes of every iteration during adaptation
 437 process are also depicted in Fig x(!). Table x(!) gives concrete precision and recall
 438 value of the 4 comparison methods on three target scenes when the F1 scores
 439 are at their highest. Examples of adaptation results are shown in Fig x(!).

440

441 **Performance evaluation** From the Table x(!) and Fig x(1), we have the fol-
 442 lowing observations:

- 443 – The recall values of method 1,2,3,4, which all utilized iteration algorithm
 444 to upgrade the target model, are larger than that of method 0, which are

	Scene 1			Scene 2			Scene 3		
	1-Pr	Re	F1	1-Pr	Re	F1	1-Pr	Re	F1
method 0	0.101	0.187	0.309	0.059	0.599	0.732	0.190	0.476	0.599
method 1	0.245	0.408	0.530	0.632	0.905	0.524	0.176	0.778	0.800
method 2	0.284	0.476	0.572	0.012	0.837	0.906	0.078	0.653	0.764
method 3	0.109	0.496	0.637	0.002	0.721	0.838	0.044	0.611	0.746
method 4	0.140	0.530	0.656	0.006	0.811	0.893	0.097	0.778	0.836

source model trained on source domain. This implies the effectiveness of our iteration algorithm in auto-annotation and iterative training.

- Compared with method 1, method 2 has higher F1 score. Their difference on whether a MMD regularizer are added into loss function demonstrates that our unsupervised regularizer can suppress data error and thus boost the final recall.
- Method 4 has both higher precision and higher recall than method 2, which demonstrates the effectiveness of additional samples from source domain during adaptation process.
- Compared with method 3, the recall of method 4 are further boosted. This results from the transformed element-wise layer which provided better regularization effect on the target model.

4.2 Domain Adaptation on Standard Classification Benchmark

Office dataset The Office dataset [1] comprises 31 categories of objects from 3 domains (Amazon, DSLR, Webcam). Example images are depicted in Fig. x!). As Amazon domain contains 2817 labelled images, which is the largest, we take it as source domain and Webcam domain as target domain. We follow the standard protocol for both supervised and unsupervised settings. Specifically, for supervised domain adaptation, we use 20 randomly sampled images with labels for each category as training data for Amazon domain. When evaluate on unsupervised domain adaptation, 3 labelled images from target domain are additional selected for each class. For both settings, the rest of images on target domain are used for evaluation.

Experimental settings and network design On supervised setting, we reused the architecture in people detection. We utilize AlexNet [18] as the generic model of both streams. Firstly, we train the source model on source stream with provided training data from Amazon domain. Then iteration algorithm mentioned in Sec 3.2 are utilized for adaptation. The auto-labelling tool takes images on target domain with confidence exceeding 0.9 as training data for next iteration. The difference is besides auto-labelled images on target domain, 3 human labelled images for each class are also included as training data for target model. For each iteration, 100 images are randomly sampled from training data. The unsupervised MMD regularizer added on the element-wise layer transformed



Fig. 6. Some examples from three domains in the Office dataset.

Fig. 6. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)

from the last full connect layer of target model set the coefficient value α as 10 on Eq 3, the same as that in people detection task.

We use the same experimental setting for our unsupervised adaptation, except that at adaptation stage, no human labelled images can be added into the training set.

Performance evaluation In Table x(!), we compare our approach with other six recently published works in both supervised and unsupervised settings. The outstanding performance on both settings confirms the effectiveness of our iteration algorithm and MMD regularizer on the element-wise layer transformed from the last full connect layer.

	$A \rightarrow W$	
	Supervised	Unsupervised
GFK(PLS,PCA)[19]	46.4	15.0
SA [20]	45.0	15.3
DA-NBNN [21]	52.8	23.3
DLID [22]	51.9	26.1
DeCAF ₆ S [23]	80.7	52.2
DaNN [11]	53.6	35.0
Ours	84.3	66.3

5 Conclusions

The paper ends with a conclusion.

References

1. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Computer Vision–ECCV 2010. Springer (2010) 213–226



Fig. 7. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)

- 585 2. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain
586 adaptation using asymmetric kernel transforms. In: Computer Vision and Pattern
587 Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1785–1792
- 588 3. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An
589 unsupervised approach. In: Computer Vision (ICCV), 2011 IEEE International
590 Conference on, IEEE (2011) 999–1006
- 591 4. Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J.: Correcting
592 sample selection bias by unlabeled data. In: Advances in neural information
593 processing systems. (2006) 601–608
- 594 5. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.:
595 Covariate shift by kernel mean matching. Dataset shift in machine learning **3**(4)
596 (2009) 5
- 597 6. Wang, X., Wang, M., Li, W.: Scene-specific pedestrian detection for static video
598 surveillance. Pattern Analysis and Machine Intelligence, IEEE Transactions on
599 **36**(2) (2014) 361–374
- 600 7. Zeng, X., Ouyang, W., Wang, M., Wang, X.: Deep learning of scene-specific class-
601 ifier for pedestrian detection. In: Computer Vision–ECCV 2014. Springer (2014)
602 472–487
- 603 8. Hattori, H., Naresh Boddeti, V., Kitani, K.M., Kanade, T.: Learning scene-specific
604 pedestrian detectors without real data. In: Proceedings of the IEEE Conference
605 on Computer Vision and Pattern Recognition. (2015) 3819–3827
- 606 9. Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I.J., Lavoie,
607 E., Muller, X., Desjardins, G., Warde-Farley, D., et al.: Unsupervised and transfer
608 learning challenge: a deep learning approach. ICML Unsupervised and Transfer
609 Learning **27** (2012) 97–110
- 610 10. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discrimin-
611 atively learning domain-invariant features for unsupervised domain adaptation.
612 In: Proceedings of The 30th International Conference on Machine Learning. (2013)
613 222–230
- 614 11. Ghifary, M., Kleijn, W.B., Zhang, M.: Domain adaptive neural networks for object
615 recognition. In: PRICAI 2014: Trends in Artificial Intelligence. Springer (2014)
616 898–904
- 617 12. Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormählen, T., Schiele, B.:
618 Learning people detection models from few training samples. In: Computer Vision
619 and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1473–
620 1480
- 621 13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D.,
622 Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings
623 of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
- 624 14. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on
625 Computer Vision. (2015) 1440–1448
- 626 15. Vu, T.H., Osokin, A., Laptev, I.: Context-aware cnns for person head detection.
627 In: Proceedings of the IEEE International Conference on Computer Vision. (2015)
628 2893–2901
- 629 16. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisser-
630 man, A.: The pascal visual object classes challenge: A retrospective. International
631 Journal of Computer Vision **111**(1) (2015) 98–136
- 632 17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama,
633 S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding.
634 arXiv preprint arXiv:1408.5093 (2014)

- 630 18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. 631 (2012) 1097–1105 632
- 633 19. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised 634 domain adaptation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2066–2073 635
- 636 20. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual 637 domain adaptation using subspace alignment. In: Proceedings of the IEEE International 638 Conference on Computer Vision. (2013) 2960–2967 639
- 640 21. Tommasi, T., Caputo, B.: Frustratingly easy nbnn domain adaptation. In: Proceedings 641 of the IEEE International Conference on Computer Vision. (2013) 897–904 642
- 643 22. Chopra, S., Balakrishnan, S., Gopalan, R.: Dlid: Deep learning for domain 644 adaptation by interpolating between domains. In: ICML workshop on challenges in 645 representation learning. Volume 2. (2013) 646
- 646 23. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: 647 Decaf: A deep convolutional activation feature for generic visual recognition. arXiv 648 preprint arXiv:1310.1531 (2013) 649
- 650
- 651
- 652
- 653
- 654
- 655
- 656
- 657
- 658
- 659
- 660
- 661
- 662
- 663
- 664
- 665
- 666
- 667
- 668
- 669
- 670
- 671
- 672
- 673
- 674