

000
001 **Unsupervised Deep Domain Adaptation on**
002 **People Detection**
003

004 Anonymous ECCV submission
005

006 Paper ID ***
007

009
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 deep detection model well-trained on source domain, we adapt
013 it into scene-specific detectors for any target domain on which no anno-
014 tations are available. Firstly, we utilize iterative algorithm to iteratively
015 auto-annotate target samples with high confidence on people instance as
016 training set for scene-specific model on target domain. However, auto-
017 annotated samples not only are lack of negative samples, but contains
018 false positive samples. Therefore, on the one hand, we reuse negative
019 samples from source domain to compensate for imbalance between the
020 amount of positive samples and negative samples. On the other hand,
021 we design an unsupervised regularizer based on deep network to mitigate
022 influence from data error. Besides, we transform the last full connected
023 layer into two sub-layers— element-wise multiply layer and sum layer, on
024 which the unsupervised regularizer can be added on. In experiments on
025 people detection, the proposed method boosts recall by nearly 30% while
026 precision stays almost the same. Furthermore, we perform our method
027 on standard domain adaptation benchmarks on both supervised and un-
028 supervised settings and our results are state of the art.
029

030
031 **Keywords:** Unsupervised Domain Adaptation, Unsupervised Regular-
032 izer, Deep Neural Network, People Detection
033

034 **1 Introduction**
035

036 Deep neural network has shown great power on traditional computer vision tasks,
037 however, the labelled dataset should be large enough to train a deep model. In
038 famous challenges such as PASCAL VOC and MS COCO, millions of labelled
039 images are needed for training. This is also the case in surveillance applications.
040 The annotation process for the task of people detection in crowded scenes is even
041 more resource consuming, cause we need to label concrete locations of people
042 instances. In modern society, there are over millions of cameras deployed for
043 surveillance. However, these surveillance situations vary in lights, background,
044 viewpoints, camera resolutions and so on. Directly utilizing models trained on
old scenes will results in poor performance on the new situations due to data
distribution changes. It is also unpractice to annotate people instances for every
surveillance situation.

When there are few or even none of labelled data in target domain, domain adaptation helps to reduce the amount of labelled data needed. Most traditional works [1–5] either learn a shared representation between source and target domain, or project features into a common subspace. Recently, there are also works [6–8] proposed to learn a scene-specific detector by deep architectures. However, these approaches are heuristic either on constructing feature space or re-weighting samples. Our motivation of developing a domain adaptation architecture is to reduce heuristic methods required during adaptation process.

In this paper, we proposed a new approach of unsupervised deep domain adaptation on people detection. Using source model trained on source domain as initialization, we utilize iterative algorithm to iteratively auto-annotate target examples with high confidence as people instance on target domain for the first iteration. During each iteration, these auto-annotated data are regarded as training set to update target model, which, then, can be taken as the auto-annotation tool to auto-annotate target samples for the next iteration. However, these auto-annotated samples are defective, including lack of negative samples and existence of false positive samples, which will no doubt lead to exploration of predictions on non-people instances. Therefore, on the one hand, to compensate for the quantitative imbalance between positive and negative samples, we randomly sample negative instances from source domain and mix into training set. On the other hand, based on deep network, we design an unsupervised regularizer to mitigate influence from data error and avoid overfitting. To have better regularization effect during adaptation process, we transform the last full connected layer of deep model into two sub-layers, element-wise multiply layer and sum layer. Thus, the unsupervised regularizer can be added on element-wise multiply layer to adjust all weights in the deep network and gain better performance.

Also, we further evaluate our approach on standard domain adaptation benchmark Office Dataset. The results of our adaptation approach outperform previously published works on both supervised and unsupervised scenarios, which also demonstrate the feasibility of our adaptation approach on both detection and classification tasks.

The contributions of our work are three folds.

- We proposed a feasible scheme to learn scene-specific deep detectors for target domains by unsupervised methodology, which can be easily deployed to various surveillance situations without any additional annotations.
- For better performance of unsupervised regularizer, we transform the last full connected layer of deep network into two sub-layers, element-wise layer and sum layer. Thus, all weights contained in the deep network can be adjusted under the unsupervised regularizer. To our knowledge, this is the first attempt to transform full connected layers for the purpose of domain adaptation.
- Experiments on standard domain adaptation benchmarks for classification also demonstrate the applicability of our approach to other deep domain adaptation tasks.

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

2 Relate Work

In many detection works, generic model trained by large amount of samples on source domain are directly utilized to detect on target domain. They assume that samples on target domain are subsets of source domain. However, when the distribution of data on target and source domain varies largely, the performance will drop significantly. Domain adaptation aims to reduce the amount of data needed for target domain.

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

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

There are also works on deep adaptation to construct scene-specific detector. Wang et al.[6] explored context cues to compute confidence, and [7] learns distributions of target samples and proposed a cluster layer for scene-specific visual patterns. These works re-weighted auto-annotated samples for their final object function and additional context cues are needed for reliable performance. However, heuristic methods are required to re-weight samples. Alternately, Hattori el al. [8] learned scene-specific detector by generating a spatially-varying pedestrian appearance model. And Pishchulin et al. [12] used 3D shape models to generate training data. However, Synthesis for domain adaptation are also costly. Our approach minimize heuristic algorithms needed during adaptation process.

3 Our Approach

In this section, we introduce our unsupervised domain adaptation architecture on the task of people detection in crowded scenes. Under the unsupervised setting, we use iterative algorithm to iteratively auto-annotate target samples and update target model. As the auto-annotated samples contain data error, the performance decline caused by false positive samples will exceed the performance boost resulting from true positive samples if the target model is trained only

with detection loss function. Therefore, an unsupervised regularizer is required to mitigate the influence from data error on target model. Based on the assumption that source domain and target domain should share the same feature space after feature extraction layers, we encode an unsupervised regularizer to make a constraint that the distribution of data representation on the element-wise multiply layer should be similar between source stream and target stream.

The adaptation architecture of our approach consists of two streams – source stream and target stream, as shown in Fig 1. Source stream takes samples from source domain as input, while target stream are trained by auto-annotated positive samples from target domain and negative samples from source domain. These two streams can utilize any deep detection network as their basic model, as well as their detection loss function as supervised loss functions of two streams. In our experiment, we use the detection network mentioned in Section 3.3 as the basic model. For the purpose of unsupervised regularizer, the last full connected layers on both source and target model are transformed into two sub-layers – element-wise multiply layer and sum layer, as mentioned in Section 3.2.

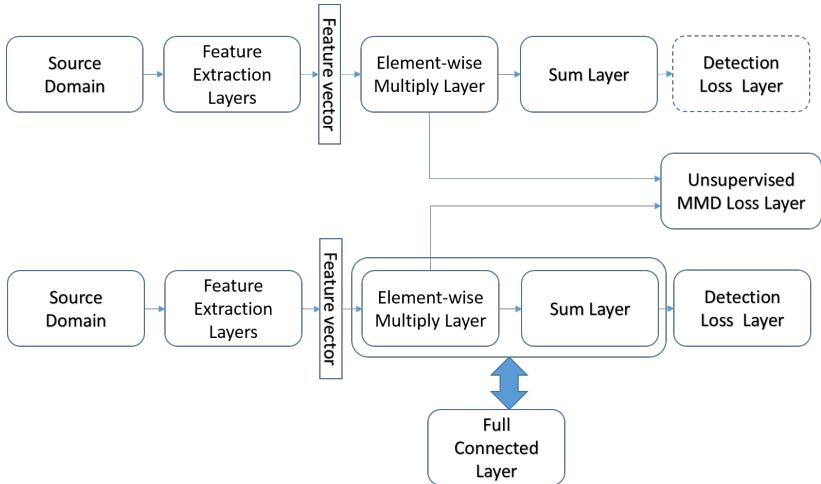


Fig. 1. xxxxxxxxxxxxxxxxxxxxxxxxxx

The loss function of adaptation architecture is composed of a supervised loss and an unsupervised regularizer. We denote training samples from source domain as $\mathbf{X}^S = \{x_i^S\}_{i=1}^{N^S}$. For training samples on source domain, we have corresponding annotations $Y^S = \{y_i^S\}_{i=1}^{N^S}$ with $y_i^S = (b_i^S, c_i^S)$, where $b_i^S = (x, y, w, h) \in R^4$ is the bounding box location and $c_i^S \in \{0, 1\}$ is the label indicating whether x_i^S is a people instance. Also, we can denote the training samples on target domain as $\mathbf{X}^{T,n} = \{x_j^{T,n}\}_{j=1}^{N^{T,n}}$ and corresponding annotations as $Y^{T,n} = \{y_j^{T,n}\}_{j=1}^{N^{T,n}}$ with

¹⁸⁰ $y_j^{T,n} = (b_j^{T,n}, c_j^{T,n})$ and $c_j^{T,n} \equiv 1$, where $n \in \{1, \dots, N^I\}$ is the index of adaptation
¹⁸¹ iteration process. N^I is the maximum number of adaptation iterations. We can
¹⁸² formulate the combination of supervised loss and unsupervised regularizer as
¹⁸³ follows:

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

$$\begin{aligned} L_S &= \sum_{j=1}^{N^{T,n}} (R(\theta^{T,n} | x_j^{T,n}, b_j^{T,n}) + C(\theta^{T,n} | x_j^{T,n}, c_j^{T,n})) \\ &\quad + \sum_{i=f(j)}^{N^{T,n}} (R(\theta^{T,n} | x_i^S, b_i^S) + C(\theta^{T,n} | x_i^S, c_i^S)) \end{aligned} \quad (2)$$

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

¹⁹⁴ where L_S is supervised loss to learn the scene-specific detector and L_U is the
¹⁹⁵ unsupervised regularizer part. $R(\cdot)$ is a regression loss for bounding box location,
¹⁹⁶ such as norm-1 loss, and $C(\cdot)$ is a classification loss for bounding box confidence,
¹⁹⁷ such as cross-entropy loss. $f(\cdot)$ randomly returns an index under the condition
¹⁹⁸ that $x_{f(\cdot)}^S$ must be a negative sample. And $L_{EWM}(\cdot)$, to be introduced in Section
¹⁹⁹ 3.2, is the MMD-based loss added on the element-wise multiply layer for unsu-
²⁰⁰ pervised regularization. α is the coefficient balancing the effect of supervised and
²⁰¹ unsupervised loss.

203 3.1 Iterative Algorithm

²⁰⁵ There are two reasons to employ iterative algorithm. Firstly, auto-annotated da-
²⁰⁶ ta on target domain change for every adaptation iteration and new samples will
²⁰⁷ be auto-annotated as positive instances. Compared to methods without iterative
²⁰⁸ algorithm, it helps to avoid overfitting caused by lack of data. Besides, unsuper-
²⁰⁹ vised regularizer performs better with more training data as it's a distribution
²¹⁰ based regularizer.

²¹¹ There are two stages for iterative algorithm. Source stream and target stream
²¹² are separately trained at different stages. At initialization stage, source model of
²¹³ source stream are trained under supervised loss function with abundant labelled
²¹⁴ data, \mathbf{X}^S , from source domain. After its convergence, the weights of source mod-
²¹⁵ el θ^S are taken to initialize target stream. At adaptation stage, target model is
²¹⁶ trained by auto-annotated data under both supervised loss function and unsuper-
²¹⁷ vised regularizer. Note that we do not jointly train two streams at adaptation
²¹⁸ stage and the weights of source model stays static which is served as a distri-
²¹⁹ bution reference for unsupervised regularizer at adaptation stage. The complete
²²⁰ adaptation process is illustrated in Algorithm 1. After a predetermined iteration
²²¹ limit N^I is reached, we obtain our final detection model on the target domain.

²²² Different from source domain, we use auto-annotation tools to auto-annotate
²²³ people instances with high confidence as training set on target domain. At first
²²⁴ iteration, we take source model as the auto-annotation tool. For subsequent

Algorithm 1 Deep domain adaptation algorithm (to be completed)

```

225 1: procedure DEEP DOMAIN ADAPTATION
226 2: Train source model on source stream with abundant annotated data
227 3: Use well-trained source model on source stream to initialize model on target stream
228 as  $M_0$ 
229 4:   for i = 0: $N^I$  do
230      $M_i$  generate "fake ground truth"  $G_i$  of target domain
231     balbal
232     balbabla
233      $G_i = G_i +$  random samples from source domain
234     Take  $G_i$  as training data to upgrade  $M_i$  into  $M_{i+1}$ 
235 10:   end for
236 11:  $M_{N^I}$ : final model.
237 12: end procedure
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269

```

adaptation iterations, updated target model from last adaptation iteration are utilized as the auto-annotation tool. During every adaptation iteration, the auto-annotation tool (also the target model) will be updated. Thus, the training samples for target domain at n^{th} adaptation iteration may differ from that at $(n + 1)^{th}$ adaptation iteration.

Note that these auto-annotated data are all regarded as positive samples. That is, we encourage the network to have more confidence on positive instances and stay conservative toward negative instances. This policy will no doubt resulting predictions of many non-people instance and impeding the further training when many non-people instances are regarded as people instances by the target model. To compensate for lack of negative instances, equal amount of negative samples are randomly selected from target domain, which are human annotated and can thus provide true negative samples.

3.2 Unsupervised weights regularizer on Element-wise Multiply Layer

Element-wise Multiply Layer In deep neural network, the last feature vector 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

270 operation of full connected layer can be thus formulated as matrix multiply:

271 $\mathbf{P} = \mathbf{F} * \mathbf{C}$ (4)

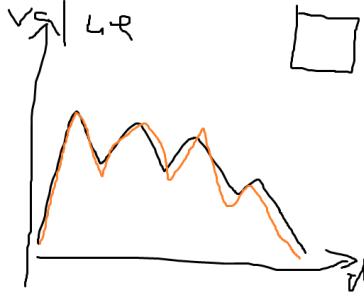
272 $P_{b,o} = \sum_d F_{b,d} * K_{d,o}$ (5)

273 Inspired by this form, we separate the above formula into two sub-operations –
274 element-wise multiply and sum, which can be formulated as:
275

276 $M_{b,o,d} = F_{b,d} * C_{d,o}$ (6)

277 $P_{b,o} = \sum_d M_{b,o,d}$ (7)

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



302 **Fig. 2.** One kernel at x_s (dotted kernel) or two kernels at x_i and x_j (left and right)
303 lead to the same summed estimate at x_s .

304

305

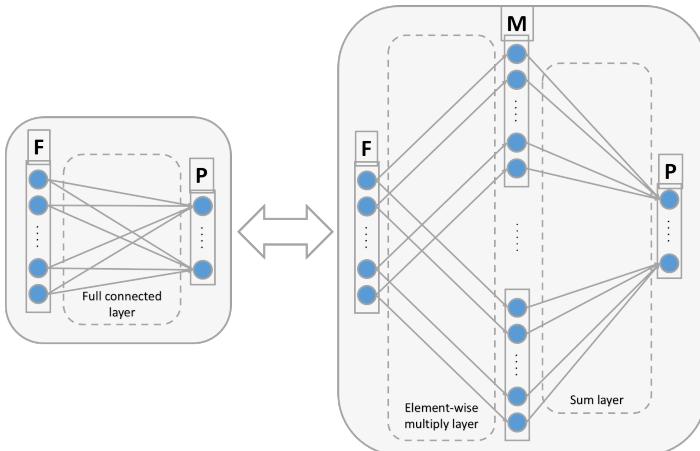
306

307 **Unsupervised weights regularizer on Element-wise Multiply Layer** In
308 works [decaf][], the last feature vector layer are regarded as the final representa-
309 tion of images. (how to introduce beyond sharing weights) In domain adaptation
310 tasks, when generic deep model are trained with abundant data from source do-
311 main, the last element-wise multiply layer mentioned in Sec 3.2 also includes
312 rich information leading to the final output. For the nodes in the output layer,
313 inputs from element-wise multiply layers that will contribute to its value are
314 not randomly decided. In this paper, we assume that the distribution of $\mathbf{m}_{b,o}$ of

315 the last element-wise multiply layer on both source and target domain should
 316 be similar. In the last element-wise multiply layer, every dimension of $\mathbf{m}_{b,o}$ may
 317 contribute to the final output. However, as the old model on source domain
 318 are trained with abundant images, the distribution of $\mathbf{m}_{b,o}^T$ should be consistent
 319 compared with $\mathbf{m}_{b,o}^S$ of source domain when adapting deep network on target
 320 domain. We utilize MMD (maximum mean discrepancy) to encode the similarity
 321 of element-wise multiply layer of source domain and target domain:

$$323 \quad L_{MMD}(\theta^T | \mathbf{X}^S, \mathbf{X}^T, \theta^S) = \frac{1}{N^O} \sum_{o=1}^{N^O} \left\| \frac{1}{N^B} \sum_{b=1}^{N^B} \mathbf{m}_{b,o}^T - \frac{1}{N^B} \sum_{b=1}^{N^B} \mathbf{m}_{b,o}^S \right\|^2 \quad (8)$$

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



349 **Fig. 3.** xxxxxxxxxxxxxxxxx

354 3.3 Detection Network

355 The generic model ¹ used in our adaptation architecture for source and tar-
 356 get stream is an end to end detection network without any precomputed region
 357 proposals needed. It consists of a GoogLeNet [13] for feature extraction and a

359 ¹ Proposed by Russel et al.

RNN-based decoder for output of bounding box and corresponding confidence, as shown in Fig x(!). Firstly, the GoogLeNet model encode the image into a feature map (15x20x1024) of which each 1024 dimension vector are data representation of its receptive field corresponding to a subregion of the image. Then the RNN-based layers with batch size 300 will decode the data representation and sequentially predict 5 possible bounding boxes by the order of its corresponding confidence. Finally, all outputs are summarized to give final detection results. When trained with abundant samples from source domain, the obtained model has a high precision on target domain, however, its recall is low. Also, different from other detection network [14, 15], which need precomputed proposals for classification and fine regression, this generic model directly predict bounding boxes with high confidence. Thus, negative instances may contain people and non-people predictions, and cannot be employed in adaptation training.

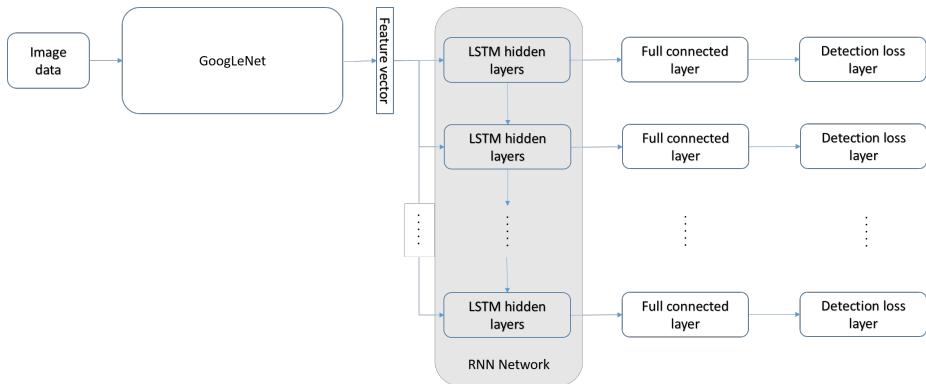


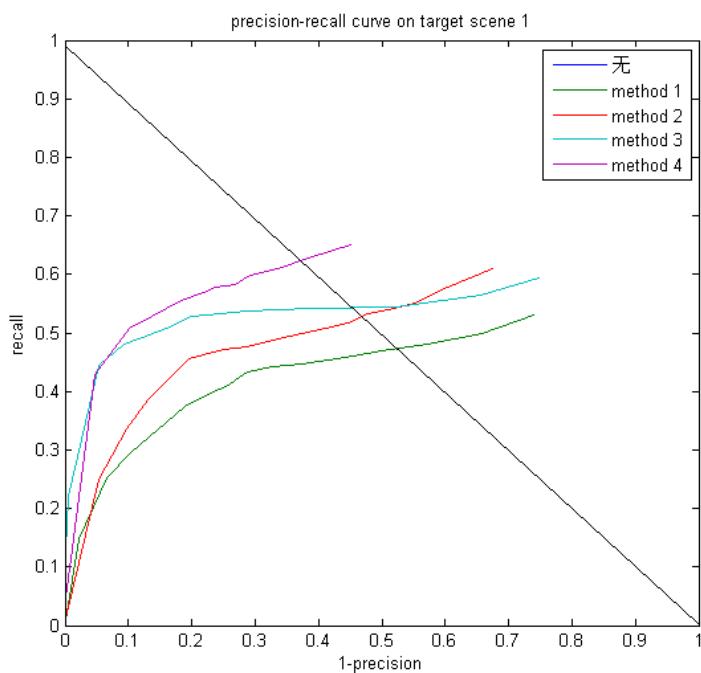
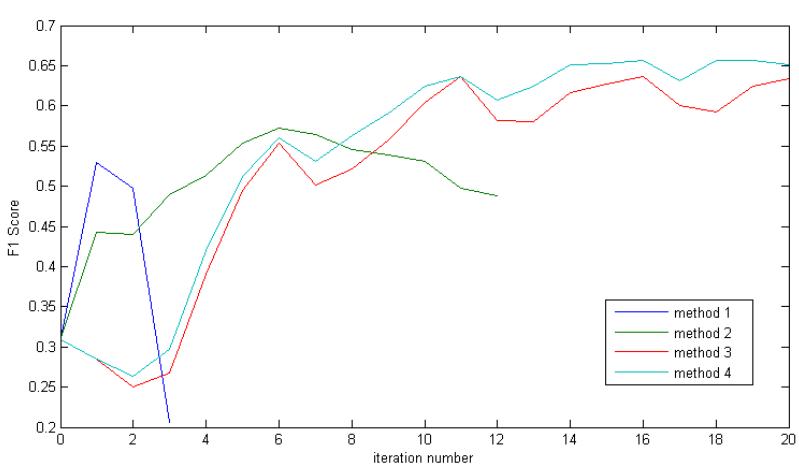
Fig. 4. xxxxxxxxxxxxxxxxx

4 Experiment Results

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

4.1 Domain Adaptation on Crowd Dataset

Dataset and evaluation metrics To show the effectiveness of our domain adaptation approach on people detection, we collected a dataset consisting of 3 target scenes for target domain. These three scenes contain 1308, 1213 and

429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
Fig. 5. xxxx448
449
Fig. 6. xxxx

450 331 un-annotated images with 0000, 0000, 0000 people instances respectively.
 451 For each scene, 100 images are annotated for evaluation. Instead of labelling the
 452 whole body of a person, we labels the head of a person as bounding box during
 453 training. The motivation for labelling only people heads comes from detection of
 454 indoor people or in crowded scenes, where the body of a person may be invisible.
 455 The dataset for source domain are Brainwash Dataset released on [http://d2.mpi-](http://d2.mpi-inf.mpg.de/datasets)
 456 [inf.mpg.de/datasets](http://d2.mpi-inf.mpg.de/datasets). Brainwash Dataset consists of over 11917 images from 3
 457 crowded scenes. Examples of images from source and target domain are shown
 458 in Fig x(!).

459 Our evaluation metrics for detection uses the protocol defined in PASCAL
 460 VOC [16]. To judge a predicted bounding box whether correctly matches a
 461 ground truth bounding box, their intersection over their union must exceed 50%.
 462 And Multiple detections of the same ground truth bounding box are regarded as
 463 one correct prediction. We plot the precision-recall curve in Fig x(!). Also, the
 464 F1 score $F1 = 2 * precision * recall / (precision + recall)$ during the adaptation
 465 process are also shown in Fig x(!).

466
 467 **Experimental settings** We use deep learning framework Caffe [17] as the
 468 adaptation architecture of our approach. During the adaptation, we set learning
 469 rate as 0.01 and momentum as 0.5. At initialization stage, GoogLeNet weights
 470 are firstly used to initialize source model of source stream, while parameters in
 471 RNN layers are randomly initialized from a uniform distribution. For each iteration,
 472 100 auto-annotated images from target domain and 1000 annotated images
 473 from source domain are alternatively used for training. The outputs of our de-
 474 tector network contains bounding box locations and corresponding confidence,
 475 thus there are two full connected layers between the last feature vector layer
 476 and the final outputs. Experiments of unsupervised regularizer on element-wise
 477 multiply layer for bounding box regression have no additional improvement on
 478 the performance when regularizer on element-wise multiply layer for box confi-
 479 dence classification are added already. Our approach on domain adaptation are
 480 executed separately in the 3 target scenes.

481
 482 **Comparison with baseline methods** To demonstrate the effectiveness of our
 483 approach, 4 methods are compared with method 4 as our final approach:

- 484 1. Only auto-labeled samples on target domain are used for training, and with-
 485 out any unsupervised regularizer.
- 486 2. Only auto-labeled samples on target domain are used for training, with M-
 487 MD regularizer on last element-wise multiply layer as unsupervised weights
 488 regularizer.
- 489 3. Both auto-labeled images from target domain and labeled images from source
 490 domain are alternately sampled for training, with MMD regularizer on last
 491 feature vector as unsupervised weights regularizer.
- 492 4. Both auto-labeled images from target domain and labeled images from source
 493 domain are alternately sampled for training, with MMD regularizer on last
 494 element-wise multiply layer as unsupervised weights regularizer.

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

	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	0.015	0.683	0.807	0.035	0.412	0.577
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

Performance evaluation From the Table x(!) and Fig x(1), we have the following observations:

- The recall values of method 1,2,3,4, which all utilized iterative algorithm to upgrade the target model, are larger than that of method 0, which are source model trained on source domain. This implies the effectiveness of our iterative 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 multiply 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.



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

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

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 iterative algorithm mentioned in Sec 3.1 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 multiply layer transformed 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 iterative algorithm and MMD regularizer on the element-wise multiply layer transformed from the last full connect layer.

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

	$A \rightarrow W$	
	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
Ours	85.4	69.3



Fig. 8. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)

- 630 2. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain
631 adaptation using asymmetric kernel transforms. In: Computer Vision and Pattern
632 Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1785–1792
633 3. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An
634 unsupervised approach. In: Computer Vision (ICCV), 2011 IEEE International
635 Conference on, IEEE (2011) 999–1006
636 4. Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J.: Correcting
637 sample selection bias by unlabeled data. In: Advances in neural information
638 processing systems. (2006) 601–608
639 5. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.:
640 Covariate shift by kernel mean matching. Dataset shift in machine learning **3**(4)
641 (2009) 5
642 6. Wang, X., Wang, M., Li, W.: Scene-specific pedestrian detection for static video
643 surveillance. Pattern Analysis and Machine Intelligence, IEEE Transactions on
644 **36**(2) (2014) 361–374
645 7. Zeng, X., Ouyang, W., Wang, M., Wang, X.: Deep learning of scene-specific class-
646 ifier for pedestrian detection. In: Computer Vision–ECCV 2014. Springer (2014)
647 472–487
648 8. Hattori, H., Naresh Boddeti, V., Kitani, K.M., Kanade, T.: Learning scene-specific
649 pedestrian detectors without real data. In: Proceedings of the IEEE Conference
650 on Computer Vision and Pattern Recognition. (2015) 3819–3827
651 9. Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I.J., Lavoie,
652 E., Muller, X., Desjardins, G., Warde-Farley, D., et al.: Unsupervised and transfer
653 learning challenge: a deep learning approach. ICML Unsupervised and Transfer
654 Learning **27** (2012) 97–110
655 10. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discrim-
656 inatively learning domain-invariant features for unsupervised domain adaptation.
657 In: Proceedings of The 30th International Conference on Machine Learning. (2013)
658 222–230
659 11. Ghifary, M., Kleijn, W.B., Zhang, M.: Domain adaptive neural networks for object
660 recognition. In: PRICAI 2014: Trends in Artificial Intelligence. Springer (2014)
661 898–904
662 12. Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormählen, T., Schiele, B.:
663 Learning people detection models from few training samples. In: Computer Vision
664 and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1473–
665 1480
666 13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D.,
667 Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings
668 of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
669 14. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on
670 Computer Vision. (2015) 1440–1448
671 15. Vu, T.H., Osokin, A., Laptev, I.: Context-aware cnns for person head detection.
672 In: Proceedings of the IEEE International Conference on Computer Vision. (2015)
673 2893–2901
674 16. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisser-
675 man, A.: The pascal visual object classes challenge: A retrospective. International
676 Journal of Computer Vision **111**(1) (2015) 98–136
677 17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama,
678 S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding.
679 arXiv preprint arXiv:1408.5093 (2014)

18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
 19. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2066–2073
 20. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 2960–2967
 21. Tommasi, T., Caputo, B.: Frustratingly easy nbnn domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 897–904
 22. Chopra, S., Balakrishnan, S., Gopalan, R.: Dlid: Deep learning for domain adaptation by interpolating between domains. In: ICML workshop on challenges in representation learning. Volume 2. (2013)
 23. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)