2PCNet: Two-Phase Consistency Training for Day-to-Night
Unsupervised Domain Adaptive Object Detection

Mikhail Kennerley\textsuperscript{1,2}, Jian-Gang Wang\textsuperscript{2}, Bharadwaj Veeravalli\textsuperscript{1}, and Robby T. Tan\textsuperscript{1}
\textsuperscript{1}National University of Singapore, Department of Electrical and Computer Engineering
\textsuperscript{2}Institute for Infocomm Research, A*STAR
mikhailk@u.nus.edu, jgwang@i2r.a-star.edu.sg, elebv@nus.edu.sg, robby.tan@nus.edu.sg

Abstract

Object detection at night is a challenging problem due to the absence of night image annotations. Despite several domain adaptation methods, achieving high-precision results remains an issue. False-positive error propagation is still observed in methods using the well-established student-teacher framework, particularly for small-scale and low-light objects. This paper proposes a two-phase consistency unsupervised domain adaptation network, 2PCNet, to address these issues. The network employs high-confidence bounding-box predictions from the teacher in the first phase and appends them to the student’s region proposals for the teacher to re-evaluate in the second phase, resulting in a combination of high and low confidence pseudo-labels. The night images and pseudo-labels are scaled-down before being used as input to the student, providing stronger small-scale pseudo-labels. To address errors that arise from low-light regions and other night-related attributes in images, we propose a night-specific augmentation pipeline called NightAug. This pipeline involves applying random augmentations, such as glare, blur, and noise, to daytime images. Experiments on publicly available datasets demonstrate that our method achieves superior results to state-of-the-art methods by 20%, and to supervised models trained directly on the target data.\textsuperscript{1}

1. Introduction

Nighttime object detection is critical in many applications. However, the requirement of annotated data by supervised methods is impractical, since night data with annotations is few, and supervised methods are generally prone to overfitting to the training data. Among other reasons, this scarcity is due to poor lighting conditions which makes nighttime images hard to annotate. Hence, methods that do not assume the availability of the annotations are more advantageous. Domain adaptation (DA) is an efficient solution to this problem by allowing the use of readily available annotated source daytime datasets.

1\textsuperscript{www.github.com/mecarill/2pcnet}
The effectiveness of our approach has been verified by comparing it with the state-of-the-art domain adaptation approaches. An improvement of +7.9AP(+20%) and +10.2AP(26%) over the SOTA on BDD100K and SHIFT has been achieved, respectively.

2. Related Work

Unsupervised Domain Adaptation (UDA) Unsupervised domain adaptation aims to learn transferable features to reduce the discrepancy between a labelled source and unlabelled target domain. Previous works minimised the distance metric (MMD) [16–18] and considered intra-class and inter-class discrepancy [12, 13]. Adversarial feature learning involved adding an adversarial classifier to play the min-max game between the domain discriminator and feature extractors to generate a domain invariant feature map [27, 28, 37]. These methods have been applied to image classification. Our work focuses on object detection, which is more complex as it involves identifying multiple objects in an image.

UDA for Object Detection Object detection with UDA is a recent challenge due to the complexities of identifying multiple objects in an image. DA-Faster RCNN [3] integrated adversarial learning with image and instance level classifiers, and several approaches have been proposed to improve on this method by introducing scale-awareness [4], class specific discriminators [31], and re-purposing the task-specific classifier as a discriminator [2]. The Mean Teacher (MT) framework [26] has been adopted in semi-supervised methods, such as UMT [7], which incorporates CycleGAN [39] augmented images; AT [15], which combines the student-teacher framework with adversarial learning; and TDD [11], which uses dual student-teacher networks with style transfer.

Nighttime UDA The majority of research on unsupervised domain adaptation (UDA) in nighttime scenarios has focused on semantic segmentation [5, 8, 9, 14, 23, 29, 33]. Translation and style transformation techniques are commonly used to reduce the domain gap between the source and target domains in these methods [8, 29, 33]. Some UDA-based techniques for nighttime also utilise paired-images to generate a shared feature space [23], while others use an intermediate domain such as twilight to reduce the domain gap during unsupervised learning [5].

Nighttime tracking has also been investigated where adversarial transformers are used to close the domain gap [35]. However, there is a gap in research when it comes to applying UDA techniques in the object detection task for nighttime.
time scenarios. Therefore, we explore the application of UDA techniques in object detection under low-light and nighttime conditions.

3. Proposed Method

Let $D_s$ be the daytime source data. $D_s = \{I_s, C_s, B_s\}$, where the variables refer to the image, class label and bounding-box label, respectively. Index $s$ indicates the daytime source. The night target data is represented by $D_t = \{I_t\}$ as we do not have the target labels available to us. Index $t$ indicates the nighttime target.

The architecture of our 2PCNet is shown in Figure 2. Our 2PCNet consists of a student and a teacher network. The student is a multi-domain network trained on both labelled daytime images, augmented with NightAug, and unlabelled nighttime images. The teacher focuses on night images to produce pseudo-labels for the student and is the exponential moving average (EMA) of the student. After an initial pretraining phase, the teacher begins producing pseudo-labels, which allows the student to initialise the feature extractor and detector.

During each iteration, in phase one of 2PCNet, the teacher produces pseudo-labels from the night images. These pseudo-labels are filtered through a confidence threshold. This is to ensure only high-confidence pseudo-labels are given to the student. The bounding-boxes from the pseudo-labels are then combined with the region proposals generated by the student’s RPN. The merged region proposals are then used to generate predictions from the student’s RoI network. In phase two, the teacher utilises the same merged region proposals to generate a matched set of pseudo-labels, where each pseudo-label has its corresponding prediction obtained from the student.

As mentioned earlier, our student network is initialised by pretraining for a set number of iterations. This is done with supervised loss on the augmented daytime images:

$$L_{\text{sup}} = L_{\text{rpn}}(B_s, I_s) + L_{\text{roi}}(B_s, C_s, I_s),$$

where $L_{\text{rpn}}$ represents the loss from the RPN, which consists of an objectness and bounding-box regression loss. $L_{\text{roi}}$ represents the loss from the detector network, consisting of a classification and bounding-box regression loss.

Once the pretraining is completed, the student’s weights are then transferred over to the teacher. In the succeeding iterations, the teacher’s weights are the exponential moving average (EMA) of the student’s. The matched pseudo-labels generated by the teacher, $\{C^*_p, B^*_p\}$, are then used to guide
In our case, these errors notably occur in dark/glare regions. Consequently, the error continues to propagate through training. The teacher begins to predict highly confident yet incorrect pseudo-labels. As the teacher provides these incorrect pseudo-labels to the student, a viscous cycle starts where the student accepts these pseudo-labels, and the teacher begins to predict highly confident yet incorrect pseudo-labels to be distilled onto the student. In phase one, the unlabelled nighttime image, $I_t$, is used as an input for the teacher to generate pseudo-labels. These pseudo-labels are filtered with a threshold to retain only high-confidence pseudo-labels, $(C_p, B_p)$. The bounding-box of the pseudo-labels, $B_p$, is then used as an input to the student. $B_p$ is concatenated to the region proposals generated by the student RPN module:

$$P^* = \text{RPN}_{\text{student}}(I_t) + B_p, \quad (3)$$

where $P^*$ is the combined region proposals, which are then used as an input to the student’s RoI module to predict the classes, $C_{\text{student}}$, and bounding-box, $B_{\text{student}}$, of each region proposal.

Phase two begins by using the same combined region proposals, $P^*$, generated in phase one as an input to the teachers RoI module to generate a matched set of pseudo-labels:

$$\{C^*_p, B^*_p\} = \text{RoI}_{\text{teacher}}(P^*). \quad (4)$$

The difference between $C_p$ and $C^*_p$ is that $C^*_p$ is derived from the same region proposals as that of the student predictions $C_{\text{student}}$. This allows us to compare $C_{\text{student}}$ and $C^*_p$ directly:

$$\{C_{\text{student}}(n), B_{\text{student}}(n)\} = \text{RoI}_{\text{student}}(P^*(n)),$$

$$\{C^*_p(n), B^*_p(n)\} = \text{RoI}_{\text{teacher}}(P^*(n)), \quad (5)$$

where $n = \{1, 2, \ldots, N\}$ and $N$ is the number of region proposals in $P^*$. This operation ensures that the knowledge of highly confident predictions generated by the teacher is distilled through to the student. In addition, information from less confident predictions can also be learnt. However, we are still required to penalise less confident samples and thus employ weighed KL-Divergence to be used as our consistency loss:

$$L_{\text{cons}} = \alpha \cdot \text{KL}(C_{\text{student}}, C^*_p), \quad (6)$$

where $\alpha$ is the highest confidence of $C^*_p$ expressed as $\alpha = \max(C^*_p)$; $\text{KL}(\cdot)$ is the KL-divergence function. Note that, pseudo-bounding boxes are not used to generate unsupervised loss, as the confidence score of each pseudo-label represents the class information rather than the bounding box. The outputs of each segment of our two-phase approach are shown in Figure 3.

3.2. Student-Scaling

In our investigation, we have found that scales of objects have a strong influence on object detection at night. This
Algorithm 1 Single Augmentation - NightAug

Algorithm:

```
imgClean ← img
if randFloat ≥ 0.5 then
    randFloat ← 0.8 * randFloat + 0.2
    img ← augmentation(img, randval)
    prob ← 0.4
while randFloat ≥ prob do
    x ← randInt(img.shape[1], 2)
    y ← randInt(img.shape[2], 2)
    img[x, y] ← imgClean[x, y]
    prob ← prob + 0.1
end while
end if
```

is due to the features of smaller objects being easily overwhelmed by glare or noise. To allow the student to overcome this, we apply scaling augmentation to the student’s inputs which includes both the image and the pseudo-labels generated by the teacher. As training proceeds, we follow a schedule to increase the scale of the student augmentation until it equals to that of the original image. By iteratively increasing the scale we allow the student to focus on smaller features earlier in the training process. This process encourages the teacher to make more accurate predictions on smaller scale objects in the later stages of training. In turn, accurate small scale pseudo-labels allow for the increase in the scale of the student’s inputs with minimal errors due to scale.

To ensure the knowledge of the previous scales is not forgotten, a gaussian function for the scaling factor is applied. The norm of the Gaussian function is obtained from the schedule values. To prevent additional noise due to pseudo-labels being too small, labels that has an area below a threshold are removed.

3.3. NightAug

Night images suffer from a range of complications that are not present in daytime scenes. This causes a problem in the student-teacher framework, where the student would be biased towards the source domain. Previous methods have attempted to address this, but have either required compute-intensive translations [7, 11] or adding additional domain classifiers to the framework [15] which complicates training. We propose NightAug, a nighttime specific augmentation pipeline that is compute-light and does not require training. NightAug consists of a series of augmentations with the aim of steering the characteristics of daytime images to resemble that of a nighttime image.

The defining features of nighttime images are that they are darker and have lower contrast than daytime images. In addition the signal-to-night ratio (SNR) could be higher due to the properties of digital cameras such as luminance and colour noise. Glare and glow from street lamps and headlights are also present in nighttime images. Additionally, images may be out-of-focus due to the camera’s inability to detect reference points to focus on in dark environments.

Keeping in mind the properties of nighttime images, our NightAug includes random; brightness, contrast, gamma, gaussian noise, gaussian blur augmentations and random glare insertion. The augmentations are randomly applied to the images and are also random in intensity. This randomness results in a wider variance of images that are exposed to the student leading to more robust training [30]. To further increase the variance of the images, at each augmentation step, random segments of the image will ignore the application of that augmentation. This allows for the representation where different areas of nighttime images may be unevenly lighted. This uneven lighting affects the above characteristics of the local region.

A single augmentation flow of NightAug is demonstrated in Algorithm 1. Samples of an image processed with NightAug are shown in Figure 4. Each augmentation has a set probability of being applied, with the strength of the augmentation being random. Random regions of the augmented image may then be replaced with that of the original image. The probability of this region replacement reduces with each iteration.

Overall Loss

Our total loss can be represented as:

\[ L_{\text{total}} = L_{\text{sup}} + \lambda L_{\text{unsup}}, \]

where \( \lambda \) represents a weight factor for the unsupervised loss, and is set experimentally. \( L_{\text{sup}}, L_{\text{unsup}} \) refer to Eq. (1) and Eq. (2), respectively.
Method | AP | Pedestrian | Rider | Car | Truck | Bus | Motor cycle | Bicycle | Traffic Light | Traffic Sign
--- | --- | --- | --- | --- | --- | --- | --- | --- | --- | ---
Lower-Bound | 41.1 | 50.0 | 28.9 | 66.6 | 47.8 | 47.5 | 32.8 | 39.5 | 41.0 | 56.5
Upper-Bound | 46.2 | 52.1 | 35.0 | 73.6 | 53.5 | 54.8 | 36.0 | 41.8 | 52.2 | 63.3
DA F-RCNN [3] | 41.3 | 50.4 | 30.3 | 66.3 | 46.8 | 48.3 | 32.6 | 41.4 | 41.0 | 56.2
TDD [11] | 34.6 | 43.1 | 20.7 | 68.4 | 33.3 | 35.6 | 16.5 | 25.9 | 43.1 | 59.5
UMT [7] | 36.2 | 46.5 | 26.1 | 46.8 | 44.0 | 46.3 | 28.2 | 40.2 | 31.6 | 52.7
AT [15] | 38.5 | 42.3 | 30.4 | 60.8 | 48.9 | 52.1 | 34.5 | 42.7 | 29.1 | 43.9
2PCNet (Ours) | 46.4 | 54.4 | 30.8 | 73.1 | 53.8 | 55.2 | 37.5 | 44.5 | 49.4 | 65.2

Table 1. Results of day-to-night domain adaptation on the BDD100K dataset, the Average Precision (AP) of all classes are reported. Faster RCNN detector with ResNet-50 feature extractor is used for all experiments to ensure a fair comparison. Faster RCNN is used as the lower-bound and upper-bound and is trained on labelled daytime and nighttime data respectively. The lower-bound provides a baseline without any domain adaptation while the upper-bound is fully supervised, the case where labelled target night data is available.

Method | AP\_coco | Car | Bus | Truck
--- | --- | --- | --- | ---
Lower-Bound | 22.1 | 37.5 | 29.8 | 30.7
Upper-Bound | 23.9 | 42.0 | 33.8 | 35.0
FDA [34] | 22.6 | 38.5 | 37.2 | 23.2
ForkGAN [38] | 22.9 | 41.2 | 33.3 | 32.1
2PCNet (Ours) | 23.5 | 40.7 | 38.2 | 35.0

Table 2. Comparison of our framework, 2PCNet, with image-to-image (I2I) translation methods. Conducted on the BDD100K dataset. ForkGAN and FDA are used for comparison. Reported AP\_coco is the averaged AP over IoUs 0.5 to 0.95.

4. Experiments

4.1. Baselines

To evaluate our method, we compare our approach with SOTA methods in domain adaptation for object detection. These include DA-Faster RCNN [3], TDD [11], UMT [7], AT [15] as well as a non-DA baseline Faster-RCNN [21]. Faster-RCNN is used as both our lower and upper-bound, where it is trained on labelled source and target data respectively. We additionally compare our approach with image-to-image translation methods, ForkGAN [38] and FDA [34]. Translation methods are trained on Faster RCNN with both the daytime and translated images.

4.2. Datasets

The majority of existing nighttime datasets either focuses on semantic segmentation which do not provide labels for object detection [5, 23, 24], or contains very few classes [19, 20]. BDD100K [36] was selected as it provides object detection labels which includes a wide range of classes (10). It also has a large number of images compared to other DA datasets covering daytime, nighttime and other adverse conditions.

The SHIFT [25] dataset is a recent simulated driving dataset that contains scenes in various environments. A continuous shift of these environments is available. SHIFT contains 6 class labels that share similarities to the BDD100K classes. For our evaluation, we use images with the ‘day’ and ‘night’ label as our source and target data respectively. We further ensure that the weather tag is ‘clear’ to isolate other weather conditions from the evaluation.

4.3. Implementation

Following previous SOTA methods, we employ Faster-RCNN [21] as our base detection model and ResNet-50 [10] pretrained on ImageNet [6] as our feature extractor. All images are scaled by resizing its shorter side to 600 pixels. For student-scaling we set a schedule for (0.57, 0.64, 0.71, 0.78, 0.85, 0.92) of the maximum iterations at scales (0.5, 0.6, 0.7, 0.8, 0.9, 1.0). Loss hyperparameters are set at $\lambda = 0.3$ and the rate smooth coefficient parameter of the EMA is 0.9996. A confidence threshold of 0.8 for phase one of Two-Phase Consistency. For the initial pretraining of the student model, we train the student for 50k and 20k iterations on the source images, for BDD100K and SHIFT respectively. Supervised inputs are daytime images with and without NightAug. We then copy the weights to the teacher and continue training with the addition of unsupervised loss for an additional 50k iterations. The learning rate is kept at 0.04 throughout training. Our network is trained on 3 RTX3090 GPUs with a batch-size of 6 source and 6 target images.

4.4. Comparison to SOTA

Comparison on BDD100K We compare our method against the SOTA on real driving scenes and evaluating their domain adaptation performance on nighttime images, the results of this experiment can be seen on Table 1. The results show that our method achieves the highest perfor-
Figure 5. Qualitative results of Faster RCNN, Adaptive Teacher (AT) and our method on the SHIFT dataset with the ground-truth on the far right. We can observe that Faster RCNN is not able to detect objects due to absence of domain adaptation, while AT has a large number of small false positive bounding boxes compared to our method which closely resembles that of the ground-truth.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower-Bound</td>
<td>41.6</td>
<td>40.4</td>
<td>44.5</td>
<td>49.9</td>
<td>53.7</td>
<td>14.3</td>
<td>46.7</td>
</tr>
<tr>
<td>Upper-Bound</td>
<td>47.0</td>
<td>49.7</td>
<td>51.5</td>
<td>56.0</td>
<td>53.6</td>
<td>19.2</td>
<td>52.4</td>
</tr>
<tr>
<td>DA FR [3]</td>
<td>43.7</td>
<td>43.0</td>
<td>48.8</td>
<td>47.8</td>
<td>52.1</td>
<td>19.9</td>
<td>55.8</td>
</tr>
<tr>
<td>UMT [7]</td>
<td>31.1</td>
<td>7.7</td>
<td>47.5</td>
<td>18.4</td>
<td>46.8</td>
<td>16.6</td>
<td>49.2</td>
</tr>
<tr>
<td>AT [15]</td>
<td>38.9</td>
<td>25.8</td>
<td>33.0</td>
<td>54.7</td>
<td>49.5</td>
<td>20.7</td>
<td>52.3</td>
</tr>
<tr>
<td>2PCNet (Ours)</td>
<td>49.1</td>
<td>51.4</td>
<td>54.6</td>
<td>54.8</td>
<td>56.6</td>
<td>23.9</td>
<td>54.2</td>
</tr>
</tbody>
</table>

Table 3. Results of Day-to-Night domain adaptation on the SHIFT dataset. The Average Precision (AP) of all classes. Faster RCNN is used as the lower-bound and upper-bound and is trained on labelled daytime and nighttime data respectively.

mance with an AP of 46.4, 20.5% higher than that of the SOTA student-teacher methods and above that of the upper-bound. We have observed in experiments that student-teacher methods underperforms with an AP below that of the lower-bound due to the error-propagation from noisy pseudo-labels. The result of the error is small false positive detections as seen in Figure 1. Our method does not suffer from the same allowing for higher performance. We can also observe that our method performs well across all classes. Even when compared with the upper-bound, 2PCNet achieves higher AP on the majority of classes. This indicates that our method is able to generalise well across large and small classes.

The comparison with image-to-image translation methods is shown in Table 2. Translation methods do not suffer from the error propagation problem as it is trained on Faster RCNN without a teacher. Even so, we can see that our method outperforms SOTA adverse vision translation methods.

Comparison on SHIFT To further compare our method with SOTA we evaluate on the SHIFT simulation dataset. Due to the nature of the simulated data, many nighttime image characteristics that we have previously mention is not exhibited in this data such as blurriness, noise and glare.

The results of this experiments are shown in Table 3. We can observe that previous SOTA methods that use the student-teacher framework perform worse than the lower-bound. The sub-par performance is again due to the error-propagation problem. AT performs better than UMT due to ATs inclusion of adversarial learning. However, adversarial learning is not enough to mitigate this problem. We can see that the performance of DA FRCNN outperforms both the SOTA student-teacher methods as it would not be affected by error-propagation. It is however, still largely below the upper-bound performance. 2PCNet outperforms these previous methods as well as the upper-bound. We achieve an improvement of +10.2 AP over previous SOTA student-teacher methods and +2.1 AP over that of the upper-bound.

4.5. Ablation Studies

To demonstrate the effectiveness of each of our components, we train several models for 100K iterations and evaluate them on the BDD100K dataset. We present our findings in Table 4.

Two-Phase Consistency We can observe in Table 4 that the addition of Two-Phase Consistency (C) demonstrated a wide performance gap when compared to the Mean-Teacher baseline, +13.5 AP (43%). This improvement in AP ex-
ists across large, medium and small objects. While the performance of MT is initially strong, it rapidly begins to decline; which can be observed in Figure 6. This drop in performance is due to the error propagation of noisy pseudo-labels. The experimental results show that Two-Phase Consistency is able to provide a solution. This ensures that highly confident pseudo-labels are bounded by less confident pseudo-label enabling a balance of knowledge into the student.

NightAug We benchmarked the effectiveness of NightAug in our framework as shown in Table 4. The inclusion of NightAug increases the detection performance of small objects with an increase of 5%. Additionally, the gradient of the training performance remains steep as seen in Figure 6. The positive gradient is displayed most strongly for APm and APs where objects are more prone to nighttime specific complications.

Student-Scaling Our final component, student-scaling, is included into the framework and the results can be seen in Table 4. We can observe that student-scaling is able to boost the performance of small object detection by 6%. This boost in performance is due to the student network focusing on smaller object earlier in the training process. We note that the performance of large objects have dropped by 1-2%; however when referring to the training curves in Figure 6, APl remains steep. As the initial focus is on smaller objects, less time is allocated to larger objects during training. This can be mitigated by lengthening training resulting in more iterations for larger objects.

Domain Classifier To conclude our study, we included a domain classifier into our network. Adversarial learning is a widely used DA technique; however when added into 2PCNet, a performance drop across all scales can be seen. This drop is shown in Table 4. The suppression of nighttime features is suspected to be the cause. Suppression is present as the adversarial loss guides the feature extractor to maintain domain invariance. By suppressing nighttime features, the teacher has less information to distil to the student. This is demonstrated in Figure 6 where the domain classifier (dotted purple) initially performs well. But as training continues, our method (solid red) is able to surpass its performance.

5. Conclusion

Our proposed framework, 2PCNet, presents a novel solution to the challenges of day-to-night domain adaptive object detection. With our Two-Phase Consistency approach, we are able to effectively leverage high and low confidence knowledge for the student, while mitigating error propagation commonly present in previous student-teacher methods. We further address issues arising from small scale and dark objects through the use of student-scaling and NightAug, respectively. Experimental results on the BDD100K and SHIFT datasets demonstrate that 2PCNet outperforms existing state-of-the-art methods. Overall, our proposed framework provides an effective and efficient solution for day-to-night domain adaptive object detection.

Acknowledgements This work is partially supported by MOE2019-T2-1-130.
References


