Mixture Domain Adaptation to Improve Semantic Segmentation in Real-World Surveillance

Sébastien Piérard
University of Liège, Belgium
S.Pierard@uliege.be

Anthony Cioppa
University of Liège, Belgium
Anthony.Cioppa@uliege.be

Anaïs Halin
University of Liège, Belgium
Anais.Halin@uliege.be

Renaud Vandeghen
University of Liège, Belgium
R.Vandeghen@uliege.be

Maxime Zanella
University of Louvain-la-Neuve, Belgium
Maxime.Zanella@uclouvain.be

Benoît Macq
University of Louvain-la-Neuve, Belgium
Benoit.Macq@uclouvain.be

Saïd Mahmoudi
University of Mons, Belgium
Said.Mahmoudi@umons.ac.be

Marc Van Droogenbroeck
University of Liège, Belgium
M.VanDroogenbroeck@uliege.be

Abstract

Various tasks encountered in real-world surveillance can be addressed by determining posteriors (e.g. by Bayesian inference or machine learning), based on which critical decisions must be taken. However, the surveillance domain (acquisition device, operating conditions, etc.) is often unknown, which prevents any possibility of scene-specific optimization. In this paper, we define a probabilistic framework and present a formal proof of an algorithm for the unsupervised many-to-infinity domain adaptation of posteriors. Our proposed algorithm is applicable when the probability measure associated with the target domain is a convex combination of the probability measures of the source domains. It makes use of source models and a domain discriminator model trained off-line to compute posteriors adapted on the fly to the target domain. Finally, we show the effectiveness of our algorithm for the task of semantic segmentation in real-world surveillance. The code is publicly available at `https://github.com/rvandeghen/MDA`.

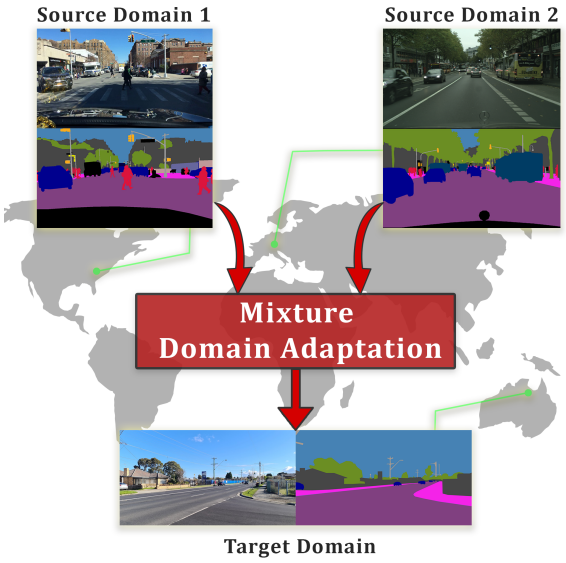

Figure 1. We present a theoretically motivated algorithm for unsupervised domain adaptation of posteriors on the fly. In particular, we express the target domain as a convex combination of the source domains. We show the effectiveness of our algorithm for the task of semantic segmentation in real-world surveillance.

1. Introduction

Applying artificial intelligence methods to real-world surveillance requires taking into account the specificity of the considered scene. In this paper, we address this issue by computing posteriors, which are probabilities related to the content of the scene. For example, for the task of semantic segmentation, the posteriors are the conditional probabilities of the semantic classes (the *hypotheses*) given a pixel from an image (an *evidence*).

There are three main reasons for focusing on posteriors: (1) they help interpret results compared to hard decisions (*e.g*. classifying an object with some degree of confidence rather than simply claiming it belongs to a class), (2) they enable optimal decisions for most common criteria (*e.g*. maximizing the posteriors for the accuracy or maximizing the likelihoods for the balanced accuracy), and (3) they allow to adapt to the specific domain of a scene, as demonstrated in this paper.

In general, the domains are rarely known in advance, which prevents any possibility of scene-specific optimization. Indeed, the domains depend on the choices made for the data acquisition (*e.g*. where the camera is installed or when the videos or images are captured) as well as on the distribution of scenarios that can occur (*e.g*. varying weather or lighting conditions). Furthermore, in some applications, the domains may change over time, making it even more challenging to take into account the specificity of the scene. This typically occurs when the surveillance cameras are onboard of moving vehicles.

When dealing with multiple domains, a training dataset can be collected in each domain. Common but oversimplistic assumptions are that the testing distribution (referred to as the *target domain*) is a fixed mixture of the various training datasets (referred to as the *source domains*) and that the mixture parameters are known in advance. Under these assumptions, it is reasonable to mix the various training datasets before training the models. This, however, is impossible in most real-world cases as the mixture parameters depend on the target domain which is unknown at learning time. An on-the-fly and unsupervised domain adaptation is therefore required for the sake of flexibility.

In this paper, we propose an algorithm that makes use of *source models* and a *domain discriminator model* trained off-line to obtain posteriors adapted on the fly to the target domain. We present a formal proof of this algorithm that is interpretable and runs in real time. Its effectiveness is shown experimentally for the task of semantic segmentation in real-world surveillance.

Contributions. Our contributions are as follows. **(i)** We define a probabilistic framework for a particular unsupervised mixture domain adaptation problem. **(ii)** Based on this framework, we present a formal proof of an unsupervised algorithm to adapt the posteriors on the fly for a changing target domain. **(iii)** We compare our algorithm with common posterior combination heuristics and show its superiority on several real-world surveillance datasets.

2. Related work

Hereafter, we present the related work first on unsupervised domain adaptation and second on semantic segmentation, with some references specific to the field of real-world surveillance.

Unsupervised domain adaptation. Domain adaptation aims at transferring knowledge from source domains to unseen target domains and alleviating the impact of domain shift in data distributions [1]. An even more challenging scenario arises when no label is available in the target domains; this is called unsupervised domain adaptation [2]. Domain shifts, which are defined as a change in the data distribution between the training distribution of an algorithm and the inference distribution, can be categorized into three main categories [1]: (1) *prior shift* (*a.k.a. target shift*), when the priors are different but the likelihoods are identical [3]; (2) *covariate shift*, when the marginal probability distributions differ between the source and target domains, but not the posteriors [4]; and (3) *concept shift*, when data distributions remain unchanged across source and target domains, while the posteriors differ between domains [5].

A common solution to unsupervised domain adaptation consists in adding a domain discriminator to learn domain-independent features [6]. This approach needs a new training phase, making real-time predictions unfeasible. Moreover, aligning the features can reduce their discriminative power [7]. A more restrictive scenario can be investigated, in which the source domain data cannot be accessed at test-time. Liang *et al*. [8] freeze the classifier and learn representations from the target domains aligned to the source hypothesis. Wang *et al*. [9] adapt the normalization layers while minimizing the prediction entropy. Boudiaf *et al*. [10] propose to keep the parameters of the model unchanged to avoid collapse in case of poorly diversified batches and only correct the predictions by encouraging neighboring representations to have consistent predictions.

Other works tackle the problem of domain shifts in the case of semantic segmentation. Zhang *et al*. [11] suggest to use a curriculum learning approach to learn more transferable global and local properties across domains. Li *et al*. [12] propose a bidirectional learning framework in which a translation module and a segmentation model are trained alternatively.

In the context of multi-source semantic segmentation, unsupervised domain adaptation has been studied with some approaches generating adapted domains [13, 14] or aligning their features distribution by matching their moments [15]. He *et al*. [16] propose to reduce the discrepancy between domains by aligning their pixel distribution before training models in a collaborative way to share knowledge between sources. One of the closest works to ours for binary classification is presented in [17]. The authors derive posteriors from Bayes' theorem by approximating priors with distances to each point in each source domain and likelihoods from nearest neighbor points. In our work, we address a multi-class case in a dense task, *i.e*. semantic segmentation. We further characterize the target domain as a mixture of the source domains.

Semantic segmentation. For the past decade, semantic segmentation has proven to be a powerful tool for global scene understanding [18, 19, 20]. To support the development of semantic segmentation networks, many annotated datasets emerged such as ADE20K [21, 22], PascalVOC2012 [23], COCO [24], or CityScapes [25]. Their availability has boosted the performance of algorithms over the years, leading to algorithms such as PSPNet [26], Mask R-CNN [27], PointRend [28], or SETR [29].

In the field of real-world surveillance, semantic segmentation can form the basis of complex downstream tasks such as urban scene characterization [30], maritime surveillance [31], water level estimation [32], low-light video enhancement [33], or super-resolution [34]. Due to the variety of scenes and downstream tasks, it is difficult to develop semantic segmentation algorithms that remain competitive without further scene-specific tuning. For instance, when placing a new surveillance camera, we need to ensure that the segmentation network will be able to perform well in this novel environment. Furthermore, the network must be robust to dynamic domain changes. Short-term changes may include illumination changes, sudden heavy rains, or different occupancy during rush hours, while long-term changes may include day-lasting heavy snow during winter or road constructions that last for months. If the network has not been trained on all those particular domains, it may simply fail to predict the correct labels and therefore lead to critical failures. Unfortunately, the networks that are robust to a wide variety of domains are often too large, may not fit in memory, may not be fast enough for real-time processing, and may require high power consumption.

One naive scene-specific adaptation consists in training a small real-time network on scene-specific data in a supervised fashion. However, this requires collecting a dataset prior to installation, which may be impractical. Also, if the domain changes dynamically, the network may be unable to adapt. As a solution, Cioppa *et al*. [35] propose the online distillation framework, in which a large network is used to train a small real-time network on the fly. This allows the real-time network to adapt to dynamically changing domains without requiring any manual annotation. We propose an alternative for dynamic domain adaptation by leveraging real-time networks trained on various domains.

3. Method

3.1. Probabilistic framework

Similarities and differences between domains. By assumption, all domains share a measurable space (Ω, Σ), where Ω is a non-empty set (the *sample space*, or *universe*) and $\Sigma \subseteq \Omega^2$ is a σ-algebra on it (the *event space*). They also share a non-empty set of *evidences*, $\mathbb{E} \subseteq \Sigma$, and a non-empty set of *hypotheses*, $\mathbb{H} \subseteq \Sigma$. The differences between the various domains stand in the probability measures. We denote the one associated with the domain d by P_d.

Bayesian inference. This task consists in determining the posterior $P(H \mid E)$ for any given hypothesis $H \in \mathbb{H}$ and any given evidence $E \in \mathbb{E}$. In the following, entities performing this task are called *models*. In some cases, *exact models* $f_H : \mathbb{E} \to \mathbb{R} : E \mapsto P(H \mid E)$ can be established theoretically. In Section 3, we relax this constraint and assume that the models are *exact up to a target shift*, *i.e.* $f_H^* : \mathbb{E} \to \mathbb{R} : E \mapsto P^*(H \mid E)$, the probability measures P and P^* having equal likelihoods:

$$P(E \mid H) = P^*(E \mid H) \qquad \forall E \in \mathbb{E}, \forall H \in \mathbb{H}. \quad (1)$$

In case of non-zero priors, one can recover an exact model by applying a correction, called *target shift*, to the output $P^*(H \mid E)$. When $\mathbb{H}$ forms a partition of Ω, this correction [3, 36] can be written as

$$P(H \mid E) = \frac{\frac{P(H)}{P^*(H)} P^*(H \mid E)}{\sum_{H \in \mathbb{H}} \frac{P(H)}{P^*(H)} P^*(H \mid E)}. \quad (2)$$

Geometric representation. When $\mathbb{H}$ is finite and forms a partition of Ω, priors and posteriors can be represented by points in any non-degenerated Euclidean simplex in $|\mathbb{H}| - 1$ dimensions (*e.g*. a point, a segment, a triangle, a tetrahedron). After establishing a bijection between its vertices and $\mathbb{H}$, the probability relative to the hypothesis H (either the prior or the posterior) can be obtained by projecting the point on the axis that passes through the vertex H and that is orthogonal to its opposite face, the probability being equal to 0 on the face and to 1 at the vertex.

Decision-making. In any domain d, making a decision for a given evidence E comes to choose a hypothesis based on the posteriors $P_d(H \mid E)\ \forall H \in \mathbb{H}$. Together, the Bayesian inference and the decision-making form a function $\mathbb{E} \to \mathbb{H}$. We recall two standard decision-making strategies.

- MAP (*maximum a posteriori*) selects the hypothesis with the highest posterior. MAP has a discontinuity where the hypotheses are equally likely given the evidence. It maximizes the probability of making a correct decision, *i.e*. the accuracy.
- MLE (*maximum likelihood estimation*) selects the hypothesis with the highest likelihood (or, equivalently, the highest posterior to prior ratio). MLE has a discontinuity where posteriors and priors are equal, *i.e*. when the evidence provides no information about the hypotheses. It maximizes the balanced accuracy.

As shown in Figure 2, both strategies partition the Euclidean simplex into $|\mathbb{H}|$ convex parts. These two strategies are related to each others by the fact that the balanced accuracy corresponds to the accuracy after shifting the priors to equally likely hypotheses.

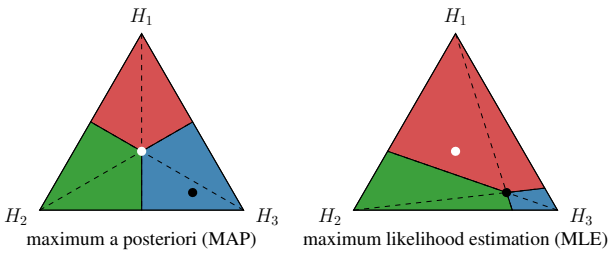

Figure 2. Comparison between two decision-making strategies: maximizing the accuracy (MAP, left) and maximizing the balanced accuracy (MLE, right). Here, the three colors represent the decisions for the three hypotheses $\mathbb{H} = \{H_1, H_2, H_3\}$ forming a partition of Ω. The black dot represents the priors, arbitrarily taken as $P(H_1) = 0.1$, $P(H_2) = 0.2$, and $P(H_3) = 0.7$.

3.2. Problem statement

Let us now focus on real-world problems in which there are two kinds of domains: the *source* and *target* domains, given respectively by the sets $\mathbb{D}_S$ and $\mathbb{D}_T$. In all source domains $d_S \in \mathbb{D}_S$, we assume that it is possible to obtain models computing or estimating the posteriors $P_{d_S}(H \mid E)$, possibly after gathering annotated data following the distribution $P_{d_S}(E, H)$, to the contrary of any "new" target domain $d_T \in \mathbb{D}_T \setminus \mathbb{D}_S$. The computation of $P_{d_T}(H \mid E)$ is an *unsupervised domain adaptation* problem.

We consider now the particular many-to-infinity domain adaptation problem in which the probability measure of any target domain $d_T \in \mathbb{D}_T$ can be obtained as a convex combination of the probability measures of the source domains in the non-empty set $\mathbb{D}_S = \{d_{S1}, d_{S2}, \dots d_{Sn}\}$ as follows

$$P_{d_T} = \sum_{k=1}^{n} \lambda_k P_{d_{Sk}}\,, \tag{3}$$

with $\lambda = (\lambda_1, \lambda_2 \dots \lambda_n) \in \mathbf{\Delta}^{n-1}$.[1] This problem is complex as the distribution of evidences $P(E)$, the priors $P(H)$, and the likelihoods (*a.k.a.* appearance models) $P(E \mid H)$ may vary from one domain to another.

From Equation 3, it results that the distribution of evidences in the target domain is a mixture of the distributions of evidences in the source domains: $P_{d_T}(E) = \sum_{k=1}^{n} \lambda_k P_{d_{Sk}}(E)$, $\forall E \in \mathbb{E}$. Therefore, the domain adaptation problem studied in this paper is a particular case of the more general *mixture adaptation problem* introduced by Mansour *et al*. in [37].

By construction, we consider that the priors can be pre-computed for source domains. Consequently, the priors in the target domain can be computed by $P_{d_T}(H) = \sum_{k=1}^{n} \lambda_k P_{d_{Sk}}(H)$, $\forall H \in \mathbb{H}$. Moreover, the likelihoods in the target domain are mixtures of the corresponding likelihoods from the source domains as $P_{d_T}(E \mid H) = \sum_{k=1}^{n} \omega_{k,H} P_{d_{Sk}}(E \mid H)$ with $\omega_{k,H} = \lambda_k {}^{P_{d_{Sk}}(H)}/_{P_{d_T}(H)}$, $\forall E \in \mathbb{E}$ and $\forall H \in \mathbb{H}$. Therefore, the domain adaptation problem studied here is a sub-case of a more general one in which the target likelihoods are mixtures of fixed components (in our case, the likelihoods in the source domains), the component weights and the target priors being known only at runtime. Some theoretical and experimental results for that on-the-fly domain adaptation problem can be found in [36], in the very specific case of two-class classifiers.

[1] We use the notation $\mathbf{\Delta}^{n-1}$ to denote the $(n-1)$-probabilistic simplex, that is $\lambda \in \mathbf{\Delta}^{n-1}$ if and only if $\sum_{k=1}^{n} \lambda_k = 1$ and $\lambda_k \geq 0$, $\forall k$.

3.3. Problem analysis and theoretical solution

Let us consider any evidence E such that $P_{d_T}(E) \neq 0$. From Equation 3 and using probability theory, we can derive an exact formula to compute $P_{d_T}(H \mid E)$ in any target domain $d_T \in \mathbb{D}_T$. We have, $\forall H \in \mathbb{H}$,

$$P_{d_T}(H \mid E) = \sum_{d_{Sk} \in \Phi(E)} \omega_{k,E} P_{d_{Sk}}(H \mid E) \tag{4}$$

with

$$\Phi(E) = \{d_S \in \mathbb{D}_S : P_{d_S}(E) \neq 0\} \neq \emptyset \tag{5}$$

and

$$\omega_{k,E} = \lambda_k \frac{P_{d_{Sk}}(E)}{P_{d_T}(E)} = \frac{\lambda_k P_{d_{Sk}}(E)}{\sum_{k'=1}^{n} \lambda_{k'} P_{d_{Sk'}}(E)}\,. \tag{6}$$

In the following, we pose $\omega_E = (\omega_{1,E}, \omega_{2,E} \dots \omega_{n,E})$. From Equation 6, we see that $\omega_E \in \mathbf{\Delta}^{n-1}$.

One recognizes in Equation 4 the *distribution weighted combining rule* proposed by Mansour *et al*. in [37]. The theoretical analysis performed in that paper shows that this combination rule behaves well when the posteriors in the source domains are affected by a bounded uncertainty.

3.3.1 Motivation for determining ω_E

In general, $P_{d_T}(H \mid E)$ cannot be determined based only on λ_k and $P_{d_{Sk}}(H \mid E)$, $\forall i \in \{1, 2, \dots n\}$. This is because, unless some λ_k are null, the vector ω_E can sweep the complete probabilistic simplex $\mathbf{\Delta}^{n-1}$. Therefore, the posteriors for the target domain can be anywhere within the convex hull of the posteriors for the source domains d_{Sk} such that $d_{Sk} \in \Phi(E)$ and $\lambda_k > 0$. Figure 3 shows this uncertainty for $|\mathbb{H}| = 3$.

Since MAP is independent of the priors, one can show, by a convexity argument, that, if the decisions are the same in all source domains, then it is also the same in the target domain. Computing ω_E is necessary only when the decisions taken in the source domains are contradictory.

On the contrary, since MLE depends on the priors, the decision made in the target domain can be other than the decisions made in the source domains. In particular, if the decisions are the same in all source domains, then the decision in the target domain could be different.

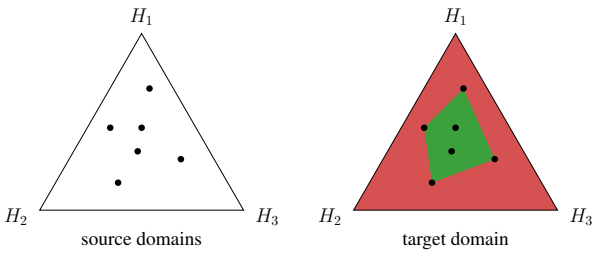

Figure 3. Given the vectors of posteriors in the source domains (black dots), there might be an important uncertainty on the vector of posteriors in the target domain (green area), as explained in Section 3.3.1. A solution is given in Section 3.3.2. In this figure, there are 6 source domains and $\mathbb{H} = \{H_1, H_2, H_3\}$.

3.3.2 Determination of ω_E

In order to establish a way of computing ω_E, we start by giving a probabilistic meaning to it. To this aim, we introduce the overall measurable space (Ω', Σ') with $\Omega' = \Omega \times \mathbb{D}_S$ and $\Sigma' = \Sigma \times 2^{\mathbb{D}_S}$. We define the event $D_{S_k} = \Omega \times \{d_{Sk}\}$ for the k-th source domain and, for any event $X \in \Sigma$, we define the corresponding event $X' = X \times \mathbb{D}_S \in \Sigma'$.

All probabilities considered until now can be written in terms of a unique probability measure P on (Ω', Σ'). It is such that $P(X \times \{d_{Sk}\}) = \lambda_k P_{d_{S_k}}(X)$ for all $X \in \Sigma$ and for all $d_{Sk} \in \mathbb{D}_S$. For example, $P_{d_{S_k}}(X) = P(X' \mid D_{S_k})$ and $P_{d_T}(X) = P(X')$, $\forall X \in \Sigma$. Moreover, λ_k and ω_E acquire a probabilistic meaning as $\lambda_k = P(D_{S_k})$ and $\omega_{k,E} = P(D_{S_k} \mid E')$.

Unfortunately, P cannot be used during the off-line stage because it depends on λ. Instead, we choose to work with the following probability measure P^* on (Ω', Σ'):

$$P^*(X \times \{d_{Sk}\}) = \kappa_k P_{d_{S_k}}(X) \ , \tag{7}$$

$\forall X \in \Sigma$ and $\forall d_{Sk} \in \mathbb{D}_S$, with any arbitrarily chosen vector of strictly positive weights $\kappa = (\kappa_1, \kappa_2 \ldots \kappa_n) \in \mathring{\mathbf{\Delta}}^{n-1}$.[2]

In the same way that we assumed the existence of a model determining $P_{d_S}(H \mid E)$ for any $E \in \mathbb{E}$ and any $H \in \mathbb{H}$, we can also assume that it is possible to obtain a model for $P^*(D_{S_k} \mid E')$, for any given $E \in \mathbb{E}$ and any given source domain. As $P^*(D_{S_k}) = \kappa_k$ and $P^*(E' \mid D_{S_k}) = P_{d_{S_k}}(E)\ \forall E \in \mathbb{E}$, Equation 6 leads to

$$\omega_{k,E} = \frac{\frac{\lambda_k}{\kappa_k} P^*(D_{S_k} \mid E')}{\sum_{k'=1}^{n} \frac{\lambda_{k'}}{\kappa_{k'}} P^*\left(D_{S_{k'}} \mid E'\right)} \ . \tag{8}$$

This equation is similar to Equation 2 and can be interpreted as a prior shift for priors on the n source domains instead of priors on the $|\mathbb{H}|$ hypotheses.

[2]We use the notation $\mathring{\mathbf{\Delta}}^{n-1}$ to denote the interior of $\mathbf{\Delta}^{n-1}$, that is $\kappa \in \mathring{\mathbf{\Delta}}^{n-1}$ if and only if $\sum_{k=1}^{n} \kappa_k = 1$ and $\kappa_k > 0\ \forall k$.

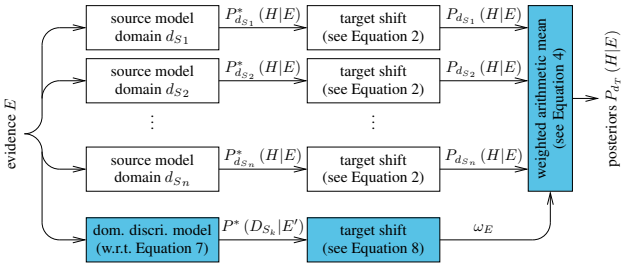

Figure 4. Our unsupervised domain adaptation algorithm. The boxes with a white background are used for determining the posteriors in the source domains. Those with a blue background depict the elements that we have added to obtain, on the fly, the posteriors in the target domain.

3.4. Our domain adaptation algorithm

Figure 4 shows the algorithm that we propose to compute $P_{d_T}(H \mid E)$ on the fly for any hypothesis $H \in \mathbb{H}$, any evidence $E \in \mathbb{E}$, and any target domain $d_T \in \mathbb{D}_T$.

The off-line step. Two types of models are obtained off-line, before knowing the target domain: $|\mathbb{D}_S|$ *source models* and 1 *domain discriminator model*. They are used to determine $P^*_{d_{S_k}}(H \mid E)$ and $P^*(D_{S_k} \mid E')$, respectively. Training the source models requires single-source labeled data. Training the domain discriminator model requires multiple-source unlabeled data. Equations 1 and 7 specify the distributions that have to be assumed by the learner.

The on-the-fly step. Equation 2 is applied to obtain $P_{d_{S_k}}(H \mid E)$ from $P^*_{d_{S_k}}(H \mid E)$. Moreover, knowing the target domain (*i.e.* the vector λ), Equation 8 is applied to obtain $\omega_{k,E}$ from $P^*(D_{S_k} \mid E')$. Finally, Equation 4 leads to the posterior $P_{d_T}(H \mid E)$ in the target domain. The overall algorithm gives the flexibility to adapt the posteriors on the fly to an evolving target domain.

Discussion. If all the source models and the domain discriminator model are *exact up to a target shift*, then our algorithm behaves as an *exact model* for the target domain. In practice, however, models are rarely exact (*i.e.* they have noisy outputs). Taking the L_1 distance to measure the errors, it can be shown that: if the error on ω_E (over $\mathbb{D}_S$) is bounded by ϵ_{ω_E} and if the error on $P_{d_S}(H \mid E)$ (over $\mathbb{H}$, a partition of Ω) is bounded by ϵ_{d_S} for all $d_S \in \mathbb{D}_S$, then the error on $P_{d_T}(H \mid E)$ is bounded by $\epsilon_{d_T} \leq \epsilon_{d_S} + \epsilon_{\omega_E}$. In other words, the last step of the algorithm behaves well when facing noisy inputs. However, theoretically, the upstream target shift operations can amplify the noise affecting the outputs of the various models, especially when priors are low. Therefore, in the next section, we will demonstrate the practical suitability of this algorithm, in the challenging case in which some priors are low. Also, we validate the predicted posteriors after the target shifts.

4. Experiments

4.1. Experimental Setup

We evaluate and compare our domain adaptation algorithm on the semantic segmentation task, which consists in predicting, for each pixel of an image, the semantic class of its enclosing object or region in the scene.

In accordance with the previous section, the notations are as follows. We denote the set of images by $\mathbb{I}$, the set of pixels by $\mathbb{P}$, and the set of semantic classes by $\mathbb{S}$. All these sets being finite in our experimental setup, we take $\Omega = \mathbb{I} \times \mathbb{P} \times \mathbb{S}$ and $\Sigma = 2^{\Omega}$. The evidence $E_{i,p}$ for the pixel p in image i is $\{(i, p, s) \mid s \in \mathbb{S}\}$ and the hypothesis H_s for the groundtruth semantic class s is $\{(i, p, s) \mid i \in \mathbb{I},\ p \in \mathbb{P}\}$. Note that $\mathbb{H}$ is a partition of Ω. With our notations, the semantic segmentation task aims at choosing an element of $\mathbb{S}$, the predicted semantic class, based on the posteriors $P_d\left(H_s \mid E_{i,p}\right)$.

4.1.1 Off-line choices

Our algorithm requires two types of off-line models: (1) source models and (2) a domain discriminator model. In the following, we describe each choice for these models.

Choice of models. Various machine learning techniques can be used as posterior estimators. In general, deep learning is suitable for predicting posteriors when the loss minimized during the training stage is carefully chosen. In particular, the cross-entropy reaches a local minimum when the output of the network corresponds to the posteriors $P_d\left(H \mid E\right)\ \forall H \in \mathbb{H}$ [38, 39]. For the architecture of the source models and the domain discriminator model, we chose the TinyNet [35, 40] segmentation network, which is a lightweight architecture that only needs a few training samples and is fast to train.

Training the source models. For all our experiments, we train each source model on its own source dataset separately. The models are trained with batches of 12 images randomly sampled from the training set of the source domain. We use a learning rate of 10^{-4} with a reduce-on-plateau-scheduling strategy, a patience of 10 and a reduction factor of 0.1. We use the Adam optimizer with default parameters and no weight decay [41]. To avoid typical problems when dealing with unbalanced distributions of classes, we use the weighted cross-entropy loss, for which the weighting factor is estimated using the priors of the source domain. These choices are motivated by the wish to obtain source models *exact up to a target shift*. We aim at satisfying Equation 1 with $P^*_{d_S}\left(H_s\right) = \left|\mathbb{S}\right|^{-1}\ \forall H_s \in \mathbb{H}$. Indeed, the chosen loss achieves a local minimum for a pixel p in an image i when the source model $f^*_{sm} : \mathbb{E} \rightarrow \mathbf{\Delta}^{|\mathbb{S}|-1}$ gives $f^*_{sm}(i,p)_s = P^*_{d_S}\left(H_s \mid E_{i,p}\right)$ for all semantic classes (indexed by s here).

Training the domain discriminator model. To estimate $\omega_{k,E}$, we train a soft discriminator to recognize the corresponding domain of each pixel. We use the same TinyNet architecture, training procedure, and hyperparameters than for training the source models. Nevertheless, we replace the number of semantic classes with the number of source domains in the output layer and the batch size by 4.

Regarding training samples, we propose a way to generate multi-domain images, inspired by the mosaic transformation presented in [42]. We combine four patches cropped from four randomly drawn images to create new training images. These choices are motivated by the wish to satisfy Equation 7 with $\kappa_k = \left|\mathbb{D}_S\right|^{-1}$. Indeed, the cross-entropy loss achieves a local minimum for a pixel p in an image i when the domain discriminator model $f^*_{ddm} : \mathbb{E} \rightarrow \mathbf{\Delta}^{n-1}$ gives $f^*_{ddm}(i,p)_k = P^*\left(D_{S_k} \mid E'_{i,p}\right)$ for all source domains (indexed by k here).

Validation of predicted posteriors. The source models and the domain discriminator model are expected to be exact up to a target shift. Throughout our experiments, we systematically perform two tests to establish their trustablility. First, we compute the posteriors estimated by these models and verify (after the necessary target shifts) that these predictions correspond, in expectation, to the respective priors. Second, we verify by visual inspection that these models are well calibrated using calibration plots following [43, 44].

4.1.2 Evaluation and comparison with heuristics

For the sake of evaluation, we simulate various target domains by mixing the test sets of the different source domains. This is achieved by weighting the images. In our experiments, all images have the same size and all test sets contain the same amount of images. Thus, we weight the images of the k-th test set by λ_k to satisfy Equation 3.

We use 4 common performance scores to evaluate the quality of the semantic segmentations: the accuracy, balanced accuracy, mean IoU (macro-averaging), and balanced mean IoU. All reported results are for the decision-making strategy MAP. We compare the scores obtained with our algorithm to those obtained with the 3 following heuristics.

1. **Source models.** The first heuristic consists in using each source model separately on the target domains. For a fair comparison, we apply a target shift on the posteriors corresponding to the target domain priors following Equation 2.

2. **Random selection of source models.** The second heuristic randomly selects the decision derived from the k-th source model with a probability given by λ_k.

3. **Linear combination of posteriors.** The third heuristic combines linearly the posteriors provided by the source models as follows: $\sum_{k=1}^{n} \lambda_k\, P_{d_{S\,k}}\left(H_s \mid E_{i,p}\right)$.

Figure 5. Examples of images (left) from the datasets Cityscapes (above) and BDD100K (below), with the corresponding groundtruth images (right).

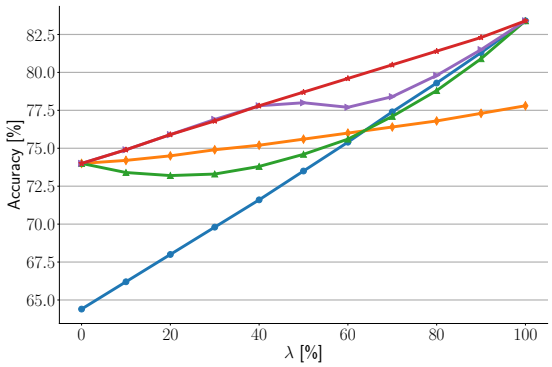

Figure 6. Results of our first experiment (all decisions made with the MAP strategy) showing the accuracy with the **source model trained on CityScapes (in blue)**, the **source model trained on BDD100K (in orange)**, the **random selection of source models (in green)**, the **linear combination of posteriors (in purple)**, and **our algorithm (in red)**. $\lambda = 0$ (resp. $= 100$) corresponds to BDD100K (resp. CityScapes) as target domain.

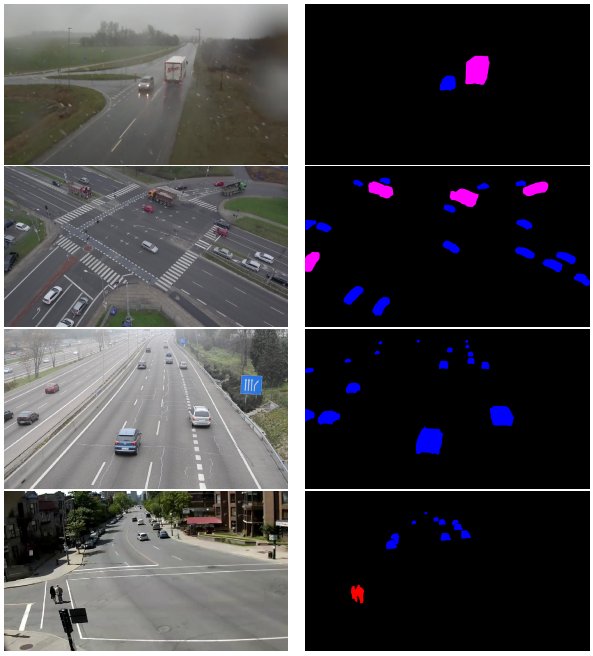

Figure 7. Examples of images (left) from the surveillance datasets (from top to bottom) RainSnow, MTID/Drone, GRAM-RTM/M-30-HD and UT/Sherbrooke, with the corresponding pseudo-groundtruth images (right).

4.2. Experiment with 2 source domains

In the first experiment, we consider the semantic segmentation of high-definition (1280×720) color images acquired by cameras installed in moving vehicles, behind the windshield. In this experiment, there are 2 source domains and 19 strongly imbalanced semantic classes.

Motivation. The semantic segmentation of such images is reputed to be a preliminary step towards autonomous vehicles. There is a native need for considering multiple domains as we expect differences in the appearance of roads, traffic signs, and cars from one country to another. Moreover, the sensors and their positioning can differ from one car to another. And, last but not least, the weather and traffic conditions can vary continuously. The authors of [45] noted a dramatic domain shift between two datasets (BDD100K [45] and Cityscapes [25]). According to their results, the semantic segmentation models perform much worse when tested on a different dataset.

Data. The source domains are represented by the datasets CityScapes (data acquired in European cities, mostly in Germany) and BDD100K (data acquired in the USA). These datasets gather color images with their respective semantic segmentation groundtruths (see Figure 5). We resized and cropped all images to 1280×720. We also randomly split each dataset into a training set (2726 images per dataset), a validation set (250 images per dataset), and a test set (500 images per dataset).

Results. We compare the accuracy of our algorithm with the 3 heuristics in Figure 6.

The first heuristic (source models) leads to the orange (model trained on BDD100K) and blue (model trained on CityScapes) curves. When the target domain coincides with one of the source domains (both sides of the graph), the corresponding source model behaves much better than the other one. This confirms the observation made in [45] and the need for domain adaptation, as simply choosing a source model and applying it directly to a different target domain leads to a drastic decrease in performance.

The second heuristic (random selection of source models, in green) is clearly suboptimal compared to the best source model for the current target domain. It is therefore not an effective way to combine the source models.

The third heuristic (linear combination of the posteriors, in purple) reaches better performances than the two previ-

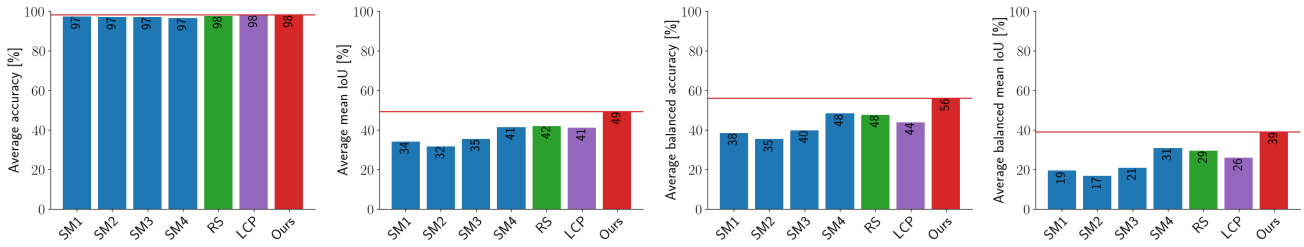

Figure 8. Results of our second experiment (all decisions made with the MAP strategy) showing the mean behaviour over 15 diversified target domains. We compare the different **source models (SM1-4, in blue)** with the **random selection of source models (RS, in green)**, **linear combination of posteriors (LCP, in purple)**, and **our algorithm (in red)**. Four performance scores are averaged over the 15 target domains: the accuracy, mean IoU, balanced accuracy, and balanced mean IoU (from left to right).

ous heuristics. This is a surprise as this simple heuristic is known to be inappropriate, by theoretical arguments [37].

Our domain adaptation algorithm (red curve) outperforms all heuristics, especially when the weight of CityScapes is above 40% in the target domain mixture.

4.3. Experiment with 4 source domains

Let us consider a second semantic segmentation task, with images acquired by fixed video surveillance traffic cameras. In this experiment, there are 4 source domains and 4 strongly imbalanced semantic classes.

Data. The four source domains are represented by the following datasets. (1) AAU RainSnow Traffic Surveillance Dataset [46] (called *RainSnow*) contains traffic surveillance videos in rainfall and snowfall from seven different intersections. (2) Multi-view Traffic Intersection Dataset (MTID) [47] (called *MTID/Drone*) contains footage of the same intersection from two different points of view. We only use the images recorded with a drone from one point of view. (3) GRAM Road-Traffic Monitoring (GRAM-RTM) dataset [48] (called *GRAM-RTM/M-30-HD*) consists of three videos recorded under different conditions and with different platforms. We only use one video, named M-30-HD. (4) The dataset from the Urban Tracker [49] project (called *UT/Sherbrooke*) contains also several videos. We only select the Sherbrooke video, filmed at the Sherbrooke/Amherst intersection in Montreal.

For each dataset, we randomly select 1299 images for the training set, 300 images for the validation set and 300 images for the test set. All images have been resized (*e.g.* by upsampling, downsampling, or cropping) to 1280×720.

Since no segmentation groundtruth is available for all of these datasets and since we need all datasets to share the same semantic classes, we use the PointRend [28] algorithm trained on the COCO [24] dataset to obtain pseudo-groundtruths. For this experiment, we consider 4 semantic classes: background, person, two wheels (bicycle and motorcycle), and four wheels (car, bus, and truck). An example of an image and its pseudo-groundtruth for each dataset is illustrated in Figure 7.

Performance analysis. Since we have 4 source domains in this experiment, it is impossible to depict the performances as we did in the first experiment. Instead, Figure 8 shows the mean behaviour over several target domains. Bar plots are provided for 4 performance scores. They are obtained by averaging the scores of our algorithm and of the heuristics over 15 target domains obtained by mixing either 1, 2, 3, or 4 source domains with equal weights. As can be seen, on average, our algorithm outperforms all heuristics, with a large margin for three of the four scores. This demonstrates the superiority of our algorithm.

5. Conclusion

In this paper, we focus on the domain adaptation problem in which the probability measure of the target domain is a convex combination of the probability measures of the source domains. We define a probabilistic framework and show that the posteriors in the target domain do not depend solely on posteriors, priors, and relative weights of the source domains. From there, we provide a theoretical proof of an algorithm to compute the posteriors for the target domain in an unsupervised way. This is particularly valuable when the target domain can change on the fly. Interestingly, this flexibility is achieved by keeping the annotated data collected in different source domains separate and by sharing only unlabeled data. Finally, we test our unsupervised domain adaptation algorithm on a semantic segmentation task in real-world surveillance, and show its superiority compared to common heuristics.

Acknowledgements

This work has been made possible thanks to the *TRAIL* initiative (`https://trail.ac`). Part of it was supported by the Walloon region (Service Public de Wallonie Recherche, Belgium) under grant n°2010235 – *ARIAC by DIGITALWALLONIA4.AI*. Anthony Cioppa is funded by the FNRS (`https://www.frs-fnrs.be/en/`).

References

[1] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," *Trans. Comput. Sci. Comput. Intell.*, pp. 877–894, 2021.

[2] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, and J. Woo, "Deep unsupervised domain adaptation: A review of recent advances and perspectives," *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, 2022.

[3] T. Sipka, M. Sulc, and J. Matas, "The hitchhiker's guide to prior-shift adaptation," in *IEEE Winter Conf. Applicat. Comp. Vis. (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 2031–2039.

[4] M. Kirchmeyer, A. Rakotomamonjy, E. de Bezenac, and P. Gallinari, "Mapping conditional distributions for domain adaptation under generalized target shift," in *Int. Conf. on Learn. Rep. (ICLR)*, 2022.

[5] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani, *Advances in domain adaptation theory*. Elsevier, 2019.

[6] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. Mach. Learn. (ICML)*, vol. 37, Lille, France, July 2015, pp. 1180–1189.

[7] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Salt Lake City, UT, USA: IEEE, June 2018, pp. 3723–3732.

[8] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation," in *Int. Conf. Mach. Learn. (ICML)*, vol. 119, July 2020, pp. 6028–6039.

[9] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv*, vol. abs/2006.10726, 2020.

[10] M. Boudiaf, R. Mueller, I. B. Ayed, and L. Bertinetto, "Parameter-free online test-time adaptation," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. IEEE, June 2022, pp. 8334–8343.

[11] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *IEEE Int. Conf. Comput. Vis. (ICCV)*. IEEE, Oct. 2017, pp. 2039–2049.

[12] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. IEEE, June 2019, pp. 6929–6938.

[13] S. Zhao, B. Li, X. Yue, Y. Gu, P. Xu, R. Hu, H. Chai, and K. Keutzer, "Multi-source domain adaptation for semantic segmentation," in *Adv. in Neural Inform. Process. Syst. (NeurIPS)*, vol. 32. Vancouver, Canada: Curran Associates, Inc., 2019, pp. 1–14.

[14] S. Zhao, B. Li, P. Xu, X. Yue, G. Ding, and K. Keutzer, "MADAN: Multi-source adversarial domain aggregation network for domain adaptation," *Int. J. Comp. Vis.*, vol. 129, no. 8, pp. 2399–2424, May 2021.

[15] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *IEEE Int. Conf. Comput. Vis. (ICCV)*. IEEE, Oct. 2019, pp. 1406–1415.

[16] J. He, X. Jia, S. Chen, and J. Liu, "Multi-source domain adaptation with collaborative learning for semantic segmentation," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. IEEE, June 2021, pp. 11 003–11 012.

[17] S.-L. Sun and H.-L. Shi, "Bayesian multi-source domain adaptation," in *Int. Conf. Mach. Learn. Cybern. (ICMLC)*. IEEE, July 2013, pp. 24–28.

[18] A. Garcia-Garcia, S. Orts, S. Oprea, V. Villena-Martinez, and J. García Rodríguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv*, vol. abs/1704.06857, 2017.

[19] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1089–1106, June 2018.

[20] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 3523–3542, 2021.

[21] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Honolulu, HI, USA: IEEE, July 2017, pp. 5122–5130.

[22] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comp. Vis.*, vol. 127, no. 3, pp. 302–321, Dec. 2018.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Eur. Conf. Comput. Vis. (ECCV)*, ser. Lect. Notes Comput. Sci., vol. 8693. Springer Int. Publ., 2014, pp. 740–755.

[25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 3213–3223.

[26] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Honolulu, HI, USA: IEEE, July 2017, pp. 6230–6239.

[27] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comput. Vis. (ICCV)*. Venice, Italy: IEEE, Oct. 2017, pp. 2980–2988.

[28] A. Kirillov, Y. Wu, K. He, and R. Girshick, "PointRend: Image segmentation as rendering," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 9796–9805.

[29] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 6877–6886.

[30] W. Zhou, J. Liu, J. Lei, L. Yu, and J.-N. Hwang, "GMNet: Graded-feature multilabel-learning network for RGB-thermal urban scene semantic segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 7790–7802, 2021.

[31] T. Cane and J. Ferryman, "Evaluating deep semantic segmentation networks for object detection in maritime surveillance," in *IEEE Int. Conf. Adv. Video and Signal Based Surveillance (AVSS)*. Auckland, New Zealand: IEEE, Nov. 2018, pp. 1–6.

[32] N. A. Muhadi, A. F. Abdullah, S. K. Bejo, M. R. Mahadi, and A. Mijic, "Deep learning semantic segmentation for water level estimation using surveillance camera," *Appl. Sci.*, vol. 11, no. 20, p. 9691, Oct. 2021.

[33] S. Zheng and G. Gupta, "Semantic-guided zero-shot learning for low-light image/video enhancement," in *IEEE Winter Conf. Applicat. Comp. Vis. (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 581–590.

[34] A. Aakerberg, A. S. Johansen, K. Nasrollahi, and T. B. Moeslund, "Semantic segmentation guided real-world super-resolution," in *IEEE Winter Conf. Applicat. Comp. Vis. (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 449–458.

[35] A. Cioppa, A. Deliege, M. Istasse, C. De Vleeschouwer, and M. Van Droogenbroeck, "ARTHuS: Adaptive real-time human segmentation in sports through online distillation," in *IEEE Int. Conf. Comput. Vis. and Pattern Recogn. Work. (CVPRW), CVsports*. Long Beach, CA, USA: IEEE, June 2019, pp. 2505–2514.

[36] S. Piérard, A. Marcos Alvarez, A. Lejeune, and M. Van Droogenbroeck, "On-the-fly domain adaptation of binary classifiers," in *Belgian-Dutch Conference on Machine Learning (BENELEARN)*, Brussels, Belgium, June 2014, pp. 20–28.

[37] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Adv. in Neural Inform. Process. Syst. (NeurIPS)*, vol. 21. Vancouver, Canada: Curran Associates, Inc., Dec. 2008, pp. 1041–1048.

[38] J. W. Miller, R. Goodman, and P. Smyth, "On loss functions which minimize to conditional expected values and posterior probabilities," *IEEE Trans. Inform. Theory*, vol. 39, no. 4, pp. 1404–1408, July 1993.

[39] D. Ramos, J. Franco-Pedroso, A. Lozano-Diez, and J. Gonzalez-Rodriguez, "Deconstructing cross-entropy for probabilistic binary classifiers," *Entropy*, vol. 20, no. 3, p. 208, Mar. 2018.

[40] A. Cioppa, A. Deliège, and M. Van Droogenbroeck, "A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games," in *IEEE Int. Conf. Comput. Vis. and Pattern Recogn. Work. (CVPRW), CVsports*, Salt Lake City, UT, USA, June 2018, pp. 1846–1855.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. on Learn. Rep. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[42] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv*, vol. abs/2004.10934, 2020.

[43] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Int. Conf. Mach. Learn. (ICML)*. ACM Press, 2005, pp. 1–8.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. of Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[45] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *IEEE Int. Conf. Comput. Vis. and Pattern Recogn. (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 2633–2642.

[46] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2802–2819, Aug. 2019.

[47] M. B. Jensen, A. Mogelmose, and T. B. Moeslund, "Presenting the multi-view traffic intersection dataset (MTID): A detailed traffic-surveillance dataset," in *IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*. Rhodes, Greece: IEEE, Sept. 2020, pp. 1–6.

[48] R. Guerrero-Gómez-Olmedo, R. J. López-Sastre, S. Maldonado-Bascón, and A. Fernández-Caballero, "Vehicle tracking by simultaneous detection and viewpoint estimation," in *Int. Work. Interplay Between Nat. Artif. Comput.*, ser. Lect. Notes Comput. Sci., vol. 7931. Springer Sci. Bus. Media LLC, 2013, pp. 306–316.

[49] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier, "Urban tracker: Multiple object tracking in urban mixed traffic," in *IEEE Winter Conf. Applicat. Comp. Vis. (WACV)*. Steamboat Springs, CO, USA: IEEE, Mar. 2014, pp. 885–892.