

Article

Unsupervised Domain Adaptation for Forest Fire Recognition Using Transferable Knowledge from Public Datasets

Zhengjun Yan ^{1,2}, Liming Wang ^{1,*}, Kui Qin ¹, Feng Zhou ^{1,2}, Jineng Ouyang ¹, Teng Wang ¹, Xinguo Hou ¹ and Leping Bu ¹

¹ School of Electrical Engineering, Naval University of Engineering, Wuhan 430033, China

² Ordnance NCO Academy, Army Engineering University of PLA, Wuhan 430075, China

* Correspondence: icesoar@163.com

Abstract: Deep neural networks (DNNs) have driven the recent advances in fire detection. However, existing methods require large-scale labeled samples to train data-hungry networks, which are difficult to collect and even more laborious to label. This paper applies unsupervised domain adaptation (UDA) to transfer knowledge from a labeled public fire dataset to another unlabeled one in practical application scenarios for the first time. Then, a transfer learning benchmark dataset called Fire-DA is built from public datasets for fire recognition. Next, the Deep Subdomain Adaptation Network (DSAN) and the Dynamic Adversarial Adaptation Network (DAAN) are experimented on Fire-DA to provide a benchmark result for future transfer learning research in fire recognition. Finally, two transfer tasks are built from Fire-DA to two public forest fire datasets, the aerial forest fire dataset FLAME and the large-scale fire dataset FD-dataset containing forest fire scenarios. Compared with traditional handcrafted feature-based methods and supervised CNNs, DSAN reaches 82.5% performance of the optimal supervised CNN on the testing set of FLAME. In addition, DSAN achieves 95.8% and 83.5% recognition accuracy on the testing set and challenging testing set of FD-dataset, which outperform the optimal supervised CNN by 0.5% and 2.6%, respectively. The experimental results demonstrate that DSAN achieves an impressive performance on FLAME and a new state of the art on FD-dataset without accessing their labels during training, a fundamental step toward unsupervised forest fire recognition for industrial applications.

Keywords: forest fire recognition; unsupervised domain adaptation; transfer learning; deep neural network



Citation: Yan, Z.; Wang, L.; Qin, K.; Zhou, F.; Ouyang, J.; Wang, T.; Hou, X.; Bu, L. Unsupervised Domain Adaptation for Forest Fire Recognition Using Transferable Knowledge from Public Datasets. *Forests* **2023**, *14*, 52. <https://doi.org/10.3390/f14010052>

Academic Editors: Rafael Coll Delgado and Rafael De Ávila Rodrigues

Received: 7 November 2022
Revised: 20 December 2022
Accepted: 20 December 2022
Published: 27 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fire is one of the major disasters facing humankind today, since it causes a large number of casualties and property damage. According to a report [1] by the Fire and Rescue Department Ministry of Emergency Management, People's Republic of China, there were 748,000 fires nationwide in 2021, resulting in 1987 deaths, 2225 injuries, and CNY 6.75 billion in property damage. In the United States, local fire departments responded to 1,353,500 fires in 2021. These fires caused 3800 civilian deaths, 14,700 civilian injuries, and USD 15.9 billion in property damage [2]. Therefore, it is essential to detect and warn of fires on time.

After temperature and particle sensors, vision sensors are now widely used in fire detection because of their long detection distance, fast response, and ability to monitor fire size, location, and expansion rate. Vision sensor-based algorithms have gone through a journey from handcrafted feature-based to DNN-based approaches.

The handcrafted feature-based approach usually contains two parts, a feature extractor and a classifier. Feature extractors were generally designed based on static features, such as color [3], shape [4], edge [5], texture [6], etc., and dynamic features, such as motion [7], shape variation [8], flicker [9], dynamic texture [10], etc. The classifiers were usually based

on rule [11], expert system [8], machine learning (such as SVM [12], Bayes networks [13], linear model [14], neural network [15]), etc. The advantages of the handcrafted feature-based approaches are simplicity and ease of deployment, while the disadvantages are high false-alarm rates and lack of robustness to different scenarios.

In recent years, DNNs have been widely used in fire detection based on the following attributes: 1. the powerful feature extraction and representation learning capabilities; 2. the “end-to-end” automatic learning ability; 3. improved accuracy, reduced false-alarm rate, and enhanced robustness over the handcrafted feature-based approaches. However, the DNNs also introduce the following new challenges: 1. the networks are time-consuming and computationally expensive when training from scratch and tend to overfit on small-scale datasets; 2. the “end-to-end” automatic learning process requires large and diverse datasets, which are time-consuming and laborious to collect and label. As for the first challenge, this paper applies the “pre-train and fine-tuning” technique to accelerate the convergence and improve the generalization of the networks [16–19], where the networks are pre-trained on ImageNet and fine-tuned on fire datasets. In response to the second challenge, this paper builds a transfer learning benchmark dataset called Fire-DA from public fire datasets with labels and applies UDA to transfer the domain-invariant knowledge from Fire-DA to other unlabeled fire datasets. Extensive experiments based on the FLAME [20] and FD-dataset [21] are conducted to compare the fire recognition capabilities of UDA with existing handcrafted feature-based and supervised CNN-based methods.

The main contributions of this paper are as follows:

- UDA is applied to fire recognition for the first time, which transfers the knowledge learned from labeled public fire datasets to unlabeled ones in specific application scenarios.
- The Fire-DA dataset for transfer learning is constructed for validation in the hope that it can serve as a benchmark for future work.
- Extensive transfer experiments are conducted on Fire-DA to study the transfer properties between different fire datasets and compare the performances of DSAN [22] and DAAN [23].
- The performance of the traditional handcrafted feature-based methods, supervised CNNs, and DSAN is compared based on the FLAME and FD-dataset. The feasibility of our UDA-based approach for fire recognition is verified.

The remainder of the paper is organized as follows: Section 2 presents related work, Section 3 describes the Fire-DA dataset and the framework of our UDA-based method for fire recognition, Section 4 contains experimental results and analysis for our method in fire recognition tasks, and Section 5 consists of conclusions and future perspectives.

2. Related Work

In this section, we focus on the current status of DNN applications in fire detection and then introduce the UDA and its applications.

2.1. DNNs in Fire Detection

Deep learning has made a big splash in the computer vision (CV) field. As a typical downstream task of CV, fire detection has yielded many methods based on deep learning [24].

First of all, various DNNs were applied to fire detection, such as AlexNet [16,21,25,26], VGG [27], GoogleNet [18,26], ResNet [17,27], EfficientNet [28,29], SqueezeNet [19], MobileNet [27,30], and other classification networks for fire recognition, Faster-RCNN [27], EfficientDet [29], SSD [31], YOLO [29,32], and other object detection networks for fire detection, DeepLab [33], FusionNet [34], UNet [35], and other segmentation networks for flame segmentation, LSTM [25], GRU [36], and other RNNs for video fire recognition, and Generative Adversarial Network (GAN) [35,37] for fire data augmentation. For forest fire detection, [29] proposed an ensemble learning method for different forest scenarios based on YOLOv5, EfficientDet, and EfficientNet. Ref. [20] collected a fire aerial image dataset called FLAME for forest fire detection and segmentation. Ref. [38] proposed a transfer

learning-based method for fire recognition on FLAME and explored the effects of different backbones, network depths, activation functions, fine-tuning layer settings, and sample augmentation (SA) methods on forest fire recognition.

Secondly, some research focused on the structural modification of DNNs for fire detection. The attention mechanism was incorporated [21,28,35,39] and the multiscale features were fused [19,21,40] with CNNs. Specifically, [28] implemented coarse object localization under weakly supervised conditions using the gradient-weighted class activation mapping (Grad-CAM) techniques. Ref. [34] proposed a semantic segmentation network for fires by combining FusionNet with a residual structure-based middle-skip connection module. Ref. [40] designed the feature-squeeze block to fuse the feature maps of different depths to improve the recognition accuracy for flames in various sizes. In addition, some research compressed CNNs to trade-off between fire detection accuracy and model complexity for lightweight deployment in real-world surveillance networks, e.g., [18] balanced the computational efficiency and accuracy by transfer learning, [19,21] reduced the parameters and saved computational cost by replacing the convolutional kernels with small ones, [19,30] further decreased the computational complexity and model size by eliminating fully connected layers, and [26] obtained a lightweight structure by reducing the depth and width of CNN.

Thirdly, some researchers tried to combine traditional methods with DNNs for fire detection. Ref. [25] converted RGB images into optical flow images to exclude background interference and then extracted spatial and temporal features for fire detection based on CNN and LSTM, respectively. Ref. [26] achieved fire localization by combining lightweight CNN and traditional super-pixel localization techniques. Ref. [27] combined the 2D Haar transform and CNN to propose a new fire detection algorithm called Haar-CNN to compensate for the limitations of convolutional and pooling layers for spectral analysis. Ref. [32] proposed a video fire detection approach by combining YOLO and a random forest classifier for night-time. Ref. [41] extracted the suspected region based on the traditional RGB model and then detected fires by CNN. Ref. [42] began with extracting the motion region in the video frame via conventional background subtraction, then obtained the suspected region by the pixel frequency matrix, and finally determined whether there was a fire by CNN. Ref. [43] solved the information loss due to the large-view angle of UAV aerial images by combining traditional saliency detection and CNN.

In summary, although plenty of DNNs were applied in fire detection tasks, they invariably required the support of a large amount of data with labels. Transfer learning-based methods have also been proposed for fire detection to solve the problem of insufficient data in the target domain [16–19,28,38,39]. However, they are still supervised because the target domain labels are accessed during fine-tuning. Therefore, this paper introduces UDA to free the hands of people working on labeling.

2.2. UDA and Its Application

UDAs project the labeled source domain and unlabeled target domain into a common feature space and then align their feature distributions by different means to learn transferable knowledge. According to the alignment method, there are two main types of existing UDAs, discrepancy-based and adversarial-based UDAs.

The discrepancy-based UDAs measure the feature distribution gap between source and target domains based on predefined distances and achieve alignment by explicitly reducing the gap. DDC [44] proposed a parallel CNN structure with a single adaptation layer to learn category-discriminative and domain-invariant features by “end-to-end” training. It designed a joint loss combining supervised classification loss in the source domain and unsupervised cross-domain distribution distance based on maximum mean discrepancy (MMD). DAN [45] applied multiple kernel variants of MMD (MK-MMD) to multiple adaptation layers based on DDC. It integrated task-specific representations of multiple layers and solved the problem that MMD is sensitive to the choice of kernel functions. DeepCoral [46] aligned the source and target domains’ second-order statistics

(covariance) by polynomial kernels. DSAN [22] addressed the problem that traditional discrepancy-based UDAs can only align the marginal distribution. It obtained a more fine-grained alignment by minimizing the local maximum mean discrepancy (LMMD) of the subdomains formed by the samples from the same category in both domains.

The adversarial-based UDAs achieve feature alignment via the minimax two-player game strategy, where one player is the domain discriminator trained to distinguish the source domain from the target domain and the other is the feature extractor trained to confuse the domain discriminator by learning domain-invariant features. As an early adversarial-based UDA, DANN [47] contained the following four parts, a feature extractor, a label classifier, a domain discriminator, and a gradient reverse layer (GRL). The GRL was designed between the feature extractor and domain classifier to let the domain classifier weights update in the opposite direction to the feature extractor weights when optimizing the domain classifier loss so that the two-step alternate training process could be implemented in one step. Since the domain discriminator accepted the overall samples from both domains, DANN is a marginal distribution adversarial-based UDA. MADA [48] extended DANN's single-domain discriminator to multiple-domain discriminators and aligned complex multi-peak distributions by considering both marginal and conditional distributions. DAAN [23] introduced dynamic factors based on MADA to play the role of different distribution alignments in different situations by dynamically aligning marginal and conditional distributions.

Recently, UDAs have been widely used in the CV field, such as image classification [49], object detection [50], image/video segmentation [51], and image/video retrieval [52] tasks. To our best knowledge, there is no work that is the same as ours for vision-based fire detection using UDA.

3. Materials and Methods

3.1. Problem Definition

As shown in Figure 1a, we are aware of a labeled source domain and an unlabeled target domain for the UDA-based fire recognition task. The two domains follow different distributions. UDA aims to train a network capable of extracting domain-invariant features so that the classifier trained on the source domain can distinguish samples from the target domain. We define the source domain \mathcal{D}_s and the target domain \mathcal{D}_t to be obtained by sampling through different distributions p and q ($p \neq q$), denoted as $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and $\mathcal{D}_t = \{x_i^t\}_{i=1}^{n_t}$, respectively, where n_s and n_t represent the number of samples in the source and target domain, respectively. The label y^s in the source domain and y^t in the target domain belong to the same label space \mathbb{R}^C , where C represents the number of categories. The goal is to learn the optimal model $\mathcal{F}: x^t \mapsto y^t$ by labeled samples in the source domain and unlabeled samples in the target domain such that the prediction error $\epsilon = \mathbb{E}_{(x,y) \in \mathcal{D}_t} [\mathcal{F}(x) \neq y]$ on the target domain is minimized. The loss function of discrepancy-based and adversarial-based UDA can be uniformly defined as:

$$\mathcal{L} = \mathcal{L}_{task}(\mathcal{D}_s) + \lambda \mathcal{L}_{adapt}(\mathcal{D}_s, \mathcal{D}_t) \quad (1)$$

where $\mathcal{L}_{task}(\cdot)$ represents task-related loss, $\mathcal{L}_{adapt}(\cdot, \cdot)$ denotes domain adaptation loss, and λ is a trade-off factor.

3.2. DSAN

Among discrepancy-based UDAs, MMD is a classical measurement of distribution distance, and its empirical approximation is usually expressed as follows:

$$\text{MMD}(\mathcal{D}_s, \mathcal{D}_t) = \left\| \frac{1}{n_s} \sum_{x_i \in \mathcal{D}_s} \phi(x_i) - \frac{1}{n_t} \sum_{x_j \in \mathcal{D}_t} \phi(x_j) \right\|_{\mathcal{H}}^2 \quad (2)$$

where \mathcal{H} represents the reproducing kernel Hilbert space (RKHS), $\phi(\cdot)$ indicates the mapping from original space to RKHS, which can be understood as layer activation in CNNs, and $\|\cdot\|_{\mathcal{H}}^2$ denotes the RKHS norm. Furthermore, the loss function of MMD-based UDA can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{cls}(\mathcal{D}_s) + \lambda \text{MMD}(\mathcal{D}_s, \mathcal{D}_t) \quad (3)$$

where $\mathcal{L}_{cls}(\mathcal{D}_s)$ is the cross-entropy loss in the fire recognition task.

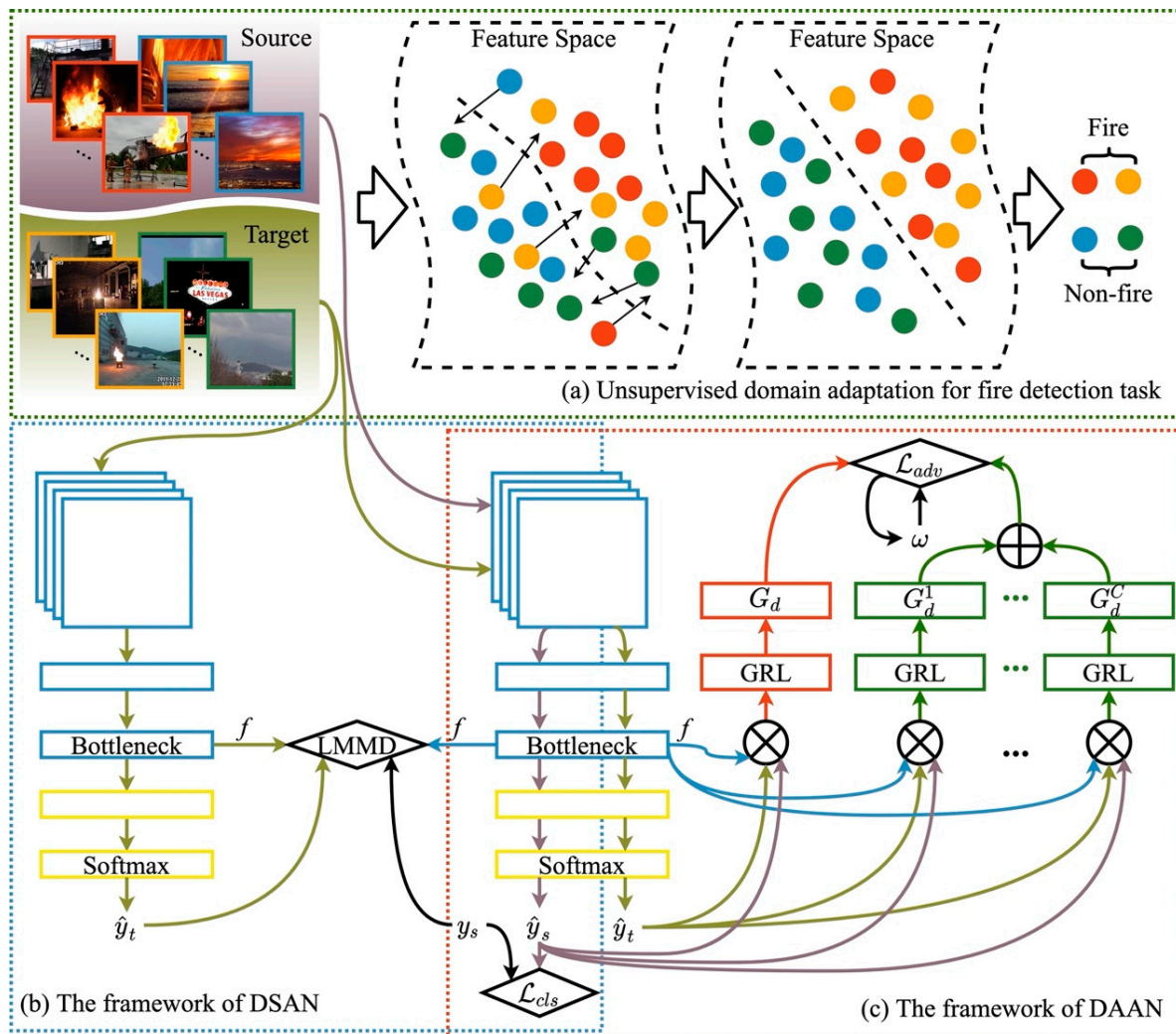


Figure 1. The UDA framework for fire recognition in this paper, including the feature alignment flowchart (a), DSAN (b), and DAAN (c).

As seen from Equation (2), the MMD-based UDAs receive all samples of both domains, resulting in only achieving global alignment and ignoring the effect of the local alignment of the samples from the same category in both domains. DSAN proposes the LMMD to ensure local alignment. Specifically, DSAN divides \mathcal{D}_s and \mathcal{D}_t into C pairs of subdomains according to the category and then performs alignment operations in subdomains separately based on LMMD. LMMD is defined as follows:

$$\text{LMMD}(\mathcal{D}_s, \mathcal{D}_t) = \frac{1}{C} \sum_{c=1}^C \left\| \sum_{x_i \in \mathcal{D}_s} \omega_i^{sc} \phi(x_i) - \sum_{x_j \in \mathcal{D}_t} \omega_j^{tc} \phi(x_j) \right\|_{\mathcal{H}}^2 \quad (4)$$

where ω_i^{sc} and ω_j^{tc} are the probabilities that samples, x_i and x_j belong to the c th category, respectively, and $\sum_{i=1}^{n_s} \omega_i^{sc} = \sum_{j=1}^{n_t} \omega_j^{tc} = 1$. We calculate ω_i^{sc} and ω_j^{tc} by the following equations:

$$\omega_i^{sc} = \frac{y_i^c}{\sum_{(x,y) \in \mathcal{D}_s} y^c} \quad (5)$$

$$\omega_j^{tc} = \frac{\hat{y}_j^c}{\sum_{(x,y) \in \mathcal{D}_t} \hat{y}^c} \quad (6)$$

In Equation (5), we calculate ω_i^{sc} using the real label y_i^c in the source domain. In Equation (6), ω_j^{tc} cannot be calculated directly since the target domain has no label in UDA, so we estimate ω_j^{tc} by the pseudo labels $\hat{y}_j = \mathcal{F}(x_j)$ in the target domain. Ultimately, the loss function of DSAN is expressed below:

$$\mathcal{L} = \mathcal{L}_{cls}(\mathcal{D}_s) + \lambda \text{LMMD}(\mathcal{D}_s, \mathcal{D}_t) \quad (7)$$

The framework of DSAN is illustrated in Figure 1b. The LMMD is deployed at the bottleneck layer and the $\phi(\cdot)$ in Equation (4) indicates the activation of the bottleneck layer.

3.3. DAAN

To achieve transferable knowledge, the adversarial-based UDAs learn domain-invariant features through a “two-player game” of feature extractor and domain discriminator. The loss function of DANN can be derived from Equation (1) as shown below:

$$\begin{aligned} \mathcal{L}(\theta_f, \theta_y, \theta_d) &= \mathcal{L}_{cls}(\mathcal{D}_s) + \lambda \mathcal{L}_{adv}(\mathcal{D}_s, \mathcal{D}_t) \\ &= \frac{1}{n_s} \sum_{(x_i, y_i) \in \mathcal{D}_s} \mathcal{L}_y(\mathcal{G}_y(\mathcal{G}_f(x_i)), y_i) - \frac{\lambda}{n_s + n_t} \sum_{x_i \in (\mathcal{D}_s \cup \mathcal{D}_t)} \mathcal{L}_d(\mathcal{G}_d(\mathcal{G}_f(x_i)), d_i) \end{aligned} \quad (8)$$

where $\mathcal{L}_y(\cdot)$ and $\mathcal{L}_d(\cdot)$ represent the label classifier loss and domain discriminator loss, respectively, $\mathcal{G}_f(\cdot)$, $\mathcal{G}_y(\cdot)$, and $\mathcal{G}_d(\cdot)$ the feature extractor, label classifier, and domain discriminator, respectively, and d_i the domain label (e.g., $d_i = 0$ for the source domain and $d_i = 1$ for the target domain). Similar to the training process of GAN, the parameters θ_f , θ_y and θ_d are alternately optimized by adversarial training of $\mathcal{G}_f(\cdot)$ and $\mathcal{G}_d(\cdot)$ until the network converges. The optimization process is shown in Equations (9) and (10).

$$(\hat{\theta}_f, \hat{\theta}_y) = \operatorname{argmin}_{(\theta_f, \theta_y)} \mathcal{L}(\theta_f, \theta_y, \hat{\theta}_d) \quad (9)$$

$$\hat{\theta}_d = \operatorname{argmax}_{\theta_d} \mathcal{L}(\hat{\theta}_f, \hat{\theta}_y, \theta_d) \quad (10)$$

To make the training process more efficient, DANN designs the GRL between $\mathcal{G}_f(\cdot)$ and $\mathcal{G}_d(\cdot)$ so that the parameters θ_f , θ_y , and θ_d can be optimized simultaneously in back-propagation by one step.

As seen in Equation (8), the domain discriminator of DANN receives all samples in both domains and, thus, DANN only achieves the alignment of the marginal distribution. DAAN achieves conditional and marginal distribution alignment through a structure with C subdomain discriminators $\mathcal{G}_d^c(\cdot)$ and one global discriminator $\mathcal{G}_d(\cdot)$, where C is the

number of categories and the subdomain consists of the samples from the same category in both domains. The loss function of DAAN can be expressed as follows:

$$\begin{aligned}
 \mathcal{L}(\theta_f, \theta_y, \theta_d, \theta_d^c |_{c=1}^C) &= \underbrace{\frac{1}{n_s} \sum_{(x_i, y_i) \in \mathcal{D}_s} \mathcal{L}_y(\mathcal{G}_y(\mathcal{G}_f(x_i)), y_i)}_{\text{Label classifier loss}} \\
 &\quad - \lambda \left((1 - \omega) \underbrace{\frac{1}{n_s + n_t} \sum_{x_i \in (\mathcal{D}_s \cup \mathcal{D}_t)} \mathcal{L}_d(\mathcal{G}_d(\mathcal{G}_f(x_i)), d_i)}_{\text{Global domain discriminator loss}} \right. \\
 &\quad \left. + \omega \underbrace{\frac{1}{n_s + n_t} \sum_{c=1}^C \sum_{x_i \in (\mathcal{D}_s \cup \mathcal{D}_t)} \mathcal{L}_d^c(\mathcal{G}_d^c(\hat{y}_i^c \mathcal{G}_f(x_i)), d_i)}_{\text{Local domain discriminator loss}} \right) \quad (11)
 \end{aligned}$$

where the total loss $\mathcal{L}(\cdot)$ consists of the label classifier loss, global domain discriminator loss, and local domain discriminator loss. λ represents the domain adaptation weights and ω the dynamic adjustment factor between global and local domain discriminators. In local domain discriminator loss, the label classifier uses the predicted output $\hat{y}_i^c = \mathcal{G}_y(\mathcal{G}_f(x_i))$ of the label classifier as the probability that sample x_i belongs to the c th category. The framework of DAAN is illustrated in Figure 1c.

3.4. Fire-DA Dataset

We collect the following popular datasets published recently to build a domain adaptation benchmark dataset for fire recognition. BoWFire is an image fire dataset in urban emergency scenarios containing 119 fire images and 107 non-fire images. It is commonly used as a testing set since it is very challenging [18,19,30,34]. VisiFire is a video fire and smoke dataset for many traditional handcrafted feature-based detection methods, containing 15 fire videos and 24 normal ones [4,13]. The MIVIA fire detection dataset has been used as a training or testing set by many CNN-based fire detection methods [8,16,39]. Further, 27 of the 31 videos are from VisiFire and the authors captured the rest. Among these videos, the first 14 have fires and the last 17 do not. FIRESENCE is a video fire dataset that includes 11 fire videos and 16 non-fire ones, 5 of which are also from VisiFire [10]. KMU Fire & Smoke Database contains 22 fire videos and 16 non-fire ones, with most clips being gasoline or heptane fires captured from afar [10]. We extract frames from the clips according to different sampling rates to obtain a balanced image dataset containing 1000 images per category. Then, a benchmark dataset called Fire-DA is constructed by the following datasets, BoWFire (B), FIRESENCE (F), MIVIA fire detection dataset (M), and KMU Fire & Smoke Database (K), and each of them is considered as a subdomain. It is important to note that we do not consider the VisiFire as a subdomain of Fire-DA since it is very similar to the MIVIA fire detection dataset. Further information is given in Table 1 and some sample images of Fire-DA are shown in Figure 2.

Table 1. The detailed statistics of the Fire-DA.

Dataset	Raw Data				Sampled Images	
	Format	Fire	Non-Fire	Downloading Links (Accessed on 1 November 2022)	Fire	Non-Fire
BoWFire (B)	Image	119	107	https://bitbucket.org/gbdi/bowfire-dataset	119	107
FIRESENSE (F)	Video	11	16	http://doi.org/10.5281/zenodo.836749	1000	1000
MIVIA (M)	Video	14	17	http://mivia.unisa.it	1000	1000
KMU (K)	Video	22	16	https://cvpr.kmu.ac.kr	1000	1000
VisiFire	Video	15	24	http://signal.ee.bilkent.edu.tr/VisiFire	-	-

**Figure 2.** Some sample images from five public fire datasets. Columns one to three show fire samples and columns four to six illustrate non-fire samples.

4. Experiments

In this section, we conduct two experiments. One is the transferability research between subdomains of Fire-DA based on two UDAs: DSAN and DAAN. The other is the performance comparison between traditional handcraft feature-based methods, supervised CNN-based methods, and DSAN on the FLAME and FD-dataset. DSAN and DAAN are implied on the PyTorch framework, and the backbones AlexNet [53] and ResNet50 [54] are pre-trained on ImageNet [55] and fine-tuned on our transfer tasks, respectively. In this paper, the experimental hardware platform is a server with an NVIDIA RTX 2080Ti graphics card with 12 GB of RAM.

4.1. Experiment I: Transferability Research on Fire-DA Based on UDA

4.1.1. Implementation Details

Following the standard protocols of UDA, we build all 12 transfer tasks based on the four subdomains of Fire-DA: $B \rightarrow F$, $F \rightarrow B$, $B \rightarrow M$, $M \rightarrow B$, $B \rightarrow K$, $K \rightarrow B$, $F \rightarrow M$, $M \rightarrow F$, $F \rightarrow K$, $K \rightarrow F$, $M \rightarrow K$, $K \rightarrow M$, using one dataset as the source domain and another as the target domain. The labels in the target domain are only used for evaluation. To verify

the effectiveness of UDA in Fire-DA, we establish a baseline named train-on-source (ToS), which only accesses samples in the source domain during training and then is evaluated directly on the target domain. In other words, when $\lambda = 0$, UDA degenerates to ToS.

We fine-tune all convolutional and pooling layers and train the classifier layers by backpropagation for UDA on Fire-DA. Since the classifier is trained from scratch, a learning rate of 10-times that of the other layers is set. We use minibatch stochastic gradient descent (SGD) with a momentum of 0.9 as an optimization scheme, and the learning rate variation strategy follows existing work [47]. Due to the high computational cost, the learning rate is not chosen by grid search but is adjusted during SGD training with the following formulas, $\eta_\theta = \eta_0 / (1 + \alpha\theta)^\beta$, where θ is the training progress and varies linearly from 0 to 1, $\eta_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$. We fix the adaptation factor $\lambda = 0.5$ in DSAN and $\lambda = 1$ in DAAN, and the *batch size* = 32 in both methods. We report the average classification accuracy with five random experiments for each transfer task and present the standard error.

4.1.2. Results and Analysis

The accuracy of UDA on Fire-DA is shown in Table 2, from which we can draw the following conclusions. Firstly, UDAs improve almost all tasks compared with ToS on Fire-DA, making unsupervised fire recognition possible. Specifically, as to the average accuracy of 12 transfer tasks, DAAN + AlexNet and DSAN + AlexNet improve by 10.6% and 10.7%, respectively, over ToS + AlexNet, as well as DAAN + ResNet50 and DSAN + ResNet50 improve by 4.8% and 14.5%, respectively, over ToS + ResNet50. It indicates that the performances of UDA on the target domain can be improved by aligning cross-domain feature distribution without accessing the sample labels of the target domain during training. Secondly, DSAN performs better than DAAN when the backbones are the same, indicating that DSAN has a more substantial domain adaptation capability than DAAN on Fire-DA. Specifically, DSAN + AlexNet improves by 0.1% over DAAN + AlexNet, and DSAN + ResNet50 improves by 9.2% over DAAN + ResNet50. Finally, the combination of DSAN and ResNet50 achieves the highest average accuracy (85.2%), which is 8.8% better than the suboptimal DSAN + AlexNet. It yields the highest accuracy in 8 out of 12 transfer tasks, owing to the excellent domain adaptation capability of DSAN and the powerful feature extraction and representation learning capability of ResNet. In addition, we obtain some interesting findings, such as task B→F shows high accuracy in most cases, and the possible reasons are the slight domain shift between B and F and the poor category separability of the source domain B. Through the qualitative analysis in Figure 2 and the quantitative analysis from the ToS results in Table 2, the slight domain shift of task B→F makes knowledge transfer easier. Meanwhile, B contains many strong interference samples despite the small sample size to make the model obtain better classification performance, both of which lead to the high accuracy of task B→F. On the contrary, the category separability of F is better than B, making the classification performance of the model obtained by training on F degraded. Thus, the accuracy of task F→B is not as good as task B→F.

To verify the effectiveness of UDA and the performance gap between different UDAs, we visualize the learned feature representations using t-SNE on task M-K. Figure 3a–c illustrate the learned feature representations of the source domain M and target domain K based on the following three methods, ToS + ResNet50, DAAN + ResNet50, and DSAN + ResNet50, respectively. As shown in Figure 3a, when trained with only source domain samples, the network performs well in classification on the source domain but poorly on the target domain due to the distribution discrepancy between the source and target domains. As to Figure 3b,c, when the UDAs are introduced to align the feature distributions in the source and target domains, the sample features from the same category tend to be aggregated, ensuring that the classification boundary obtained by supervised learning on the source domain is also discriminative for the target domain. Comparing Figure 3b,c, DSAN confuses the sample features of the same category in both domains together, further improving the network's accuracy on the target domain.

Table 2. Accuracy (%) on Fire-DA for unsupervised domain adaptation. The values show the average classification accuracy with five random experiments and the standard error.

Method	B→F	F→B	B→M	M→B	B→K	K→B	F→M	M→F	F→K	K→F	M→K	K→M	Avg
ToS + AlexNet	89.9 ± 2.1	70.4 ± 1.4	70.9 ± 3.1	65.3 ± 2.2	55.0 ± 2.4	63.1 ± 0.7	78.4 ± 0.2	87.0 ± 1.0	51.2 ± 0.8	70.0 ± 1.8	72.9 ± 0.7	74.3 ± 2.3	70.7
DAAN + AlexNet	91.0 ± 1.8	75.0 ± 3.4	79.8 ± 2.5	65.1 ± 1.3	87.1 ± 0.8	66.4 ± 1.9	82.3 ± 1.5	89.3 ± 1.2	74.4 ± 0.7	77.6 ± 3.0	74.6 ± 4.0	76.2 ± 1.0	78.2
DSAN + AlexNet	94.9 ± 3.3	71.9 ± 1.9	78.6 ± 2.8	73.4 ± 1.7	81.6 ± 1.7	66.0 ± 0.9	79.0 ± 1.2	88.3 ± 1.2	74.1 ± 0.7	77.3 ± 1.3	75.4 ± 0.4	79.0 ± 1.7	78.3
ToS + ResNet50	87.4 ± 1.5	78.1 ± 1.6	86.3 ± 1.3	74.1 ± 1.4	57.3 ± 3.1	68.5 ± 1.1	80.1 ± 0.7	83.8 ± 1.8	48.0 ± 1.1	76.7 ± 3.2	71.6 ± 1.8	80.5 ± 2.2	74.4
DAAN + ResNet50	91.5 ± 1.8	79.9 ± 1.9	88.0 ± 1.4	78.9 ± 2.6	69.1 ± 2.0	71.2 ± 1.1	81.7 ± 1.4	87.8 ± 3.2	46.9 ± 2.5	80.3 ± 3.6	78.4 ± 3.6	82.1 ± 2.1	78.0
DSAN + ResNet50	91.0 ± 2.8	84.4 ± 3.9	89.7 ± 1.6	74.7 ± 1.8	95.3 ± 3.1	83.0 ± 0.9	81.9 ± 0.8	90.1 ± 1.2	62.3 ± 1.4	83.8 ± 2.2	97.1 ± 1.6	89.5 ± 2.3	85.2

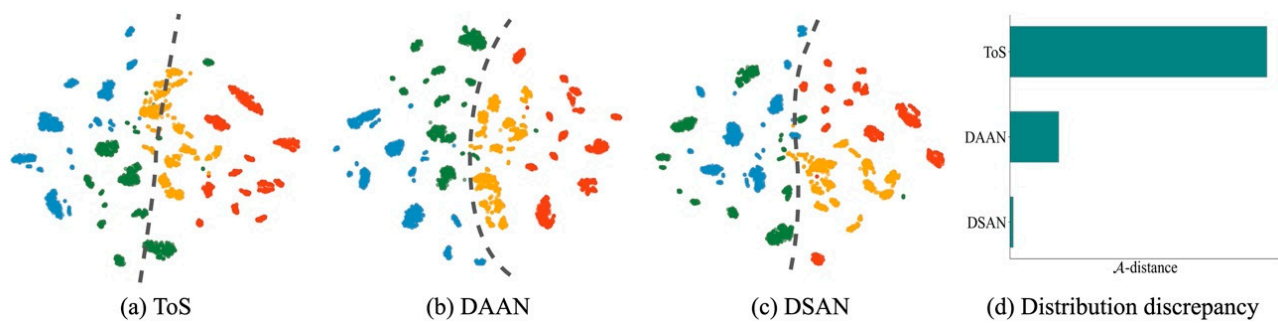


Figure 3. (a–c) Visualizations of the learned feature representations using t-SNE on task M→K w/o or w/ UDA. The red and yellow circles represent the source and target domain fire samples, respectively. The blue and green circles represent the source and target domain non-fire samples, respectively. The grey dashed lines indicate possible classification boundaries. (d) Distribution discrepancy between domain M and K measured by A-distance.

Further, based on the theoretical results of [56], we quantify the feature distribution discrepancy between domains using A-distance. Since it is challenging to compute the exact value of A-distance, a general approximation defines $A\text{-distance} = 2(1 - 2\sigma)$, where σ represents the generalization error of a trained binary classifier to discriminate input samples from the source and target domains. A larger A-distance means a more significant distribution gap between domains. Figure 3d presents the A-distances on task M-K under those three different methods in Figure 3a–c, respectively. It can be concluded that the use of UDA can effectively reduce the A-distance by a large margin, and DSAN outperforms DAAN, which is consistent with the classification accuracy results shown in Table 1 and the feature visualization findings in Figure 3a–c.

4.2. Experiment II: Comparison of Different Fire Recognition Methods on FLAME and FD-Dataset

4.2.1. Dataset

The FD-dataset is a large-scale fire detection dataset consisting of 25,000 fire and 25,000 non-fire images [21]. It is based on MIVIA fire detection and BoWFire and is enriched with many pictures from the Internet. The fire images are captured from diverse scenarios, such as fires on cars, buildings, boats, forests, etc. The non-fire images contain several fire-like objects, such as fallen yellow leaves, red cars, flying flags, burning clouds, sunsets, and glaring lights. The original paper divided the data into the training, validation, and testing sets with a ratio of 7:2:1. It also selected the most challenging images from the testing set to create a challenging testing set with 250 fire and 250 non-fire images. FD-dataset is not a dedicated forest fire dataset but contains many positive and negative samples for forest fire recognition. FLAME is a forest fire dataset collected by UAV in an Arizona pine forest [20], as shown in Figure 4. It is captured by Zenmuse X4S, Phantom 3, both from DJI company (Shenzhen, China), and annotated frame by frame to obtain a training/validation set of 39,375 images and a testing set of 8617 images. Compared with Fire-DA and FD-dataset, the FLAME dataset has fewer scenes, the camera views are primarily top-down and wide range, and the flames are smaller in the frame. Some representative sample images of the FLAME and FD-dataset are illustrated in Figure 5.

Since there is no benchmark image/video dataset in fire recognition, we fuse the image samples from the four subdomains in Fire-DA to form a new composite fire recognition dataset with diverse real-world scenery environments. As a result, the fused Fire-DA contains 3119 fire images and 3107 non-fire images. Then, we build two UDA-based transfer tasks: Fire-DA→FLAME and Fire-DA→FD-dataset for fire recognition and compare them with the traditional handcrafted feature-based and supervised CNN-based methods. Notably, in the transfer task of Fire-DA→FD-dataset, we remove the images of Fire-DA that appeared in the FD-dataset in order to make the comparison more rigorous.

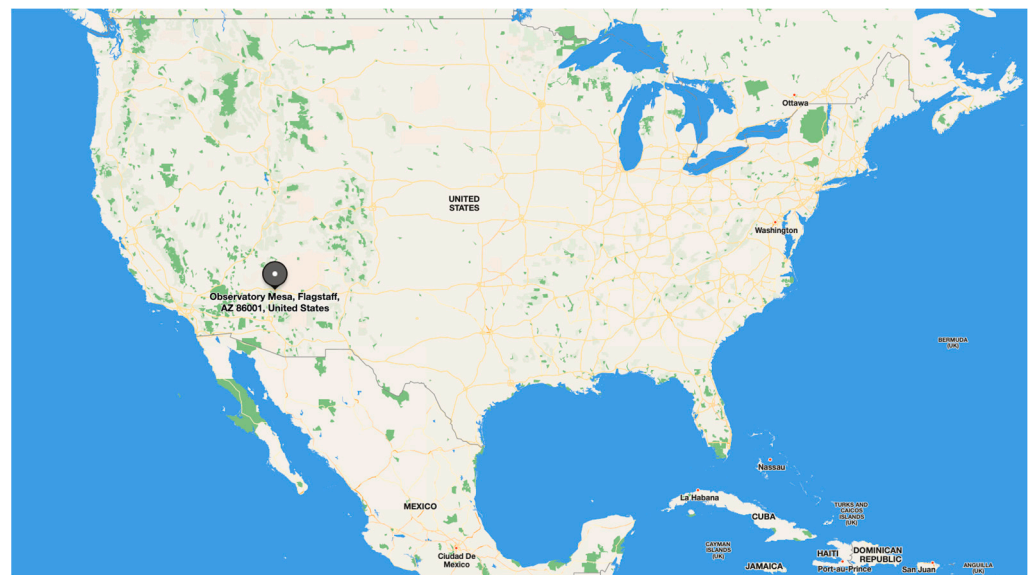


Figure 4. The FLAME is collected in Flagstaff, Arizona, USA, Ponderosa pine forest on Observatory Mesa.



Figure 5. Some sample images from the FLAME and FD-dataset. The first row shows fire samples and the second row illustrates non-fire samples for each dataset.

4.2.2. Implementation Details

Considering that DSAN performs better than DAAN in Experiment I, we adopt DSAN as the UDA-based method in this experiment. We treat the fused Fire-DA as the source domain with labels and each training set of the FLAME and FD-dataset as the target domain without accessing labels during training and evaluate metrics on their testing sets for performance comparison. The details of the experiment settings are shown in Table 3. Furthermore, AlexNet and ResNet50 are used as the backbones, pre-trained on ImageNet, and fine-tuned on these transfer tasks. We fix the adaptation factor $\lambda = 5.0$ for both transfer tasks. The other hyperparameters are the same as the settings of DSAN in Experiment I.

Table 3. The detailed statistics of training and testing sets for DSAN.

Transfer Task	Data Division		Fire Images	Non-Fire Images	Labels
Fire-DA→FLAME	Training	Fire-DA (source)	3119	3107	✓
		The training set of FLAME (target)	25,018	14,357	×
	Testing	The testing set of FLAME	5137	3480	✓
Fire-DA→FD-dataset	Training	Fire-DA (source)	2654	3107	✓
		The training set of FD-dataset (target)	17,500	17,500	×
	Testing	The testing set of FD-dataset	2500	2500	✓
		The challenging testing set of FD-dataset	250	250	✓

Then, we set up two baselines with different backbones, train on source (ToS) and train on target (ToT). ToS is the same as the baseline in Experiment I, which will obtain an undesirable result on the target domain because of the domain shift. In contrast, ToT is the traditional supervised CNN-based approach on the target domain with labels, which will yield promising results. The result obtained by ToS and ToT can be regarded as the lower and upper bound for the performance of UDA, respectively.

In addition, we also discuss the recognition performance of DSAN with the traditional methods based on handcrafted features and supervised CNNs. For FLAME [20], the supervised CNNs are based on Xception [20] and ResNet50 with sample augmentation (SA) [38]. For the FD-dataset, the traditional methods are based on heuristic rules [3,57], fuzzy inference [57], and BP neural network [58], respectively, and the supervised CNNs are based on AlexNet [16], ResNet50 [17], GoogleNet [18], SqueezeNet [19], improved lightweight AlexNet [21], and MobileNet [30]. The quantitative results of these methods are from [20,21,38]. As for our transfer tasks, the average results of the evaluation metrics are reported with three random experiments.

4.2.3. Evaluation Metrics

The evaluation metrics used to validate the performance of the fire recognition methods in this section include recall, precision, accuracy, and *F1* score, which are expressed as follows:

$$recall = \frac{TP}{TP + FN} \quad (12)$$

$$precision = \frac{TP}{TP + FP} \quad (13)$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$F1 = 2 \times \frac{recall \times precision}{recall + precision} \quad (15)$$

where *TP*, *FP*, *FN*, and *TN* represent True-Positive, False-Positive, False-Negative, and True-Negative values, respectively.

4.2.4. Results and Analysis

The average results of evaluation metrics on the testing set of the FLAME are shown in Table 4. DSAN is inferior to the supervised CNNs in all metrics. Ref. [20] uses Xception to achieve 76.2% recognition accuracy. Ref. [38] uses ResNet50 and combines it with data augmentation to improve the recognition accuracy to 77.5%. However, the recognition accuracy of DSAN is 63.9%. The possible reasons are as follows: 1. never forget that DSAN does not access the labels of FLAME during training but only transfers the domain knowledge from Fire-DA to FLAME and then evaluates the performance on the testing set of FLAME; 2. through the qualitative analysis of the sample images in Figures 2 and 5, as well as the quantitative analysis of the results of ToS, the domain shift between Fire-DA

and FLAME is relatively large, which leads to a destructive impact on the transfer effect; 3. the UDA is under such a setting that the target domain labels are not accessible during training, resulting in the inability to perform model selection as in supervised CNNs; 4. to obtain more convincing results, we report the average result of three randomized experiments instead of the best one, which is also lossy. Nevertheless, DSAN reaches 82.5% performance in [38]. Under the same training conditions and settings, the accuracy of DSAN is improved by 41.4% over ToS to 94.7% of ToT. From the side, DSAN achieves an impressive performance on the transfer task Fire-DA \rightarrow FLAME.

Table 4. Quantitative results of supervised CNNs and DSAN on the testing set of FLAME.

	Method	Recall	Precision	Accuracy	F1-Score
Supervised CNNs	Xception [20]			76.2	
	ResNet50 w/SA [38]	82.0	80.6	77.5	0.813
	ToT + ResNet50 (Upper bound)	86.5	68.3	67.5	0.761
Unsupervised DSAN (Ours)	ToS + ResNet50 (Lower bound)	8.3	98.3	45.2	0.152
	DSAN + ResNet50	60.7	78.2	63.9	0.662

The average results of evaluation metrics on the testing set of the FD-dataset are shown in Table 5. Firstly, in terms of overall metrics, DSAN is better than ToS and close to ToT. Specifically, the accuracy of DSAN + AlexNet is improved by 8.3% over ToS + AlexNet to 98.1% of ToT + AlexNet, while the accuracy of DSAN + ResNet50 is improved by 14.2% over ToS + ResNet50 to 98.3% of ToT + ResNet50. The results are consistent with the conjecture made when setting up the baselines, indicating that DSAN can improve the recognition performance in the FD-dataset by domain adaptation. Secondly, DSAN is substantially better than traditional methods based on handcrafted features. We can see that even ToS, regarded as the lower bound of UDA, shows significant improvements in all metrics except for recall. The accuracy is 85.7% for ToS + AlexNet and 83.9% for ToS + ResNet50, which improved by 19.5% and 17.0% over the best result in [58]. However, the accuracy of ToS is not sufficient for practical applications. As for DSAN, all metrics are increased, with DSAN + AlexNet and DSAN + ResNet50 achieving 92.8% and 95.8% accuracy, which is 29.4% and 33.6% better than the best result in [58], respectively. Even though [57] has a higher recall than DSAN with 99.9%, it has the lowest accuracy with only 53.9%. The reasons for the improvement are: 1. CNNs have stronger feature extraction and representation learning capabilities than the handcrafted feature-based methods; 2. due to the similarity in distribution between the Fire-DA and FD-dataset, ToS trained on Fire-DA achieves a decent performance on the FD-dataset; 3. DSAN transfers knowledge from Fire-DA to FD-dataset through domain adaptation, leading to a good performance on FD-dataset. Lastly, DSAN achieves better performance compared to supervised CNNs on the FD-dataset. Considering that [18,19,21,30] use lightweight network structures, we compare DSAN + AlexNet with [16] since they are all based on AlexNet, and DSAN + ResNet50 with [17], since they are all based on ResNet50, for the sake of fairness. In terms of accuracy, precision, and F1 score, DSAN + AlexNet outperforms [16] by 6.4%, 10.9%, and 5.7%, respectively, and DSAN + ResNet50 exceeds [17] by 6.4%, 15.6%, and 5.5%, respectively. Regarding recall, DSAN + AlexNet is 0.3% superior to [16], but DSAN + ResNet50 is 4.1% inferior to [17]. We should note that [21] yields the best result of the supervised CNN-based methods by extending the multiscale feature extraction, implicit deep supervision, and channel attention mechanisms to AlexNet for fire recognition. Compared with [21], DSAN + AlexNet achieves poorer but close results on all metrics. In contrast, DSAN + ResNet50 achieves better accuracy, precision, and F1-score results, improving by 0.5%, 4.8%, and 0.3%, respectively, but its recall is 3.9% lower than [21].

The average results of evaluation metrics on the challenging testing set of FD-dataset are shown in Table 6. The metrics of all methods degrade since the challenging testing set contains more strong-interference fire-like images, but the findings are similar to those of the testing set, except for the F1 score of DSAN + ResNet50. Considering the limitation of

the page in this section, we expand on this, in detail, in Appendix A. As a result, DSAN achieves competitive performance with the supervised CNNs on this challenging testing set. It also demonstrates that DSAN is robust to different and challenging datasets.

Table 5. Quantitative results of traditional handcraft feature-based methods, supervised CNNs, and DSAN on the testing set of FD-dataset.

Method		Recall	Precision	Accuracy	F1-Score
Traditional	[57]	99.9	52.0	53.9	0.684
	[3]	90.0	63.9	69.6	0.747
	[58]	73.2	71.1	71.7	0.721
Supervised CNNs	[16]	93.2	83.3	87.2	0.879
	[18]	98.0	88.0	92.3	0.928
	[19]	91.3	84.6	87.3	0.879
	[30]	98.7	88.3	92.8	0.932
	[17]	97.6	84.8	90.0	0.907
	[21]	97.4	93.5	95.3	0.954
	ToT + AlexNet (Upper bound)	97.2	92.3	94.6	0.947
	ToT + ResNet50 (Upper bound)	98.8	96.3	97.5	0.976
Unsupervised DSAN (Ours)	ToS + AlexNet (Lower bound)	75.1	95.3	85.7	0.840
	ToS + ResNet50 (Lower bound)	69.2	98.1	83.9	0.812
	DSAN + AlexNet	93.5	92.4	92.8	0.929
	DSAN + ResNet50	93.6	98.0	95.8	0.957

Table 6. Quantitative results of supervised CNNs and DSAN on the challenging testing set of FD-dataset.

Method		Recall	Precision	Accuracy	F1-Score
Supervised CNNs	[16]	82.8	57.7	61.0	0.680
	[18]	84.8	72.9	76.6	0.784
	[19]	87.6	62.0	67.0	0.726
	[30]	85.2	75.8	79.0	0.802
	[17]	89.6	72.7	78.0	0.803
	[21]	86.4	78.5	81.4	0.823
	ToT+ AlexNet (Upper bound)	89.6	78.4	82.5	0.836
	ToT + ResNet50 (Upper bound)	92.1	88.2	89.9	0.901
Unsupervised DSAN (Ours)	ToS + AlexNet (Lower bound)	30.0	75.0	60.0	0.428
	ToS + ResNet50 (Lower bound)	30.3	92.3	63.9	0.456
	DSAN + AlexNet	77.3	78.2	77.7	0.777
	DSAN + ResNet50	72.8	92.7	83.5	0.815

To verify the effectiveness of DSAN on the transfer task Fire-DA→FD-dataset, we visualize the learned feature representations using t-SNE on Fire-DA and the testing set of the FD-dataset. Figure 6a,b illustrate the learned feature representations of Fire-DA and the testing set of FD-dataset, without domain adaptation and with domain adaptation, respectively. As shown in Figure 6a, ToS performs poorly on the testing set of FD-dataset as there are many misclassified samples according to the classification boundary trained on Fire-DA. As for Figure 6b, the feature distributions of both domains are confused by DSAN, resulting in the classification boundary performing well on both datasets. Figure 6c presents the A-distance between Fire-DA and the testing set of the FD-dataset. The distribution discrepancy of the two datasets is decreased by DSAN, allowing DSAN to obtain good performance on the FD-dataset in an unsupervised way.

As a result, DSAN achieves an impressive performance on FLAME and a new state of the art on the FD-dataset in an unsupervised way, a trade-off between recognition accuracy and the cost of labeling.

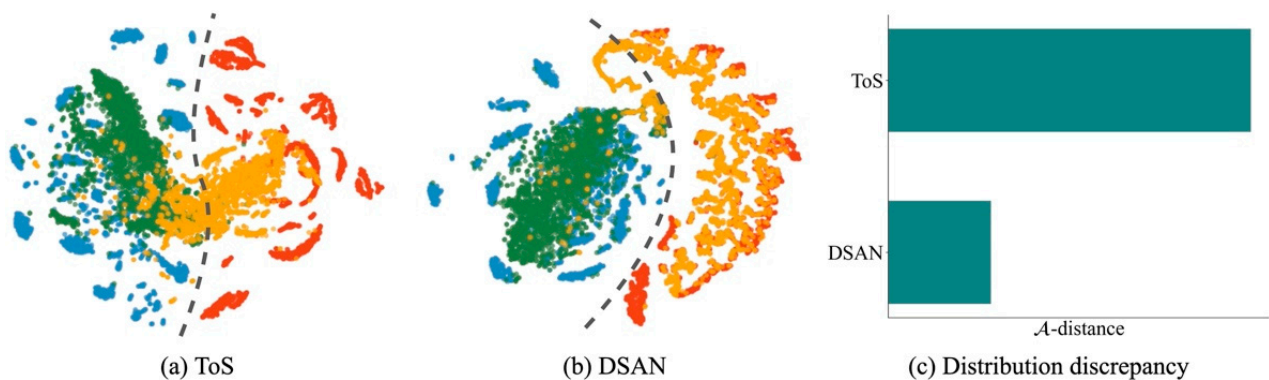


Figure 6. Visualizations of the learned feature representation using t-SNE on Fire-DA and the testing set of FD-dataset, (a) w/o domain adaptation, and (b) w/ domain adaptation. The red and yellow circles represent fire samples in Fire-DA and the testing set of FD-dataset, respectively. The blue and green circles represent non-fire samples in Fire-DA and the testing set of FD-dataset, respectively. The grey dashed lines indicate possible classification boundaries. (c) Distribution discrepancy between Fire-DA and the testing set of FD-dataset measured by A-distance.

5. Conclusions

Recently, the capabilities of DNNs have shown promising results in fire detection tasks. However, DNNs rely on large-scale datasets for supervised training, but labeling fire samples is often cumbersome. First, we apply UDA to transfer the knowledge learned from publicly available fire datasets to fire recognition tasks in practical application scenarios, such as forests. Then, Fire-DA is proposed as a benchmark dataset for future transfer learning research in fire recognition, and the benchmark result is obtained via DSAN and DAAN. Finally, DSAN achieves a better trade-off between the performance and the labeling cost for forest fire recognition than the existing handcrafted feature-based and supervised CNN-based methods, which will drive forest fire recognition and monitoring in a more intelligent direction. Future work will focus on unsupervised forest fire recognition in the following three areas: improving UDA to enhance performance, investigating model selection for UDA, and building a transfer learning pipeline from virtual to real fires.

Author Contributions: Conceptualization, Z.Y.; methodology, Z.Y.; software, Z.Y.; validation, K.Q. and F.Z.; formal analysis, J.O. and T.W.; investigation, Z.Y.; resources, X.H. and L.B.; data curation, Z.Y.; writing—original draft preparation, Z.Y.; writing—review and editing, L.B. and L.W.; visualization, Z.Y.; supervision, L.W.; project administration, X.H.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 41771487, No. 41974005) and the National Science Fund for Distinguished Young Scholars (No. 42122025).

Data Availability Statement: Publicly available datasets were analyzed in this study. The download links can be found here: FD-dataset in [21], FLAME in [20], and others in Table 1. The original codes of DAAN and DSAN are available at <https://github.com/jindongwang/transferlearning> (accessed on 1 November 2022). The modified codes in this paper and the Fire-DA dataset are available from the authors upon request.

Acknowledgments: This paper is a result of research conducted in the School of Electrical Engineering, Naval University of Engineering, which involved work supported by the Naval Armament Department of PLA.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Discussion of Table 6

As shown in Table 6, DSAN is better than ToS and close to ToT in terms of overall metrics. Specifically, the accuracy of DSAN + AlexNet is improved by 29.5% over ToS +

AlexNet to 94.2% of ToT + AlexNet, while the accuracy of DSAN + ResNet50 is enhanced by 30.7% over ToS + ResNet50 to 92.9% of ToT + ResNet50. The results of ToS show that the domain shift between Fire-DA and the challenging testing set of FD-dataset are significant. However, DSAN still achieves a vast improvement, indicating that DSAN can handle the case with a significant domain shift. Moreover, DSAN fulfills competitive performance with supervised CNNs on the challenging testing set of FD-dataset. In terms of accuracy, precision, and F1 score, DSAN + AlexNet outperforms [16] by 27.4%, 35.5%, and 14.3%, respectively, and DSAN + ResNet50 exceeds [17] by 7.1%, 27.5%, and 1.5%, respectively. Regarding recall, DSAN + AlexNet is inferior to [16] by 6.6% and DSAN + ResNet50 is to [17] by 18.8%. Compared with [21], DSAN + AlexNet achieves poorer but close results on all metrics. In contrast, DSAN + ResNet50 achieves better accuracy and precision than [21], with a 2.6% and 18.1% improvement, respectively, but lower recall and F1 score, with a 15.7% and 1.0% decrease, respectively.

References

1. Available online: <https://www.119.gov.cn/article/46TiYamnnrs> (accessed on 1 August 2022).
2. Ahrens, M.; Evarts, B. Fire Loss in the United States During 2021. Available online: <https://www.nfpa.org/News-and-Research/Data-research-and-tools/US-Fire-Problem/Fire-loss-in-the-United-States> (accessed on 1 August 2022).
3. Çelik, T.; Demirel, H. Fire detection in video sequences using a generic color model. *Fire Saf. J.* **2009**, *44*, 147–158. [CrossRef]
4. Borges, P.V.K.; Izquierdo, E. A probabilistic approach for vision-based fire detection in videos. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 721–731. [CrossRef]
5. Qiu, T.; Yan, Y.; Lu, G. An autoadaptive edge-detection algorithm for flame and fire image processing. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1486–1493. [CrossRef]
6. Günay, O.; Taşdemir, K.; Töreyn, B.U.; Çetin, A.E. Fire detection in video using LMS based active learning. *Fire Technol.* **2010**, *46*, 551–577. [CrossRef]
7. Verstockt, S.; Van Hoecke, S.; Tilley, N.; Merci, B.; Sette, B.; Lambert, P.; Hollemeersch, C.-F.J.; Van De Walle, R. FireCube: A multi-view localization framework for 3D fire analysis. *Fire Saf. J.* **2011**, *46*, 262–275. [CrossRef]
8. Foggia, P.; Saggese, A.; Vento, M. Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1545–1556. [CrossRef]
9. Stadler, A.; Windisch, T.; Diepold, K. Comparison of intensity flickering features for video based flame detection algorithms. *Fire Saf. J.* **2014**, *66*, 1–7. [CrossRef]
10. Dimitropoulos, K.; Barmpoutis, P.; Grammalidis, N. Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 339–351. [CrossRef]
11. Qureshi, W.S.; Ekpanyapong, M.; Dailey, M.N.; Rinsurongkawong, S.; Malenichev, A.; Krasotkina, O. QuickBlaze: Early Fire Detection Using a Combined Video Processing Approach. *Fire Technol.* **2016**, *52*, 1293–1317. [CrossRef]
12. Gong, F.; Li, C.; Gong, W.; Li, X.; Yuan, X.; Ma, Y.; Song, T. A real-time fire detection method from video with multifeature fusion. *Comput. Intell. Neurosci.* **2019**, *2019*, 1939171. [CrossRef]
13. Ko, B.; Cheong, K.-H.; Nam, J.-Y. Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian Networks. *Fire Saf. J.* **2010**, *45*, 262–270. [CrossRef]
14. Kong, S.G.; Jin, D.; Li, S.; Kim, H. Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. *Fire Saf. J.* **2016**, *79*, 37–43. [CrossRef]
15. Mueller, M.; Karasev, P.; Kolesov, I.; Tannenbaum, A. Optical flow estimation for flame detection in videos. *IEEE Trans. Image Process.* **2013**, *22*, 2786–2797. [CrossRef] [PubMed]
16. Muhammad, K.; Ahmad, J.; Baik, S.W. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* **2018**, *288*, 30–42. [CrossRef]
17. Sharma, J.; Granmo, O.-C.; Goodwin, M.; Fidge, J.T. Deep convolutional neural networks for fire detection in images. *Commun. Comput. Inf. Sci.* **2017**, *744*, 183–193. [CrossRef]
18. Muhammad, K.; Ahmad, J.; Mehmood, I.; Rho, S.; Baik, S.W. Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access* **2018**, *6*, 18174–18183. [CrossRef]
19. Muhammad, K.; Ahmad, J.; Lv, Z.; Bellavista, P.; Yang, P.; Baik, S.W. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1419–1434. [CrossRef]
20. Shamsoshoara, A.; Afghah, F.; Razi, A.; Zheng, L.; Fulé, P.Z.; Blasch, E. Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Comput. Netw.* **2021**, *193*, 108001. [CrossRef]
21. Li, S.; Yan, Q.; Liu, P. An Efficient Fire Detection Method Based on Multiscale Feature Extraction, Implicit Deep Supervision and Channel Attention Mechanism. *IEEE Trans. Image Process.* **2020**, *29*, 8467–8475. [CrossRef]
22. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep subdomain adaptation network for image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 1713–1722. [CrossRef] [PubMed]

23. Yu, C.; Wang, J.; Chen, Y.; Huang, M. Transfer learning with dynamic adversarial adaptation network. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 778–786. [\[CrossRef\]](#)
24. Gaur, A.; Singh, A.; Kumar, A.; Kumar, A.; Kapoor, K. Video Flame and Smoke Based Fire Detection Algorithms: A Literature Review. *Fire Technol.* **2020**, *56*, 1943–1980. [\[CrossRef\]](#)
25. Hu, C.; Tang, P.; Jin, W.; He, Z.; Li, W. Real-Time Fire Detection Based on Deep Convolutional Long-Recurrent Networks and Optical Flow Method. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 9061–9066. [\[CrossRef\]](#)
26. Dunnings, A.J.; Breckon, T.P. Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1358–1362. [\[CrossRef\]](#)
27. Huang, L.; Liu, G.; Wang, Y.; Yuan, H.; Chen, T. Fire detection in video surveillances using convolutional neural networks and wavelet transform. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104737. [\[CrossRef\]](#)
28. Majid, S.; Alenezi, F.; Masood, S.; Ahmad, M.; Gündüz, E.S.; Polat, K. Attention based CNN model for fire detection and localization in real-world images. *Expert Syst. Appl.* **2022**, *189*, 116114. [\[CrossRef\]](#)
29. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A forest fire detection system based on ensemble learning. *Forests* **2021**, *12*, 217. [\[CrossRef\]](#)
30. Muhammad, K.; Khan, S.; Elhoseny, M.; Ahmed, S.H.; Baik, S.W. Efficient Fire Detection for Uncertain Surveillance Environment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3113–3122. [\[CrossRef\]](#)
31. Li, P.; Zhao, W. Image fire detection algorithms based on convolutional neural networks. *Case Stud. Therm. Eng.* **2020**, *19*, 100625. [\[CrossRef\]](#)
32. Park, M.; Ko, B.C. Two-step real-time night-time fire detection in an urban environment using static elastic-yolov3 and temporal fire-tube. *Sensors* **2020**, *20*, 2202. [\[CrossRef\]](#)
33. Barmpoutis, P.; Stathaki, T.; Dimitropoulos, K.; Grammalidis, N. Early fire detection based on aerial 360-degree sensors, deep convolution neural networks and exploitation of fire dynamic textures. *Remote Sens.* **2020**, *12*, 3177. [\[CrossRef\]](#)
34. Choi, H.-S.; Jeon, M.; Song, K.; Kang, M. Semantic Fire Segmentation Model Based on Convolutional Neural Network for Outdoor Image. *Fire Technol.* **2021**, *57*, 3005–3019. [\[CrossRef\]](#)
35. Yang, Z.; Wang, T.; Bu, L.; Ouyang, J. Training with Augmented Data: GAN-based Flame-Burning Image Synthesis for Fire Segmentation in Warehouse. *Fire Technol.* **2021**, *58*, 183–215. [\[CrossRef\]](#)
36. Kou, L.; Wang, X.; Guo, X.; Zhu, J.; Zhang, H. Deep learning based inverse model for building fire source location and intensity estimation. *Fire Saf. J.* **2021**, *121*, 103310. [\[CrossRef\]](#)
37. Qin, K.; Hou, X.; Yan, Z.; Zhou, F.; Bu, L. FGL-GAN: Global-Local Mask Generative Adversarial Network for Flame Image Composition. *Sensors* **2022**, *22*, 6332. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Zhang, L.; Wang, M.; Fu, Y.; Ding, Y. A Forest Fire Recognition Method Using UAV Images Based on Transfer Learning. *Forests* **2022**, *13*, 975. [\[CrossRef\]](#)
39. Shahid, M.; Virtusio, J.J.; Wu, Y.-H.; Chen, Y.-Y.; Tanveer, M.; Muhammad, K.; Hua, K.-L. Spatio-Temporal Self-Attention Network for Fire Detection and Segmentation in Video Surveillance. *IEEE Access* **2022**, *10*, 1259–1275. [\[CrossRef\]](#)
40. Jeon, M.; Choi, H.-S.; Lee, J.; Kang, M. Multi-Scale Prediction For Fire Detection Using Convolutional Neural Network. *Fire Technol.* **2021**, *57*, 2533–2551. [\[CrossRef\]](#)
41. Zhong, Z.; Wang, M.; Shi, Y.; Gao, W. A convolutional neural network-based flame detection method in video sequence. *Signal Image Video Process.* **2018**, *12*, 1619–1627. [\[CrossRef\]](#)
42. Xie, Y.; Zhu, J.; Cao, Y.; Zhang, Y.; Feng, D.; Zhang, Y.; Chen, M. Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features. *IEEE Access* **2020**, *8*, 81904–81917. [\[CrossRef\]](#)
43. Zhao, Y.; Ma, J.; Li, X.; Zhang, J. Saliency detection and deep learning-based wildfire identification in uav imagery. *Sensors* **2018**, *18*, 712. [\[CrossRef\]](#)
44. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
45. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning transferable features with deep adaptation networks. In Proceedings of the 32nd International Conference on Machine Learning, ICML, Lille, France, 7 July 2015; pp. 97–105.
46. Sun, B.; Saenko, K. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin, Germany, 2016; Volume 9915, pp. 443–450. [\[CrossRef\]](#)
47. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the 32nd International Conference on Machine Learning, ICML, Lille, France, 7 July 2015; Volume 2, pp. 1180–1189.
48. Pei, Z.; Cao, Z.; Long, M.; Wang, J. Multi-Adversarial Domain Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 3934–3941.
49. Wang, M.; Deng, W.; Liu, C.-L. Unsupervised Structure-Texture Separation Network for Oracle Character Recognition. *IEEE Trans. Image Process.* **2022**, *31*, 3137–3150. [\[CrossRef\]](#)
50. Zhao, T.; Shen, Z.; Zou, H.; Zhong, P.; Chen, Y. Unsupervised adversarial domain adaptation based on interpolation image for fish detection in aquaculture. *Comput. Electron. Agric.* **2022**, *198*, 107004. [\[CrossRef\]](#)

51. Liu, W.; Luo, Z.; Cai, Y.; Yu, Y.; Ke, Y.; Junior, J.M.; Gonçalves, W.N.; Li, J. Adversarial unsupervised domain adaptation for 3D semantic segmentation with multi-modal learning. *ISPRS J. Photogramm. Remote Sens.* **2021**, *176*, 211–221. [[CrossRef](#)]
52. Wang, W.; Zhao, F.; Liao, S.; Shao, L. Attentive WaveBlock: Complementarity-Enhanced Mutual Networks for Unsupervised Domain Adaptation in Person Re-Identification and Beyond. *IEEE Trans. Image Process.* **2022**, *31*, 1532–1544. [[CrossRef](#)]
53. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *6*, 84–90. [[CrossRef](#)]
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
55. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
56. Ben-David, S.; Blitzer, J.; Crammer, K.; Pereira, F. Analysis of representations for domain adaptation. *Adv. Neural Inf. Process. Syst.* **2007**, *19*, 137–144. [[CrossRef](#)]
57. Celik, T.; Ozkaramanli, H.; Demirel, H. Fire pixel classification using fuzzy logic and statistical color model. In Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing—ICASSP '07, Honolulu, HI, USA, 15–20 April 2007; Volume 1, pp. I-1205–I-1208. [[CrossRef](#)]
58. Zhang, D.; Han, S.; Zhao, J.; Zhang, Z.; Qu, C.; Ke, Y.; Chen, X. Image based Forest fire detection using dynamic characteristics with artificial neural networks. In Proceedings of the 2009 International Joint Conference on Artificial Intelligence, Hainan, China, 25–26 April 2009; pp. 290–293. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.