Event Detection in Optical Signals via Domain Adaptation

Event Detection in Optical Signals via Domain Adaptation

Antonino Maria Rizzo[1], Luca Magri[1], Pietro Invernizzi[2], Enrico Sozio[2], Gabriele Aquaro[2], Stefano Binetti[2], Cesare Alippi[1,3], Giacomo Boracchi[1]

[1]Politecnico di Milano, [2]Cisco Photonics, [3]Università della Svizzera Italiana

***Abstract*—Data-driven models trained in an end-to-end manner can reliably detect events within optical signals. Unfortunately, event detection models poorly generalize when monitoring signals collected from devices with different acquisition procedures. We overcome this limitation by presenting a novel domain adaptation solution for event detection networks that enables inference across multiple types of devices. Rather than training a black-box detection network, we decouple event localization and classification tasks. Localization is performed by the Interval Proposal Algorithm (IPA), which leverages signal processing techniques to localize candidate events and derive *context* features. These events are then standardized and fed to a feature extractor to obtain *morphological* features. By combining domain-specific context features with domain-invariant morphological features, the classifier achieves good generalization capabilities through different domains. Our method can successfully detect events in OTDR traces achieving a mAP@0.5 of 75.33% on traces from the source domain and generalizing well (mAP@0.5 of 69.27%) on traces from the target domain, despite being trained solely from the source domain.**

***Index Terms*—otdr, domain adaptation, event-detection.**

I. Introduction

The success of machine learning models is closely tied to the availability of large amounts of data. Unfortunately, in some applications, acquiring new data is costly, and data scarcity undermines the application of such techniques.

A notable example comes from the monitoring of fiber optics, where optical signals are analyzed to guarantee the quality of transmissions. In particular, Optical Time Domain Reflectometer (OTDR) devices [11] are used to monitor the status of fiber links. These devices inject a short laser pulse at one end of a fiber link and measure the backscattered and reflected light. This process results in an *OTDR trace*, i.e., a graphical representation of the optical power as a function of the distance along the fiber.

Automatic event-detection networks simultaneously localize and classify *event signatures* in OTDR traces to identify the presence of specific impairments along the fiber, such as a bad connector or a broken fiber, which require triggering corrective actions. Specifically, [12] solves the event-detection problem by leveraging a 1D Faster R-CNN. Despite its accuracy, this approach is limited to operating on OTDR traces acquired with the very same type of device used for collecting the training set, and falls short when applied to traces acquired by other types of devices. In fact, the acquisition technology heavily influences the event shapes.

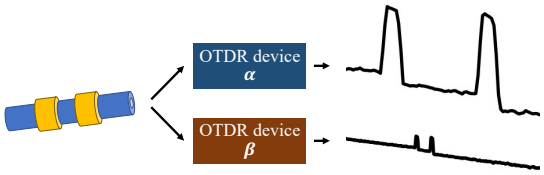


Fig. 1: On the left-hand side, a fiber link with two connectors. The PASS-THROUGH event signatures on the right-hand side generated by devices of type α and β show substantial differences both in terms of the number of samples and power intensity.

An example of this issue is depicted in Fig. 1 representing two types of OTDR devices, α and β, connected to the same fiber link. The resulting traces present two bumps corresponding to two physical connectors along the fiber link (depicted in yellow). Despite representing the same fiber span, the two OTDR traces are extremely different in magnitude and scaling. In particular, the events (two bumps) from β appear smaller and much shrank, as a result of the lower sampling frequency of device β in that region. These differences are caused by the physical configuration of the OTDR devices acquiring the signal. Therefore, an event detection network trained in an end-to-end manner on OTDR traces acquired by device α (source domain) could not extract relevant features to detect events acquired by device β (target domain).

The lack of feature invariance across devices hinders the portability of the event detection network from α to β. As a matter of fact, these models must be trained using data specific for each type of device, limiting their applicability and requiring time-consuming data preparation.

We address this limitation as a Domain Adaptation (DA) problem. First, we show that event-detection networks relying on end-to-end training have little flexibility for adaptation without training data from the target domain. Therefore, we design a novel event-detection network that is easier to adapt when the device changes.

Inspired by the R-CNN architecture [7], we rely on expert-driven algorithms to localize the events leveraging their morphology. In particular, our *Interval Proposal Algorithm* (IPA) exploits signal processing techniques and robust fitting methods to obtain a sequence of candidate events with their *context*

features, representing domain-specific information such as their power intensity and position along the OTDR trace.

Then, to perform event classification, we standardize the events, and train a feature extractor f_θ to obtain *morphological* features that focus exclusively on the shape of the events regardless of their context. These domain-invariant morphological features are concatenated with the domain-specific context features computed by the IPA and fed to the classifier K to predict the event type. The combination of context and morphological features enables accurate prediction and allows our method to be successfully used across devices. On the contrary, the 1D Faster R-CNN, like other black-box models trained end-to-end, disregards expert-driven knowledge, and is unable to work across different devices.

Quantitative experiments on real-world traces obtained from two types of OTDR devices by *Cisco*, termed Mystique (device α) and Sirius (device β), demonstrate that our method attains a mAP@0.5 of 75.33% on traces from device α, where the 1D Faster R-CNN reaches 69.27%. When we employ our method on traces from device β, we get 64.93% without any form of fine-tuning. In contrast, the 1D Faster R-CNN is unable to detect events. These results confirm that our framework can be successfully employed to yield an OTDR event-detection network that is portable across different devices.

A. Related Work

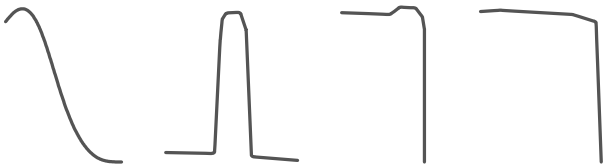
Domain Adaptation (DA) is a well-studied problem and several methods have been proposed. A stream of research learns domain-invariant feature representations. Specifically, [6] adds a domain classifier to the feature extractor using a gradient reversal layer. This ensures that the feature distributions of the two domains are made similar. Another line of research performs domain mapping using a Generative Adversarial Network (GAN). For example, [1] adapts source-domain data to appear as if drawn from the target domain. A context similar to ours is discussed in [3], where domain adaptation is performed on ECG heartbeats. All these approaches require target-domain data at training time, even unlabelled. In contrast, we assume no data from the target domain is available at training time.

A different strategy proposed in [9] involves replacing the mean and standard deviation of the Batch Normalization layers by recalculating them during inference. This method however requires modifying the network during inference, which is not always a viable option in an industrial scenario and for devices that are meant to operate stand-alone. In contrast, we only apply a pre-processing of the input trace by leveraging expert knowledge, and a standardization of candidate events before classification.

Inspired by [2], where DA is performed by decoupling the image content from its style, we separate the morphology of the OTDR events from the context information to ease their recognition and improve the detection performance.

II. Problem Formulation

An OTDR trace is represented as a sequence of n samples $T = \{(x_1, p_1), (x_2, p_2), ..., (x_n, p_n)\}$, where $p_i \in \mathbb{R}$ is the reflection loss (in dBm) measured at distance $x_i \in \mathbb{R}$ (in km) from the fiber's beginning. A trace T may contain multiple events, each one described by a triplet $e = (y, x^s, x^e)$, where x^s and x^e are the start and end points of the portion of the trace containing the event of class y. We assume to have a fully labeled training set $\mathrm{TR}^{\mathcal{S}} = \{(T_j, E_j)\,,\, j = 1, \ldots, N\}$, containing N traces from the source domain $\mathcal{D}^{\mathcal{S}}$, where $E_j = \{e_i\}_{i=1}^{M_j}$ is the set of events in trace T_j. We consider two OTDR devices, α and β, each implementing different sampling and post-processing techniques, resulting in traces belonging to source domain $\mathcal{D}^{\mathcal{S}}$ and target domain $\mathcal{D}^{\mathcal{T}}$, respectively. In particular, events collected from domain $\mathcal{D}^{\mathcal{T}}$ might have different power intensities and widths, and an event-detection network trained on traces from $\mathcal{D}^{\mathcal{S}}$ might be unable to generalize to events from $\mathcal{D}^{\mathcal{T}}$. Due to limited data availability, fine-tuning a model specifically for the target domain $\mathcal{D}^{\mathcal{T}}$ is not feasible.



(a) Face-Plate (b) Pass-Through (c) Fiber-Cut (d) Fiber-End

Fig. 2: Instances of OTDR events in the training set.

Our objective is to design a method trained exclusively on domain $\mathcal{D}^{\mathcal{S}}$ that can automatically detect all events in a trace T, regardless of whether they originate from $\mathcal{D}^{\mathcal{S}}$ or $\mathcal{D}^{\mathcal{T}}$. We formulate this as a *Domain Adaptation* (DA) problem, meaning we aim to compensate for the shift between training and test data distribution.

In our study, we consider four types of basic events, which are reported in Fig. 2:

- FACE-PLATE denotes the fiber start and determines the location at which the OTDR tool injects its pulses.
- PASS-THROUGH denotes the mechanical conjunction of two fiber links.
- FIBER-END denotes the endpoint of the fiber link.
- FIBER-CUT denotes the abrupt termination of the fiber induced by a cut.

As customary in detection networks, we include also the NO-EVENT label to discard regions that do not contain an event.

III. Proposed Method

Our proposed method (depicted in Figure 3) decouples event localization and classification, enabling the standardization of candidate events to learn domain-invariant features. Furthermore, we combine context information about the event's location within the trace to improve classification accuracy.

The input OTDR trace is first fed to the Interval Proposal Algorithm (IPA) to obtain a collection of candidate events with their corresponding context features (Sec. III-A). Next, we standardize (Sec. III-B) the events that are then passed through the Feature Extractor f_θ to extract morphological features and concatenated with the context features. Finally, we input the

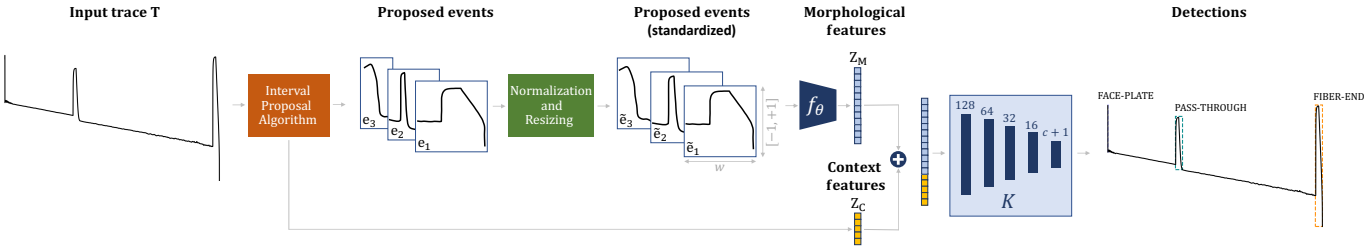


Fig. 3: Overview of the proposed architecture.

concatenated feature vector into the classifier K to predict the corresponding event labels. The network architecture and training phases are described in Sec. III-C, while Sec. III-D contains the implementation details.

A. Interval Proposal Algorithm

The Interval Proposal Algorithm analyzes the trace T to identify tentative events e_i, extracting variable-size intervals (x_i^s, x_i^e). Specifically, the Interval Proposal Algorithm (IPA) performs two main steps.

Peak detection. The IPA detects a set of peak locations $x_i \in \mathcal{P}$ that will be the candidate events. To achieve this, we detect the local maxima $\mathcal{M}$ in the detrended trace, obtained by fitting a linear trend using MSAC [4]. To be more robust to noise and avoid detecting too low peaks in dense regions, this stage first identifies high peaks $\mathcal{P}^h$, while the second stage identifies low peaks $\mathcal{P}^l$. High peaks $\mathcal{P}^h = \{x_i \in \mathcal{M} \colon \rho(x_i) \geq H\}$ are detected as local maxima with prominence $\rho(x_i)$ larger or equal than the device-specific threshold H, where the prominence $\rho(\cdot)$ measures how much the peak stands out with respect to the neighboring peaks. Conversely, low peaks $\mathcal{P}^l = \{x_i \in \mathcal{M} \colon L \leq \rho(x_i) < H \text{ and } \forall x_k \neq x_i, |x_i - x_k| > d\}$ are those local maxima with prominence between L and H, and d samples apart from each other. The set of candidate peaks is obtained as $\mathcal{P} = \mathcal{P}^h \cup \mathcal{P}^l$.

Interval estimation. As depicted in Fig 4, for each detected peak $x_i \in \mathcal{P}$, the IPA crops two windows of size w to the left and right of the peak, i.e., $W_i^l = \{x_{i-1} - w, \ldots, x_{i-1}\}$ and $W_i^r = \{x_{i+1}, \ldots, x_{i+1} + w\}$. When another peak falls inside one of the windows, the cropping is limited to its position. Each window W_i^l and W_i^r is used to fit a trend line, thus obtaining two polynomials, ν^l and ν^r, describing the regions around the event. The left limit x_i^s of the interval is taken as the right-most inlier[1] of ν^l, and the right limit x_i^e of the interval is taken as the left-most inlier of ν^r. A margin can be added to better accommodate the tails of the event.

For each candidate event window (x_i^s, x_i^e), the IPA extracts the context features z_C, i.e., the minimum, average, and maximum power within the localized event window, as well as the location of the peak x_i expressed in km. Moreover, considering the relative position of events within the trace can be beneficial for classification, as certain events commonly occur at specific positions. Therefore, inspired by the Transformer architecture [13], we leverage also the cosine-based *positional encoding*, which is included into z_C. Finally, the IPA returns the tentative event-window intervals (x_i^s, x_i^e) and the context features.

[1]An inlier is a sample that lies closer to the model than a threshold ε.

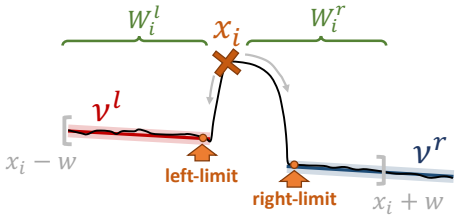


Fig. 4: Interval estimation.

B. Event Standardization

Event standardization is key to obtain features that are invariant across devices and hence focusing on the morphology of the events. To addresses the differences in event width from the different domains, we resize the variable-sized candidate events e_i localized by the IPA to a fixed-size window of w samples. Furthermore, to address the variation in power intensity observed across different types of devices, each event is normalized within the $[-1, 1]$ range. The standardized events are denoted with $\tilde{e}_i$ and are then fed to the feature extractor to obtain the morphological features. This stage is crucial for enabling the model to generalize effectively across domains.

C. Network Architecture and Training

Our feature extractor, denoted by f_θ, draws inspiration from the ResNet [8] architecture. It includes 28 convolutional layers, a receptive field of 446, and only 372 352 parameters. The model operates on normalized event windows of size w. To improve generalization on events with significant width variability, we leverage resampling as a form of data augmentation. Specifically, we interpolate the event samples to generate windows up to 50% longer or shorter. When an event is down-sampled, padding is applied to preserve the window-size. Conversely, when an event is up-sampled, we crop it.

Training is performed exclusively on data from the source domain $\mathcal{D}^S$. It is divided into two phases. In the first phase, a feature extractor f_θ is concatenated to an auxiliary linear classifier K_0 for training. The classifier K_0 allows the feature extractor to learn useful representations of the input data and is discarded at the end of this phase. The feature extractor is said to extract *morphological features* z_M because it is trained on standardized windows that focus on the shape of the events.

In the second phase, depicted in Fig 3, the feature extractor is frozen, and the classifier K is trained on top of it. The classifier K takes as input both the morphological features z_M extracted by the feature extractor f_θ and the context features z_C extracted by the IPA. By combining morphological and context features, classifier K leverages both the intrinsic event shape and context-related information to produce a more informative feature representation, ultimately leading to improved performance of the classifier.

D. Implementation Details

A preprocessing stage is performed on the entire trace before event detection, which involves clipping the portions of the trace where the signal drops below a certain threshold, as regions with very low power contain only noise. In addition, padding is applied to both the left and right borders to center events at the edges of the trace. As regard the prominence values used in peak detection, we set $L = 0.3$ for traces in D^S, while $L = 0.1$ for traces in D^T where events appear much smaller. Due to their physical interpretation, it is easy to tune the higher prominence value H and the inlier threshold ε for MSAC, which have been set to $H = 2.0$ and $\varepsilon = 0.1\text{dB}$, respectively.

IV. Experiments

To validate the performance of our method, we perform two experiments. First, we assess the quality of the Interval Proposal Algorithm. Then, we evaluate the entire detection pipeline against the 1D Faster R-CNN [12], with receptive field 278 and trained with a learning rate of 10^{-3}.

A. Dataset

The dataset of OTDR traces used to validate our method has been acquired in Cisco facilities over long spans of optical fiber, where optical events have been generated by different devices connected at different locations along the fiber link. Overall, 915 traces have been collected from the Mystique device (α), which constitutes the source domain D^S, and 31 traces from the Sirius device (β), which constitutes target domain D^T. Notice that only the traces from Mystique are used for training.

B. Figures of Merit

We assess detection performance by the mean average precision. In particular, we consider mAP@0.5, which was introduced in the *PASCAL VOC* challenge [5], and the mAP@$[.5 : .05 : .95]$ which was introduced in the *COCO* challenge [10] that evaluate mAP at 10 different Intersections-over-Union (IoU) thresholds.

In addition, to better characterize the performance of the methods, we resort to standard metrics used for classification: accuracy, precision, recall, F_1 Score, False Positive Rate (FPR), and False Negative Rate (FNR). Specifically, we consider a detected event a *True Positive* when its IoU with the ground-truth is greater than 0.5, and the label is correctly estimated. *False Negatives* refer to ground-truth events that were not detected, while *False Positives* represent detected events that do not correspond to any ground-truth event (i.e., NO-EVENTs). In addition to the above metrics, we use confusion matrices to better inspect the sources of errors for the different models. To do so, we classify all the localized events. The IPA is evaluated in terms of precision and recall for the retrieved peaks.

	Mystique (Domain $\mathcal{D}^S$)		*Sirius* (Domain $\mathcal{D}^T$)	
	Faster	Our	Faster	Our
Accuracy	92.45	**93.78**	57.91	**93.18**
Precision	77.91	**83.89**	13.77	**79.52**
Recall	**89.93**	88.32	14.67	**88.59**
F_1 Score	83.49	**86.05**	14.20	**83.81**
False Positive Rate	6.86	**4.69**	28.61	**5.67**
False Negative Rate	**10.06**	11.67	85.33	**11.40**
mAP@0.5	69.27	**75.33**	3.12	**64.93**
mAP@$[.5 : .05 : .95]$	38.86	**74.46**	0.83	**64.36**

TABLE I: Performance computed from Mystique and Sirius traces. Mystique values are computed by Cross-validation.

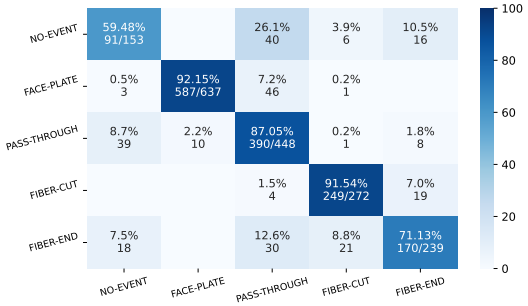


Fig. 5: Confusion matrix computed from Mystique traces computed by Cross-validation. Rows denote the ground-truth, while columns denote the predictions.

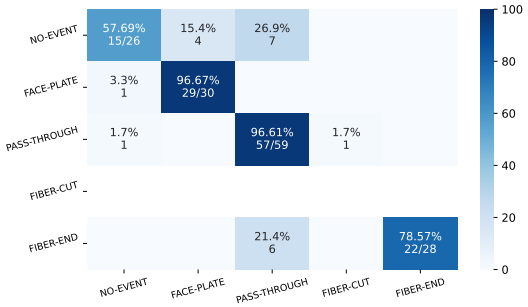


Fig. 6: Confusion matrix computed from Sirius traces. Rows denote the ground-truth, while columns denote the predictions.

C. Discussion

1) Interval Proposal Algorithm: The IPA employed on Mystique traces attains 96.45% recall and 67.1% precision, while on Sirius traces, it reaches 31.19% precision and 97.00% recall. Despite the low precision, the high recall is crucial as it indicates that almost all events are fed to the network, which will possibly classify them as NO-EVENTs.

2) Event-detection performance: The performance of our method is reported in Table I, where it is compared against

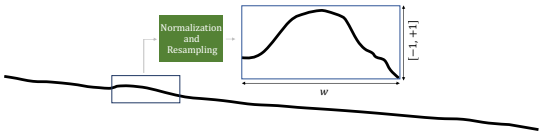


Fig. 7: This negligible bump results in a PASS-THROUGH after the normalization.

the Faster R-CNN on traces from the source domain D^S (Mystique device) and target domain D^T (Sirius device). In the source domain D^S, the first two columns show that our method performs comparably to 1D Faster R-CNN. However, it is important to note that our method achieves higher mAP. This is due to our expert-driven localization and interval estimation that better fit the events. This is supported by the fact that increasing the IoU threshold does not decrease mAP.

When applied to the Sirius traces, the advantages of our solution become apparent. The 1D Faster R-CNN shows poor performance as it fails to detect most events. This is because the events in the Sirius domain have different shapes compared to domain D^S, which is indicated by the high false negative rate (FNR) and low mAP. In contrast, our method can operate successfully without being affected by the domain change, as can be appreciated by its performance that is in line with the ones on the D^S source domain.

3) Confusion matrices: We can better inspect the errors of our method by inspecting the confusion matrices in Fig. 5 and Fig. 6. These confusion matrices reveal that the primary source of errors is the misclassification of NO-EVENTs as PASS-THROUGHs, which is reflected by the false positive rate of 11.4%. This happens when negligible bumps in the trace can be mistakenly classified as a PASS-THROUGH. This is due to the normalization stage makes noisy NO-EVENTs similar to PASS-THROUGHs, as depicted in Fig. 7. Qualitative experiments performed on Mystique and Sirius traces confirm that our method is accurate and capable of detecting optical events in the trace. An example is depicted in Fig. 8 and Fig. 9, where all the events have been correctly detected.

V. Conclusions and Future Work

We present a novel method to address Domain Adaptation in OTDR traces. Unlike traditional data-driven models, trained end-to-end and working as black-box methods, our approach leverages domain knowledge to analyze traces from different domains enabling the same model to perform event detection across different types of devices.

As future work, we plan to extend this framework to monitor fiber spans that include a broader range of physical events, e.g., presence of amplifiers and fiber bends and knots.

VI. Acknowledgements

We thank *Edoardo Peretti* (Politecnico di Milano) for his support in designing the Interval Proposal Algorithm. Furthermore, we gratefully acknowledge *NVIDIA Corporation* for the GPUs donated within the Academic and Applied Research Program to *Giacomo Boracchi* and *Luca Magri*.

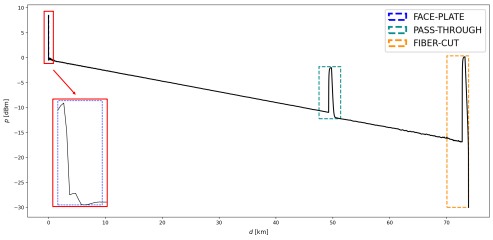


Fig. 8: Qualitative result on a Mystique trace.

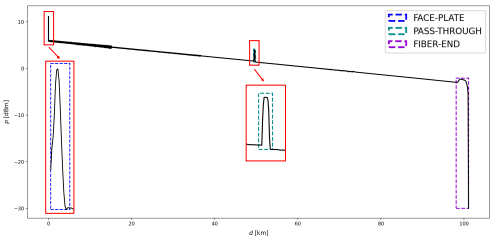


Fig. 9: Qualitative result on a Sirius trace.

References

[1] Konstantinos Bousmalis, Nathan Silberman, David Dohan, D. Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2016.

[2] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems 29*, 08 2016.

[3] Diego Carrera, Beatrice Rossi, Pasqualina Fragneto, and Giacomo Boracchi. Domain adaptation for online ecg monitoring. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 775–780, 2017.

[4] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In Bernd Michaelis and Gerald Krell, editors, *Pattern Recognition*, pages 236–243, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[5] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2007.

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[11] Barnoski M. K., Rourke M. D., Jensen S. M., and Melville R. T. Optical time domain reflectometer. *Applied Optics*, 16(9):2375–2379, 1977.

[12] Antonino Maria Rizzo, Luca Magri, Davide Rutigliano, Pietro Invernizzi, Enrico Sozio, Cesare Alippi, Stefano Binetti, and Giacomo Boracchi. Known and unknown event detection in OTDR traces by deep learning networks. *Neural Computing and Applications*, 2022.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.