UNIVERSITÀ DEGLI STUDI DI CATANIA

Dipartimento di Matematica e Informatica

Dottorato di Ricerca in Matematica e Informatica XXXVI

Ciclo

Giovanni Pasqualino

Unsupervised Domain Adaptation for Object Detection and Action Recognition

Tesi di Dottorato di Ricerca

Supervisor: Prof. Sebastiano Battiato

Anno Accademico 2022 - 2023

Abstract

This thesis addresses the problem of unsupervised domain adaptation (UDA) for the object detection and action recognition. UDA is a machine learning technique that aims to minimize the domain shift between a source domain (with labeled data) and a target domain (with unlabeled data). The main goal is to develop a model capable of adapting to different scenarios, eliminating the need for resource-intensive data labeling and retraining, while maximizing the performance on the target domain.

We investigate the UDA problem and explore its applications in object detection and action recognition. For object detection, we introduce two datasets and propose novel architectures based on adversarial learning, self-training, and image-to-image translation to learn domain-invariant representations that can generalize across single or multiple target domains.

For action recognition, we analyze the ability of state-of-the-art methods to generalize across first-person and third-person actions, identifying the most efficient techniques for detecting actions from both point of view.

We conclude by discussing the limitations and future directions of UDA research in computer vision tasks. We have publicly released the code of the proposed algorithms and the datasets, facilitating further research in this area.

Acknowledgements

To my family.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions 4

1.3 Outline 6

2 Background **7**

2.1 Unsupervised Domain Adaptation 7

2.2 Unsupervised Domain Adaptation for Object Detection 13

2.2.1 Object Detection 14

2.2.2 State-of-the-Art UDA Methods for Object Detection . 15

2.3 Unsupervised Domain Adaptation for Action Recognition . . . 18

2.3.1 Action Recognition 18

2.3.2 State-of-the-Art UDA Methods for Action Recognition 19

3 UDA for Object Detection **22**

3.1 Introduction 22

3.2 Methods 23

3.2.1 Baseline approaches without adaptation 24

3.2.2 Domain adaptation through image-to-image translation 25

3.2.3 Domain adaptation through feature alignment 25

3.2.4 Proposed Method: DA-RetinaNet 26
3.2.5 Domain adaptation through feature alignment and image to image translation 27
3.3 Experimental Settings and Results 28
3.3.1 Dataset 28
3.3.2 Experimental Settings 30
3.3.3 Baseline Results 31
3.3.4 Image-to-Image translation Results 32
3.3.5 Feature Alignment and Image-to-Image translation Results 33
3.3.6 Ablation Study 35
3.3.7 Summary table and Qualitative Results 36
3.3.8 Analysis of Computational Resources 39
3.3.9 Results on Cityscapes Dataset 40
3.4 Conclusion 41

4 Multi-Target UDA for Object Detection 43
4.1 Introduction 43
4.2 Dataset 45
4.3 Methods 48
4.3.1 Baselines without domain adaptation 48
4.3.2 Domain adaptation based on feature alignment 49
4.3.3 Domain adaptation through feature alignment and image to image translation 49
4.3.4 Proposed Method 49
4.3.4.1 Image to Image Translation 50
4.3.4.2 Feature Alignment 51
4.3.4.3 Self-Training 53

4.3.5 Experimental Settings 54
4.4 Results 55
4.4.1 Feature Alignment Results 55
4.4.2 Feature Alignment and Image to Image translation Results 57
4.4.3 Ablation Study 58
4.4.4 Comparison between MDA-RetinaNet and DA-RetinaNet 61
4.4.5 Qualitative Results 62
4.4.6 Results on Cityscapes Dataset 62
4.5 Conclusion 64

5 UDA for Action Recognition 66
5.1 Introduction 66
5.2 Methods 67
5.3 Experimental Settings and Results 68
5.3.1 Datasets 69
5.3.2 Baseline Results 69
5.3.3 Domain Adaptation Results 70
5.4 Conclusion 74

6 Conclusion 76

Appendix 77

A GAN against Malicious Tampering 78
A.1 Introduction 78
A.2 Related Work 80
A.2.1 GAN Applications in Medical Imaging 80
A.2.2 Adversarial Attacks 80

A.2.3 Image Manipulation Prevention 81
A.3 Proposed Method 81
A.3.1 Method Overview 81
A.4 Results 82
A.4.1 Experimental Setup 83
A.4.2 Qualitative Results 83
A.4.3 Quantitative Results 83
A.5 Conclusion 86

Bibliography **88**

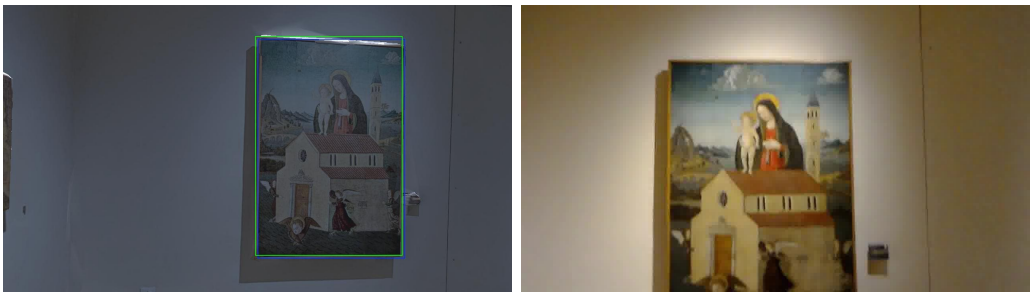
Chapter 1

Introduction

1.1 Motivation

Object detection and action recognition are fundamental tasks in computer vision that play a crucial role in numerous applications, ranging from video surveillance [1] and autonomous driving [2] to augmented reality [3] and human-computer interaction [4]. With the increasing computational capabilities of modern devices, including wearable devices and smartphones equipped with powerful processors, the integration of object detection and action recognition algorithms into various applications has become more accessible. This integration empowers users with enhanced features and enables the extraction of valuable insights from visual data [5], [6], [7], [8].

However, the progress made in object detection and action recognition models is often hindered by the expensive and time-consuming nature of manual data labeling. The labor-intensive process of annotating datasets with bounding boxes for object detection or labeling frames for action recognition demands substantial resources in terms of both financial costs and time investments. To address this challenge, researchers have explored the



(a) Training on synthetic images and testing on synthetic (left) and real (right) images. The algorithm fails to detect the artwork in the real image, highlighting the challenge of domain shift.

(b) Training on sunny images and testing on sunny (left) and foggy (right) images. The detection results show incorrect bounding box predictions, pointing out the impact of environmental variations on performance.

Figure 1.1: Qualitative examples of Faster RCNN. The blue box represents the ground truth, while the red and green boxes indicate predictions. The red box denotes wrong detection (object localization or classification), whereas the green box represents correct detection.

use of tools that automatically generate large amounts of labeled synthetic images [9], [10], [11]. These tools have the potential to significantly reduce labeling costs. However, models trained on synthetic data often face a performance gap when applied to real-world images (Figure 1.1a). This performance degradation can be attributed to the domain shift between the synthetic and real data, where models fail to generalize well due to the distribution disparities in real-world scenarios. This is primarily because super-

vised learning approaches typically assume that the test data (target domain) and the labeled training data (source domain) adhere to the same underlying distribution. In reality, the test data often exhibits a distribution shift from the training data, even if the images are not synthetic. The domain shift occurs due to variations in operating conditions and environmental factors (Figure 1.1b), posing a significant challenge for many deep learning methods. The distribution difference between the source and target domains must be addressed to achieve optimal performance in real-world scenarios.

To tackle these challenges, researchers have turned to domain adaptation methods, a subset of transfer learning techniques [12] that offer a promising solution. These approaches facilitate the transfer of knowledge from a source domain to a closely related target domain. The key objective is aligning the source and target distributions, thereby mitigating the need for extensive labeled data in the target domain. Within this realm, Unsupervised Domain Adaptation (UDA) stands out as a specific subtype. UDA involves the adaptation of a model from a labeled source domain to an unlabeled target domain [13], [14].

This thesis aims to investigate the field of unsupervised domain adaptation for object detection and action recognition tasks. The primary research objective is to address the limitations of manual data labeling, explore techniques to bridge the performance gap between synthetic and real data, and develop novel algorithms and methodologies for UDA in these tasks.

For the object detection task, we created the UDA-CH (Unsupervised Domain Adaptation on Cultural Heritage) dataset, which consists of 16 artworks. This dataset includes both synthetic images labeled automatically using a tool and real images captured with Hololens and manually labeled. State-of-the-art methods have been evaluated on this dataset, and as result,

we introduce our algorithm named DA-RetinaNet, which achieves superior performance on the UDA-CH and Cityscapes datasets. Furthermore, we have addressed the challenge of unsupervised domain adaptation in the presence of multiple target domains by extending the UDA-CH dataset introducing manually labeled images acquired by a GoPro, and developing the (ST)MDA-RetinaNet algorithm.
For the action recognition task, we have studied state-of-the-art algorithms and their behavior for both first-person and third-person actions. We conducted a preliminary analysis emphasizing methodologies that yield better recognition results for both types, aiming to enhance the performance and applicability of action recognition algorithms in different scenarios.

1.2 Contributions

In summary, the main contributions of this thesis are:

1. We introduce the UDA-CH [1] dataset and its extension [2] to study, respectively, the unsupervised domain adaptation and the unsupervised multi-target domain adaptation for object detection.

2. We propose DA-RetinaNet [3] and (ST)MDA-RetinaNet [4] to address the UDA problem for object detection in a single and multi-target environment.

3. We conduct a preliminary study of UDA for action recognition focusing on the methodologies that perform best for first-person and third-

[1] https://iplab.dmi.unict.it/EGO-CH-OBJ-UDA/
[2] https://iplab.dmi.unict.it/OBJ-MDA/
[3] https://github.com/fpv-iplab/DA-RetinaNet
[4] https://github.com/fpv-iplab/STMDA-RetinaNet

person views. We analyze different approaches and presents insights into their strengths and weaknesses.

This thesis draws on the research conducted and published/in progress during the Ph.D., which includes the following works:

- Journal paper:
  G. Pasqualino, A. Furnari, G.M. Farinella, "A Multi Camera Unsupervised Domain Adaptation Pipeline for Object Detection in Cultural Sites through Adversarial Learning and Self-Training", Computer Vision and Image Understanding 2022 (CVIU) 2022

- Journal paper:
  G. Pasqualino, A. Furnari, G. Signorello, G.M. Farinella, "An Unsupervised Domain Adaptation Scheme for Single-Stage Artwork Recognition in Cultural Sites", Image and Vision Computing (IMAVIS), 2021

- Conference paper:
  G. Pasqualino, A. Furnari, G.M. Farinella, "Unsupervised Multi-camera Domain Adaptation for Object Detection in Cultural Sites", International Conference on Image Analysis and Processing (ICIAP) 2022

- Conference paper:
  G. Pasqualino, A. Furnari, G. M. Farinella, "Unsupervised Domain Adaptation for Object Detection in Cultural Sites", International Conference on Pattern Recognition (ICPR), 2020

- In progress:
  G. Pasqualino, "Unsupervised Domain Adaptation for Action Recognition: A Preliminary Study from First and Third-Person Views"

1.3 Outline

The remainder of the thesis is organized as follows. In Chapter 2 we present an overview of the foundational concepts and techniques that form the background for the research. We cover the fundamental principles of Unsupervised Domain Adaptation, Object Detection and Action Recognition providing insights into the existing literature and state-of-the-art approaches. In Chapter 3 we delve into the study of unsupervised domain adaptation techniques for object detection tasks. We explore various methods aimed to aligning the distributions of source and target domains and introduce the UDA-CH dataset and the DA-RetinaNet algorithm. In Chapter 4, building upon the concepts of unsupervised domain adaptation, we focus on addressing the UDA problem involving multiple target domains presenting the OBJ-MDA multi-camera dataset and the (ST)MDA-RetinaNet algorithm. In Chapter 5 we investigate unsupervised domain adaptation techniques for action recognition tasks. We analyze methodologies for first-person and third-person action recognition, highlighting approaches that yield better recognition results for both types of actions. Finally, in Chapter 6 we summarize the main findings and contributions of the thesis. We discuss the limitations of the proposed methods and identify potential areas for future research.

Chapter 2

Background

This chapter provides a comprehensive background on the fundamental concepts and techniques that form the basis of the research conducted in this thesis. The aim is to establish a solid understanding of the key elements related to unsupervised domain adaptation techniques and their application for object detection and action recognition. By exploring the existing literature and state-of-the-art approaches in these fields, we lay the groundwork for the advanced methodologies and approaches discussed in later chapters.

2.1 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation is a technique that aims to improve the performance of a model in the presence of a domain shift between the training (source) and test (target) data. UDA methods leverage additional unlabeled data from the target domain to reduce the domain shift and improve the model's performance on the target test data.

Formally, the UDA problem can be defined as follows: Let $S = \{(x_s^n, y_s^n)\}_{n=1}^{N_s}$ be the set of N_s labeled images from the source domain D_s, where x_s^n indi-

cates the n^{th} source image and y_s^n represents its corresponding annotation. Let $T = \{x_t^n\}_{n=1}^{N_t}$ denote the set of N_t unlabeled images from the target domain D_t, where x_t^n represents the n^{th} target image. The goal is to learn a function $f : X \to Y$ that can accurately predict the labels y_t for target domain samples x_t using the labeled source domain data D_s and the unlabeled target domain data D_t.

During the training procedure, the objective is to minimize the domain discrepancy between the source and target domains while maximizing the classification accuracy on the source domain. The loss function for UDA can be formulated as follows:

$$\min_f H(f, D_s) + \lambda \cdot L(f, D_s, D_t) \tag{2.1}$$

$H(f, D_s)$ denotes the source domain classification loss term, which measures the accuracy on the source domain; $L(f, D_s, D_t)$ represents the domain discrepancy loss term, which capture the difference between the source and target domains; λ is the trade-off parameter that controls the adaptation loss and the classification. The specific form of the loss functions and the adaptation mechanisms may vary depending on the UDA method and the specific problem being addressed. For action recognition tasks, the focus is on minimizing the standard Cross Entropy loss to accurately classify actions while for the object detection, the loss functions are designed to handle both classification and regression tasks associated with object bounding boxes.

At test time, the performance of the adapted model is evaluated on the target test set to assess its ability to generalize to unseen target domain samples. The structure and pipeline of the training and test processes for a standard algorithm and a UDA algorithm are compared in Figure 2.1. In an unsupervised domain adaptation setting, the task-specific models are modified to use a source labeled dataset and a target unlabeled dataset during

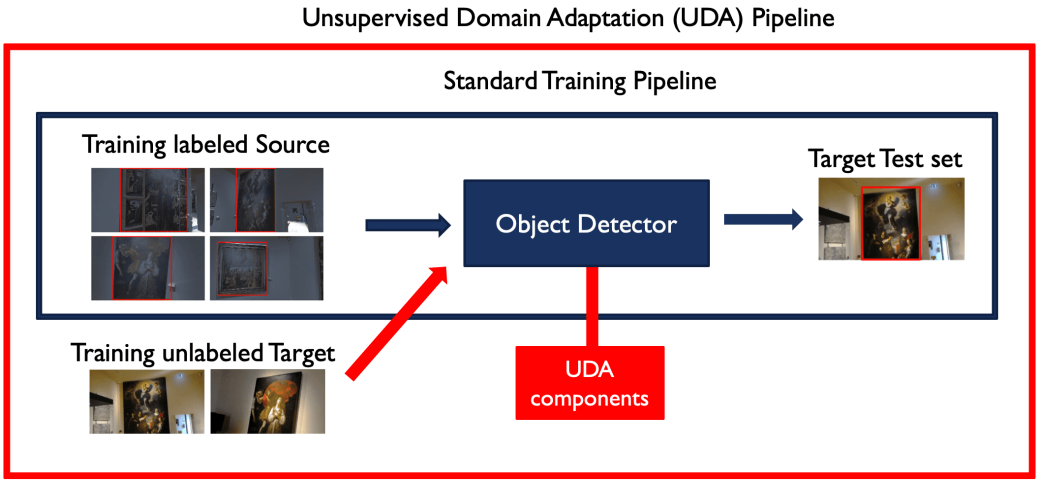


Figure 2.1: Comparison between the standard training and test pipeline and the UDA pipeline for object detection. The blue box shows the standard training procedure, where the object detector is trained on a source dataset and tested on the target test set. The red box shows the UDA pipeline, which includes the use of an unlabeled training target and the UDA components to reduce the domain gap.

training. Additionally, one or multiple domain adaptation components are integrated into the algorithms to reduce the domain shift.

Thanks to the ever-growing interest of the scientific community in this problem, several techniques have been developed. The authors of [15], [16], [17] proposed to minimize divergence quantities that can be measured between source and target distributions. Minimizing these quantities allows the model to extract features that are invariant with respect to the two domain distributions. In particular, the authors of [17] proposed a method that aims to minimize the intra-class discrepancy and maximize the inter-class discrepancy. The authors of [15] exploited the MMD metric [18] in a CNN to reduce the distribution mismatch. The model consists of two branches, one for each domain, whose weights are not shared but lead equally to extract similar

features from both domains minimizing the following loss:

$$L = L_s + L_t + L_w + L_{MMD} \tag{2.2}$$

L_s and L_t are the standard classification losses, L_w and L_{MMD} represent two regularizers that allow to extract invariant features from the two distributions. The first represents the loss between the corresponding levels of the two flows, the second encodes the MMD metric defined as:

$$MMD^2(\{f_j^s\}, \{f_j^t\}) = \left\| \sum_{i=1}^{N^s} \frac{\phi(f_i^s)}{N^s} - \sum_{j=1}^{N^t} \frac{\phi(f_i^t)}{N^t} \right\|^2 \tag{2.3}$$

Where $\phi(\cdot)$ denotes feature mapping in a RKHS [19], f_i^s and f_i^t represent respectively the features extracted from the last layers from the source and target streams. The authors of [16] used the CORAL metrics [20] inside a CNN to align the covariances of the source and target distributions by minimizing:

$$L = L_{CLASS} + \sum_{i=1}^{t} \lambda_i L_{CORAL} \tag{2.4}$$

where L_{CLASS} is the classification loss, $L_{CORAL} = \frac{1}{4d^2}||C_S - C_T||_F^2$ is the loss defined as the distance between the covariance of the source (C_S) and target domain (C_T), t is the number of layers with CORAL loss, λ is the weight assigned to each of them.

Other works used the adversarial learning paradigm to align the distributions of the features extracted by the models of the source and target domains. The authors of [13] introduce a gradient reversal layer into a standard CNN to align the distributions of source and target features using adversarial learning. Specifically, the model they propose includes two components. The first one processes the input samples to solve the supervised task (e.g., classification). The second one is devoted to discriminate if the features extracted

from the input sample belong to the source or target domain. The network is trained to minimize the supervised loss of the first component and the discriminator loss of the second one. The gradient reversal layer, which implements a minmax game similar to the one described in [21], is used to invert the gradients of the discriminator multiplying them by $-\lambda$ when they are used to update the parameters of the first component. In this way, during the backpropagation optimization pass the weights are update as follow:

$$\theta_f \leftarrow \theta_f - \mu(\frac{\partial L_y^i}{\partial \theta_f} - \lambda \frac{\partial L_d^i}{\partial \theta_f}) \tag{2.5}$$

$$\theta_f \leftarrow \theta_f - \mu \frac{\partial L_y^i}{\partial \theta_f} + \mu\lambda \frac{\partial L_d^i}{\partial \theta_f} \tag{2.6}$$

where μ is the learning rate, L_y is the classification loss and L_d is the domain classification loss. The authors of [14] propose a method based on two stages: in the first stage a CNN is trained on the source dataset. In the second stage the weights of the CNN are adapted to extract domain-invariant features. During the test phase, the weights obtained during the second stage are used to extract the features, whereas the classification layers are obtained from the network trained on the source domain. The authors of [22] proposed a clustering based method to generate pseudo labels for the target domain, than the method minimizes the discrepancy of the gradients generated by the source and target images.

Image-to-image translation: The approaches outlined in the preceding section operate at the feature level. However, when the images of the source and target domains are visually different (e.g., color, style), an effective strategy for mitigating the domain gap is to employ image-to-image translation techniques [23], [24], [25]. These techniques address differences at the pixel level, aiming to transform an image from the source domain into one belonging to the target domain. Importantly, this transformation

preserves content while adjusting style and color elements. When pairs of images belonging to the source and target domains are available, a mapping between the two domains can be learned exploiting a conditional adversarial network [26]. The authors of [27] note that paired datasets are difficult to obtain in practice and introduce a method that translates images from a source domain X to a target domain Y in the absence of paired examples. As proposed in [27], the goal is to learn a function $G : X \longrightarrow Y$ such that the distribution of the transformed images $G(X)$ is indistinguishable from the distribution of Y. Since the translation between the two domains should be consistent, an inverse mapping $F : Y \longrightarrow X$ is introduced such that $F(G(X)) \approx X$. As discussed in previous works [28], [29], [30], the algorithms just described can be used in combination with the previous described domain adaptation techniques to deal with the domain gap. The images belonging to the source domain can be translated into the target domain and subsequently used as training images. The resulting model can be used directly on the target domain at test time. Vice versa, it is possible to train the model on the source domain and translate the test images to the source domain at inference time.

UDA in presence of multiple source or target domains: The domain adaptation problem typically considers a pair of source-target domains. However, in real-world scenarios, there are multiple source and target domains, necessitating algorithms to generalize across them. To address this, researchers have explored a broader scope of the problem by examining different contexts. The first context involves unsupervised domain adaptation with multiple labeled source domains and only one unlabeled target domain. In [31], the authors proposed a method based on the gradient reversal layer, discriminating between all $Target - Source_n$ pairs, where $d = 0, ..., D$ and D

is the number of source domains. Another approach presented in [32] comprises three components: a feature extractor, a moment matching module, and final classifiers. These methods often require access to multiple labeled source datasets, which may be available or easy to produce in certain cases. The second context focuses on a more realistic scenario with a single source domain and multiple target domains. Here, the presence of multiple target domains simulates situations where different devices with distinct lenses and image generation pipelines (IGP) produce diverse target domains. The authors of [33] proposed a method based on an autoencoder which finds a latent space which can capture domain invariant and domain dependent features that can generalize over multiple target domains. The authors of [34] presented a method that extends the idea proposed by [14] replacing a binary discrimination with a multi-class discrimination. The authors of [35] proposed a method based on an iterative multi-teacher knowledge distillation from multiple teachers to a common student.

2.2 Unsupervised Domain Adaptation for Object Detection

Object detection represents one of the most significant challenges in computer vision, which in recent years has found several applications in people's daily lives [36], [37], [38]. In this section we briefly describe the object detector algorithms and give more details on the UDA state-of-the-art algorithms for object detection.

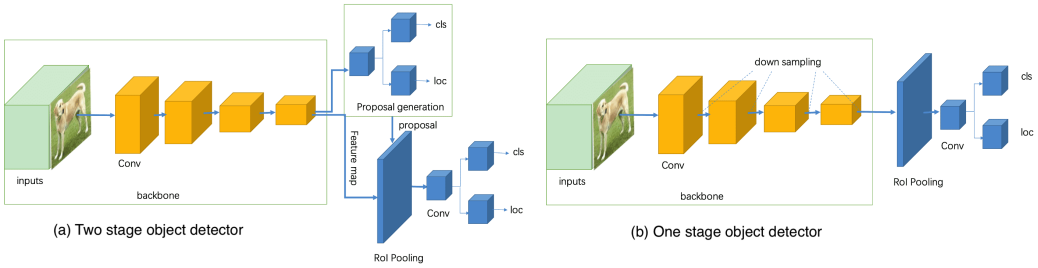


Figure 2.2: The figure illustrates the fundamental architectures of two-stage and one-stage detectors for object detection. (a) Two-stage detectors employ a region proposal network to generate region proposals, which are then passed to the classifier and regressor for further processing. (b) One-stage detectors operate by directly predicting bounding boxes from the information contained in input images, without the need for an intermediate proposal generation step.

2.2.1 Object Detection

Deep learning-based object detection algorithms can be divided into two main categories according to their architecture (Figure 2.2): the algorithms belonging to the "two-stage" category, whose main representative are Faster R-CNN [39], Cascade R-CNN [40] and Mask-RCNN [41], and those belonging to the "single-stage" category such as RetinaNet [42], SSD [43] and YOLO [44]. The former category is known for its high accuracy in object recognition and classification, while the latter category prioritizes computational efficiency over precision. Two-stage algorithms follow a two-step approach. In the first stage, they propose potential bounding boxes that may contain objects. Then, in the second stage, they perform classification and regression tasks on these proposed bounding boxes. For example, Faster R-CNN consists of a backbone network, a Region Proposal Network (RPN), ROI pooling, and classification/regression modules. The backbone network extracts a feature map from the input image, while the RPN generates region proposals using

convolution and classification/regression operations. The ROI pooling layer selects the corresponding regions on the feature map based on the generated bounding boxes, and the classification/regression modules process these regions to classify objects and refine the bounding boxes.
On the other hand, single-stage algorithms like RetinaNet utilize a backbone network, a feature pyramid network (FPN), and classification/regression subnetworks. The backbone network extracts multiple feature maps of varying resolutions. The FPN combines these feature maps to create a feature pyramid, enabling the detection of objects at different scales. The classification/regression subnetworks operate on the feature pyramid to classify objects and refine their bounding boxes. This allows single-stage algorithms to perform both tasks in a single forward pass, simplifying the overall process.

2.2.2 State-of-the-Art UDA Methods for Object Detection

Due to the crucial role of object detectors, researchers have sought to address the unsupervised domain adaptation challenge in the context of object detection. This entails introducing novel methodologies or adapting existing ones, initially designed for classification tasks and discussed in the UDA section. The authors of [45] present DA-Faster RCNN, a custom version of a Faster RCNN [39] that includes two modules: the first one aligns the features of the entire input (i.e., at the image level), the second module aligns the features before they are used for classification and regression (i.e. at the instance level). The authors of [29] propose to adapt source and target domains exploiting both high-and low-level features. The authors of [46] propose an architecture similar to the one presented in [45], but they add more discrim-

inators with a gradient reversal layer to the Faster-RCNN backbone. The authors of [47] propose a framework to align the source and target domains at the level of image regions extracted from the "region proposal network" of a Faster-RCNN. This architecture has two main components: 1) region mining, which extracts the regions of interest from the source and target images, groups them and selects the most important regions containing the objects; 2) the region level alignment, which learns to align the patches of the reconstructed images starting from the features selected by the previous module through adversarial learning. The authors of [30] presented an approach composed of two stages: 1) a domain diversification stage where the distribution of the labeled data is diversified by generating various distinctive domains shifted from the source domain using image to image techniques; 2) multi-domain-invariant representation learning, where adversarial learning is applied with a multi-domain discriminator to encourage feature to be indistinguishable across domains. The authors of [48] proposed to translate images from the source domain into the target domain using CycleGAN and trained an object detector using a self-training procedure to create pseudo label for the target dataset. The authors of [49] introduced in a SSD architecture a novel self-training method called weak self-training (WST) combined with the adversarial background score regularization (BSR) to prevent the degeneration of the performance due to incorrect pseudo label obtained using a naive approach reducing the amount of false negative and positive detections. The authors of [50] presented a framework which combines intermediate domains to progressively adapt feature alignment for object detection and a weighted task loss which weights the samples in the intermediate domain. The authors of [51] presented a method based on SSD which is divided in three steps: in the first step the SSD detector is pretrained using

the source images; in the second step the source images are converted to real with CycleGAN; in the third step SSD is trained using the converted source images, the target images and using the weak self-training method proposed by [49]. The authors of [52] proposed a Implicit Instance-Invariant Network (I^3Net), a single stage object detector which adapt the source and the target domain considering: 1) a strategy to assign large weights to those sample-scarce categories and easy-to-adapt samples considering the intra-class and intra-domain variation, 2) a module to suppress uninformative background features boosting the foreground object matching, 3) a module that align the category at different domain specific layers and regularize the average prediction of different layer respect to the same category. The authors of [53] introduced a generic approach based on an attention mechanism which allows to detect the important regions of the feature map extracted from the backbone on which adaptation should focus. The authors of [54] proposed a method which works at image and instance level aligning the two distributions so that well-aligned and poor-aligned samples are adaptively weighted based on the uncertainty of each sample. The authors of [55] presented a feature alignment method based on Faster RCNN which consist of three modules: 1) a global discriminator which align the feature extracted from the backbone; 2) category wise discriminators which aligns the features of each class belonging to the source and the target domains; 3) a memory guided attention mechanism which aids the category-wise discriminators to align category specific features between the two domains.

2.3 Unsupervised Domain Adaptation for Action Recognition

2.3.1 Action Recognition

Action recognition problem represents one of the most difficult task in Computer Vision. Unlike other tasks such as classification, object detection and segmentation, accurately classifying an action requires considering the space-time coordinates and capturing correlations between features extracted from consecutive frames. Additionally, actions can be analyzed from both a third-person and first-person perspective, increasing the complexity of the problem. Given the wide range of approaches that have been developed to tackle this problem, action recognition algorithms can be categorized into the following categories:

2D CNNs: These algorithms capture temporal information by aggregating frame-level features into a compact representation exploiting 2D convolutions. One popular approach is to process each frame independently using a 2D CNN and then employ techniques like pooling predictions across the entire video or utilizing recurrent neural networks like LSTM to model temporal dependencies ([56], [57], [58], [59]). These methods are computationally efficient but may struggle with capturing long-term temporal dependencies and modeling complex motion patterns.

3D CNNs: These algorithms leverage 3D convolutions to directly process video frames and capture spatio-temporal information. By extending 2D convolutions to the temporal dimension, 3D CNNs create a hierarchical representation of spatio-temporal data capturing both appearance and motion cues. These methods have shown promising results in action recognition tasks ([60], [61], [62], [63], [64], [65], [66]). They can be computationally

demanding and require a large amount of training data.

Transformers: Inspired by Transformers [67] and Vision Transformers (ViT) [68], researchers have adapted these architectures for action recognition tasks. Transformers excel in capturing long-range dependencies and modeling interactions between different frames in the video. They utilize self-attention mechanisms to effectively capture complex temporal relationships. These methods, such as [69], [70], [71], [72], [73], have demonstrated competitive performance in action recognition.

2.3.2 State-of-the-Art UDA Methods for Action Recognition

The challenging problem of unsupervised domain adaptation for action recognition has been addressed in recent years by an increasing number of works due to the several possible applications it finds in real-world scenarios. The authors of [74] presented a method that creates a subspace representation of the source and target domains and applies SA ([75]) or GFK ([76]) domain adaptation methods to perform sequence of adaptations for the video clips. The authors of [77] introduced TA^{3}N which utilizes a temporal relation module to align the source and target domains and learn the temporal relation across video sequences. The authors of [78] presented an Adversarial Bipartite Graph (ABG) framework. This framework learns a domain-agnostic video classifier by treating the features extracted by the convolutional layers as a Bipartite Graph and uses the adversarial learning paradigm to align class-conditional distributions. The authors of [79] proposed a self-supervised predictive method for video domain adaptation. It aims to predict the clip order using an adversarial loss. The authors of [80] presented CoMix which combines temporal contrastive learning, background mixing and supervised

contrastive learning through the use of pseudo target labels. The authors of [81] presented MA^{2}L-TD which performs multi-level alignments from low level (frame), to middle levels (segment), and high level (video). The authors of [82] proposed CO^{2}A, an approach for UDA that utilizes a contrastive loss for domain alignment and incorporates a classification and a contrastive head to stabilize training. The authors of [83] presented a novel approach that combines transformers with a novel alignment loss derived from the Information Bottleneck principle ([84], [85]). The authors of [86] proposed TCoN, a deep architecture integrating a crossdomain attention module in order to focus on shared key frames between source and target domains. The authors of [87] proposed MM-SADA which combines the adversarial learning paradigm with a self-supervision alignment classifier and learns a temporal correspondence between modalities across source and target features to enhance the feature generality in both domains. The authors of [88] presented a framework that performs unsupervised domain adaptation in two parts. It includes a spatio-temporal contrastive learning framework for self-supervised learning and a video-based contrastive distance metric to mitigate domain shift. The authors of [89] introduced a multimodal framework based on contrastive learning which jointly adapts cross-modal and cross-domain feature between source and target domains. The authors of [90] presented an audio adaptive encoder that, starting from the adaptation of the sound of the actions, guides the visual features to be invariant respect the source and target domains. The authors of [91] presented RNA-Net which utilizes RGB and audio information simultaneously and employs a novel feature-level loss to balance the contributions of the two modalities. The authors [92] presented a multi-modal methods that shares the knowledge between modalities integrating missing information of a specific modality exploiting the adversarial

learning paradigm and the complementary and spatial consensus modules to align the source and target domain.

Chapter 3

Unsupervised Domain Adaptation for Object Detection

3.1 Introduction

In this chapter, we investigate the use of unsupervised domain adaptation techniques for artwork detection. Specifically, we consider a scenario in which large quantities of labeled synthetic images are available, whereas only unlabeled real images can be used at training time. The synthetic images can be easily obtained starting from a 3D model of the cultural site acquired with a 3D scanner such as Matterport [1] and using the tool proposed in [9] to automatically generate the labeled data. The real unlabeled images can be easily collected visiting the cultural site acquiring videos with a wearable camera. Note that, since no manual labeling is required for the real images in the unsupervised settings, this procedure has a low cost. We hence

[1] https://matterport.com/

aim to train the object detection models using labeled synthetic images and real unlabeled images. To the best of our knowledge, there are not publicly available datasets to study domain adaptation for artwork detection in cultural sites. Therefore we collect and publicly release a suitable one which we name UDA-CH (Unsupervised Domain Adaptation on Cultural Heritage). We hence study the main unsupervised domain adaptation techniques for object detection on UDA-CH: 1) image-to-image translation and 2) feature alignment. We compare the performance of two popular object detection approaches, Faster R-CNN [39] and RetinaNet [42]. Since in our study RetinaNet obtained results more robust to the domain gap than Faster-RCNN, we propose a novel approach which combines feature alignment techniques based on adversarial learning [13] for unsupervised domain adaptation with the RetinaNet architecture. Our experiments show that the proposed approach greatly outperforms prior art. When combined with image to image translation, our method achieves a mAP of 58.01% on real data without seeing a single labeled real images at training time. To better demonstrate the effectiveness of the proposed method, we have also tested the generalization of the approach in urban scenario exploiting the popular Cityscapes dataset [93], [94].

3.2 Methods

We compare several approaches to unsupervised domain adaptation for object detection. Specifically we considered the following: 1) a baseline object detector without adaptation, 2) domain adaptation through image-to-image translation, 3) domain adaptation through feature alignment, 4) the proposed method based on RetinaNet and feature alignment and 5) approaches com-

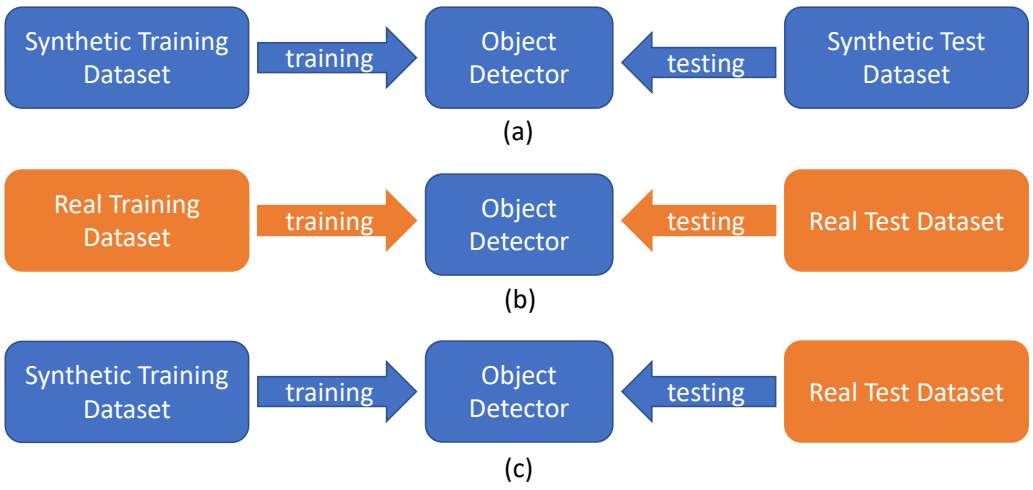


Figure 3.1: We used 3 different pipelines: (a) training and testing on the synthetic domain, (b) training and testing on the real domain, (c) training using synthetic images and testing on real images.

bining feature alignment and image-to-image translation. In the following section, we give details on all the compared approaches.

3.2.1 Baseline approaches without adaptation

To assess performance in the absence of domain shift, we train and test Faster RCNN and RetinaNet on the same domain (either synthetic or real images), as illustrated in Figure 3.1(a) and Figure 3.1(b). We also consider a model trained on synthetic images and tested directly on real test images, as illustrated in Figure 3.1(c). These methods allow to assess the gap between the two domains.

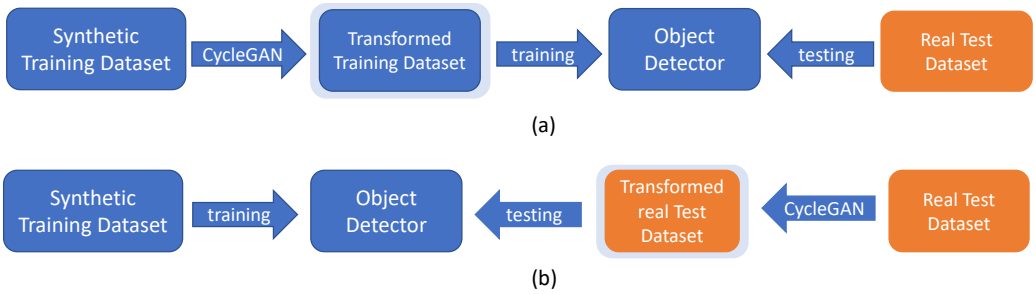


Figure 3.2: (a) Pipeline used to train models on synthetic images transformed to real with test performed on real images. (b) Pipeline used to train models on synthetic images with test performed on real images transformed to synthetic.

3.2.2 Domain adaptation through image-to-image translation

Transforming images from synthetic to real and vice versa is a common way to reduce the domain gap. In particular, we use CycleGAN [27] to transform images from one domain to another. We compare two approaches: 1) translating synthetic images to real, training Faster RCNN and RetinaNet on the transformed images and testing the two detectors with real images. This approach is illustrated in Figure 3.2(a); 2) translating real test images to synthetic, testing the two models that were previously trained on synthetic images as illustrated in Figure 3.2(b).

3.2.3 Domain adaptation through feature alignment

We consider DA-Faster-RCNN [45] and Strong-Weak [29] and compare their results with our method DA-RetinaNet described in the next subsection. All these methods use synthetic labeled images and unlabeled real images for training as shown in Figure 3.3.

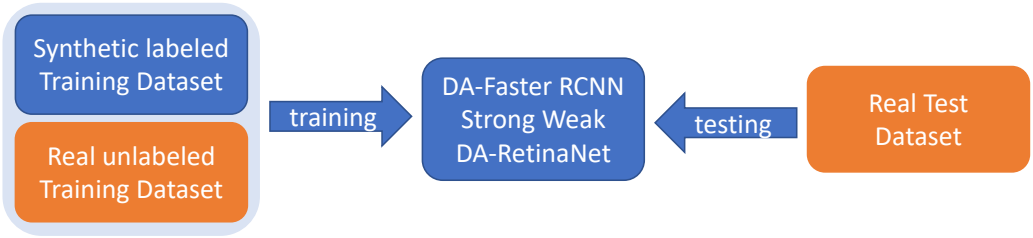


Figure 3.3: Pipeline used to train models based on feature alignment.

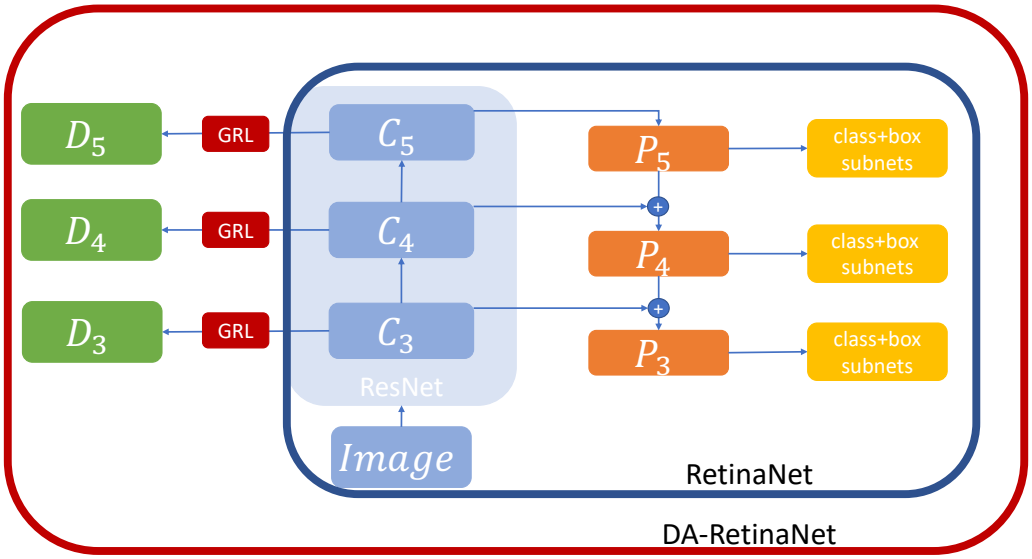


Figure 3.4: Architecture of the proposed DA-RetinaNet.

3.2.4 Proposed Method: DA-RetinaNet

The proposed method is based on RetinaNet architecture [42] and it is illustrated in Figure 3.4. At each level of the feature pyramid map (C_3, C_4 and C_5) in the ResNet backbone, we add a discriminator (D_3, D_4, D_5) with a Gradient Reversal Layer. The three discriminators have different architectures: D_3 has 3 convolutional layers with a kernel size of 1 and ReLU as activation function; D_4 has 3 convolutional layers with kernel size of 3 followed by batch normalization, ReLU and Dropout. At the end of the last convolutional layer there is a fully connected layer; D_5 has 3 convolutional layers with kernel size

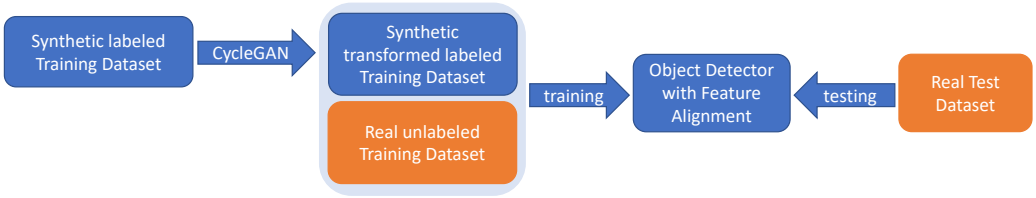


Figure 3.5: Pipeline used to combine feature alignment and image to image translation from synthetic to real techniques.

of 3 followed by batch normalization, ReLU and Dropout. After the convolutional layer there are 2 fully connected layers. Our idea follows [13], thus we train our model to minimize the cost function:

$$L = L_{class} + L_{box} - \lambda(L_{D3} + L_{D4} + L_{D5}) \tag{3.1}$$

where L_{class} is the sum of the losses of each classification subnet module, L_{box} is the sum of the losses of each regression subnet module. Their sum represent the standard RetinaNet loss. L_{D3}, L_{D4}, L_{D5} are the losses of each discriminator module and each of them is given by $L_{D_i} = \frac{1}{2}(L_{D_{s,i}} + L_{D_{t,i}})$ where $L_{D_{s,i}}$ and $L_{D_{t,i}}$ are respectively the losses computed by the discriminators when receive in input respectively synthetic and real images and defined using the Focal loss [42]. λ is the hyperparameter that balances RetinaNet and discriminators losses.

3.2.5 Domain adaptation through feature alignment and image to image translation

We combine the feature alignment techniques presented in Section 3.2.3 and Section 3.2.4 with image-to-image translation. This approach is similar to CyCADA proposed in [28] with the difference that we consider state-of-art feature alignment methods to perform the adaptation. We combine these

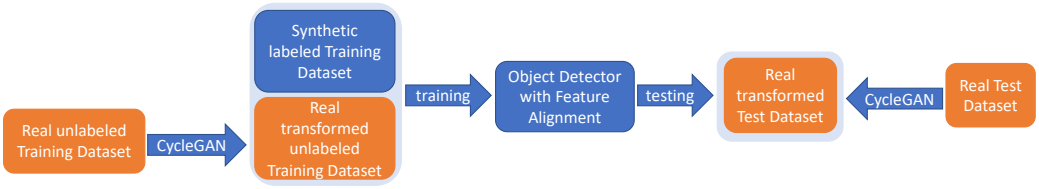


Figure 3.6: Pipeline used to combine feature alignment and image to image translation from real to synthetic techniques.

techniques in two ways: 1) transforming synthetic labeled images to real, then training feature-alignment-based architectures using transformed labeled and real unlabeled images (Figure 3.5); 2) transforming real unlabeled images to synthetic, then training feature alignment based architecture using synthetic labeled and transformed unlabeled images and testing on real images transformed to synthetic (Figure 3.6).

3.3 Experimental Settings and Results

This section presents the proposed dataset, reports and analyze the results of the methods presented in the previous section and discusses the computational resources required to train all the models.

3.3.1 Dataset

The proposed dataset [95] contains 16 objects that cover a variety of artworks which can be found in a museum like sculptures, paintings and books. Specifically, the dataset has been collected inside the cultural site "Galleria Regionale di Palazzo Bellomo" located in Siracusa, Italy[2]. We generated 75244 synthetic labeled images, we have used a 3D model of the museum acquired

[2] http://www.regione.sicilia.it/beniculturali/palazzobellomo/

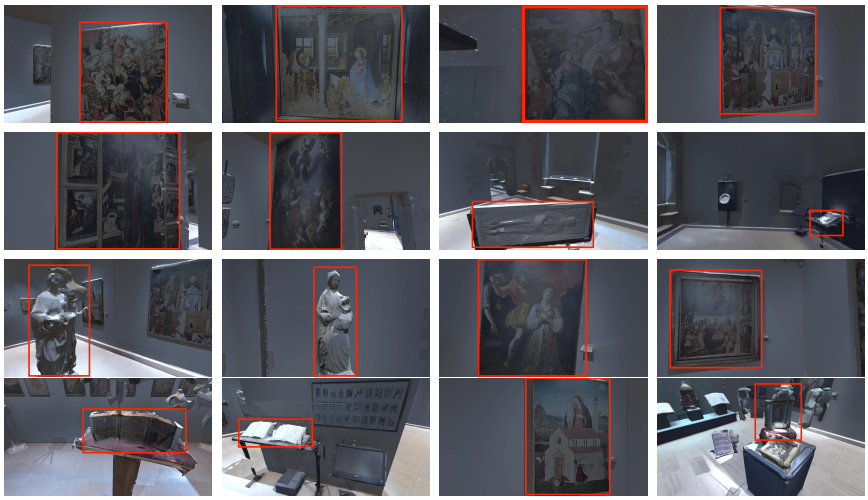


Figure 3.7: Sample synthetic images of the 16 artworks of our dataset.

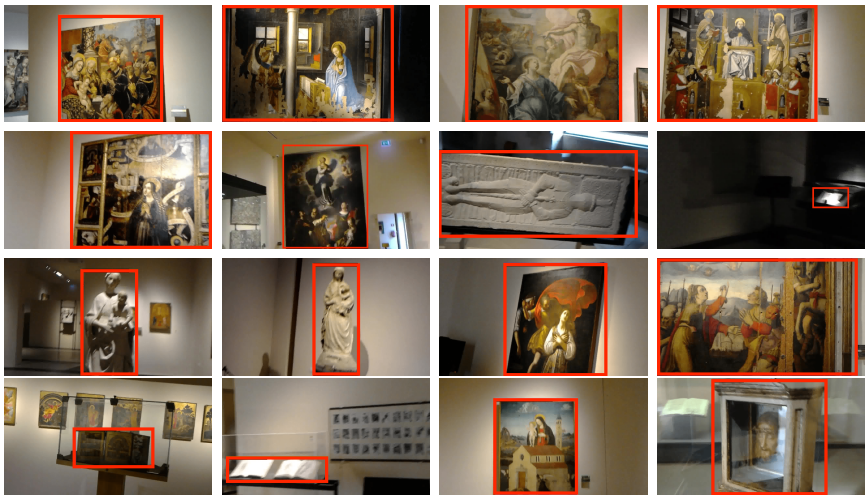


Figure 3.8: Sample real images of the 16 artworks of our dataset.

using Matterport[3], of the 16 artworks using the public tool proposed by the authors of [9] (see Figure 3.7). The tool proposed in [9] allows generate automatic labeled synthetic images, simulating a visitor who walks around the site while observing the artworks. Each image acquired during the simulation is associated to a semantic mask which allows to obtain bounding box anno-

[3]https://matterport.com/

tations for each image. Real images of the same 16 artworks are taken from the EGO-CH dataset proposed in [96] (see Figure 3.8), which contains videos of 70 subjects who visited two cultural sites which have been captured using a Microsoft HoloLens device. EGO-CH includes 176999 images manually annotated with bounding boxes. For the experiments, a subset of EGO-CH was taken into account. In particular, we considered 2190 images which contain the 16 artworks present in the synthetic dataset. To perform the experiments, we split both sets of synthetic and real images to training and a test set. We used 51284 synthetic and 1502 real images as training set and 23960 synthetic and 688 images as test set. The proposed dataset is available at the following URL: `https://iplab.dmi.unict.it/EGO-CH-OBJ-UDA/`.

3.3.2 Experimental Settings

We trained all the object detectors for 62K iterations starting from ImageNet [97] pre-trained weights. We used Faster-RCNN and RetinaNet Detectron2 [98] architectures[4] with ResNet101 [99] as backbone. The batch size has been set to 4 and the learning rate to 0.0002 for the first 30K iterations and multiplied by 0.1 for the remaining iterations. CycleGAN was trained for 60 epochs using the default parameters. For DA-Faster RCNN[5] and Strong-Weak[6] we used the settings proposed by the authors in their respective works [45] and [29]. DA-RetinaNet[7] was implemented using Detectron2 and it was trained with a batch size of 6 and learning rate of 0.0002 for the first 30K iterations. Also in this case, the learning rate has been then multiplied by

[4] `https://github.com/facebookresearch/detectron2`
[5] `https://github.com/krumo/Detectron-DA-Faster-RCNN`
[6] `https://github.com/VisionLearningGroup/DA_Detection`
[7] `https://github.com/fpv-iplab/DA-RetinaNet`

Table 3.1: Performance of Faster RCNN and RetinaNet trained and tested on images from the same domain.

	mAP	
Model	Synthetic	Real
Faster RCNN	**93.08%**	92.04%
RetinaNet	91.67%	**92.15%**

Table 3.2: Performance of Faster RCNN and RetinaNet trained on synthetic images for a different amounts of iterations and tested on real images.

	Training Iterations						
Model	6K	12K	22K	32K	42K	52K	62K
F. RCNN	2.27%	**9.67%**	5.79%	3.58%	3.33%	3.81%	3.62%
RetinaNet	9.83%	**14.44%**	13.22%	12.31%	12.09%	12.44%	11.97%

0.1 for the remaining iterations.

3.3.3 Baseline Results

Table 3.1 reports the results of the two models when they are trained and tested on the same domain. As can be noted, when images are sampled from the same distribution, these algorithms achieve good performance. Table 3.2 shows the performance achieved by Faster RCNN and RetinaNet when trained on synthetic images and tested on real images. The results highlight that models trained for few iterations generalize better than models trained for more iterations. RetinaNet is in general more robust to domain shift than Faster RCNN. In particular, RetinaNet trained for 12K iterations achieves an mAP of 14.44% vs 9.67% obtained by Faster RCNN and 11.97% vs 3.62%

Table 3.3: Results obtained transforming real images to synthetic at test time. The models have been trained on synthetic images. N.A. stands for No Adaptation.

		Training epochs for CycleGAN					
Model (iter)	N.A.	10	20	30	40	50	60
F. RCNN (62K)	3.62%	25.16%	25.49%	25.51%	26.68%	27.65%	**28.25%**
RetinaNet (62K)	11.97%	27.30%	32.14%	**34.15%**	32.66%	32.79%	32.82%
F. RCNN (12K)	9.67%	29.93%	32.84%	33.95%	31.45%	**34.19%**	31.58%
RetinaNet (12K)	14.44%	34.51%	35.45%	34.84%	35.34%	**35.76%**	35.74%

Table 3.4: Results obtained training the models on synthetic images transformed to real and tested on real images. N.A. stands for No Adaptation.

		Training epochs for CycleGAN					
Model	N.A.	10	20	30	40	50	60
F. RCNN	9.67%	18.76%	20.92%	21.22%	23.17%	24.45%	**26.03%**
RetinaNet	14.44%	40.13%	44.29%	46.05%	47.89%	49.96%	**55.54%**

considering 62K iterations. This suggests that training for more iterations both models increases the domain gap between the two distributions because the models learn to extract features specific to the source domain that do not generalize to the target domain. It is worth noting that, even the best RetinaNet model (14.44%) exhibits a drastic drop in performances if compared with the results of Table 3.1 (92.15%). This is due to the domain shift between synthetic images used for training and real images used for test.

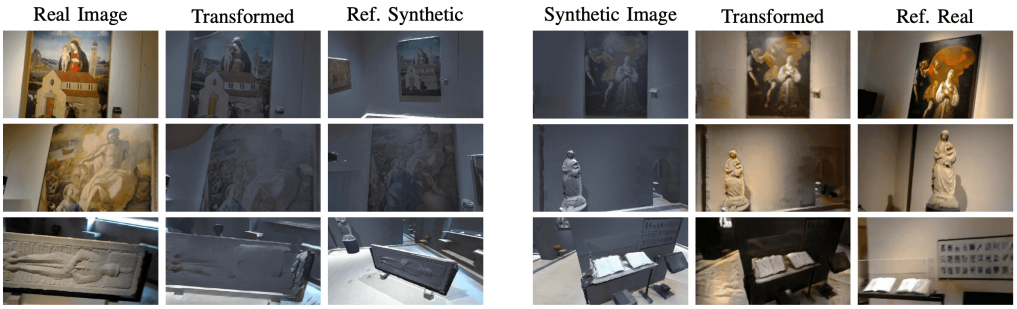
3.3.4 Image-to-Image translation Results

Table 3.3 shows the results of Faster RCNN and RetinaNet tested on real images transformed to synthetic using CycleGAN. We analyzed the perfor-

mances of both models trained for 12K and 62K iterations to explore the impact of overfitting. As shown in the Table 3.3, CycleGAN improves the performance of both models. RetinaNet performs better than Faster RCNN. Indeed Faster RCNN achieves performance similar to RetinaNet only when tested on images transformed using a CycleGAN model trained for 50 epochs. Table 3.4 reports the results of both models trained using synthetic images transformed to real. In this case, the performance of Faster RCNN (26.03%) are lower than the previous method which uses images translated from real to synthetic (28.25%). RetinaNet increases its performance by ∼20% from 35.76% to 55.54%. Even in this case, the results seem to confirm that RetinaNet is more robust to domain shift. While training CycleGAN for more epochs may allow for minor improvements, it should be noted that training CycleGAN for 60 epochs required about 61 days with a single NVIDIA® Tesla® K80. A detailed discussion on the training times of all methods is provided in Section 3.3.8. Figure 3.9 shows qualitative example obtained translating images from real to synthetic and vice versa. The first row of Figure 3.9 (a) shows an example of successful translation. In the second row the image is not correctly transformed due to light reflection, whereas in the third row the texture is destroyed during the transformation. First two rows of Figure 3.9 (b) show an example of successful translation while the last rows show a bad translation example where the background contains many artifacts.

3.3.5 Feature Alignment and Image-to-Image translation Results

Table 3.5 reports the mAP of the methods based only on feature alignment and combined with CycleGAN. As can be seen from Table 3.5, the proposed



(a) Transformation from real to synthetic. **(b)** Transformation from synthetic to real.

Figure 3.9: Qualitative CycleGAN results. We show the source domain (real synthetic), the transformed image, and a reference image for visual comparison.

Table 3.5: Results of DA-Faster RCNN, Strong-Weak and the proposed DA-RetinaNet combined with two different image-to-image translation approaches.

	image-to-image translation		
Model	None	Real2Syn	Syn2Real
DA-Faster RCNN	12.94%	19.88%	33.20%
Strong-Weak	25.12%	33.33%	47.70%
DA-RetinaNet	**31.04%**	**37.49%**	**58.01%**

DA-RetinaNet achieves better performances when compared to other methods. Without image-to-image translation, DA-RetinaNet obtains an mAP of 31.04% which is an increase in performance of about 6% as compared to Strong-Weak (25.12%). The improvement is about 11% when the models are combined with CycleGAN (58.01% vs 47.70%). Furthermore, it is worth noting that all models benefit from a performance improvement which varies between 21% and 27% when combined with CycleGAN.

Table 3.6: Ablation study about the impact of each discriminator D_i.

Model	D_3	D_4	D_5	mAP
RetinaNet (62K)				11.97%
RetinaNet (12K)				14.44%
DA-RetinaNet			✓	15.84%
DA-RetinaNet		✓		16.38%
DA-RetinaNet	✓			28.61%
DA-RetinaNet	✓	✓		30.52%
DA-RetinaNet	✓	✓	✓	**31.04%**

3.3.6 Ablation Study

Table 3.6 reports the results of DA-RetinaNet considering one, two or three Discriminators D_i without any image-to-image translation technique. As shown in the table, aligning features using the paradigm of adversarial learning improves in each case the performance of standard RetinaNet. D_3, the discriminator that aligns low level features, doubles the performance with respect to the standard RetinaNet model achieving a mAP of 28.61% vs 14.44%. The use of D_4 and D_5 allows to achieve similar performance (16.38% and 15.84%) improving the baseline results by about 1.5%. This is probably due to the design of the RetinaNet architecture. Indeed between the feature map C_4 and C_5 there are few convolutive layers. Combining the two discriminators which achieve the best performance allows to improve the mAP of about 2% (28.61% vs 30.52%). The best model is obtained by using all the discriminators (31.04%), which is our suggested design, as shown in Figure 3.4

Table 3.7: Summary table of the analyzed methods.

Object Detector	Adaptation	mAP
Faster RCNN	None	9.67%
RetinaNet	None	14.44%
Faster RCNN	Real2Syn (Test set)	34.19%
RetinaNet	Real2Syn (Test set)	35.76%
Faster RCNN	Syn2Real (labeled Training set)	26.03%
RetinaNet	Syn2Real (labeled Training set)	55.54%
DA-Faster RCNN	Feat.Align.	12.94%
DA-Faster RCNN	Feat.Align.+Real2Syn (Test set and unlabeled Training set)	19.88%
DA-Faster RCNN	Feat.Align.+Syn2Real (labeled Training set)	33.20%
Strong-Weak	Feat.Align.	25.12%
Strong-Weak	Feat.Align.+Real2Syn (Test set and unlabeled Training set)	33.33%
Strong-Weak	Feat.Align.+Syn2Real (labeled Training set)	47.70%
DA-RetinaNet	Feat.Align.	31.04%
DA-RetinaNet	Feat.Align.+Real2Syn (Test set and unlabeled Training set)	37.49%
DA-RetinaNet	Feat.Align.+Syn2Real (labeled Training set)	**58.01**%

3.3.7 Summary table and Qualitative Results

Table 3.7 summarizes all the performances of the analyzed methods with respect to the considered adaptation techniques. The table confirms that the proposed DA-RetinaNet achieves better performance than the compared methods. In particular, considering only feature alignment techniques, our architecture increases performance by about 5% when compared to Faster

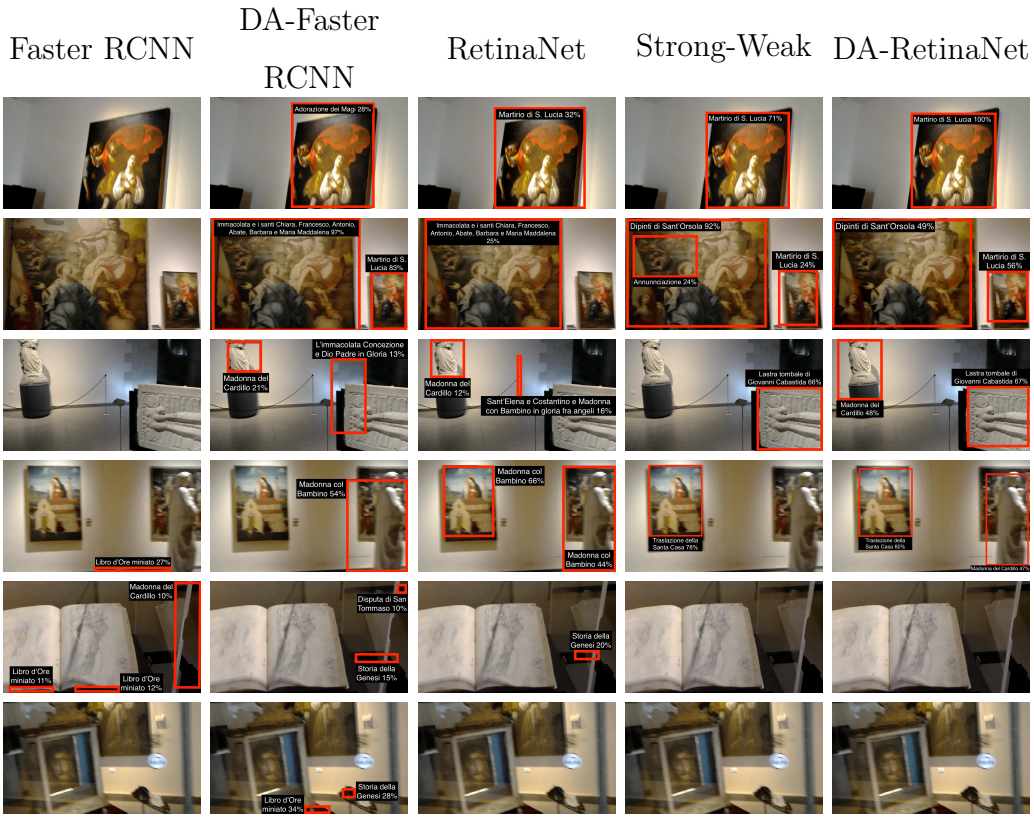


Figure 3.10: Qualitative results of baseline and feature alignment approaches.

RCNN with CycleGAN, and by 6% compared to Strong-Weak. Again, our method increases the performance of a standard RetinaNet with CycleGAN by about 2.5% (55.54% vs 58.01%). Figure 3.10 shows qualitative result of the baseline and the models based on feature alignment. Faster RCNN does not detect any object and in some cases its predictions are false positive. DA-Faster RCNN and RetinaNet correctly detect objects in the "easy" examples (first two rows), with some misclassification problems when there are more objects and occlusions (last three rows). Strong-Weak and DA-RetinaNet are more accurate in detection but they still produce some false positive and false negative predictions. Figure 3.11 reports the qualitative results of the previous five methods combined with CycleGAN to translate images from

Figure 3.11: Qualitative results of the baseline and feature alignment combined with CycleGAN.

synthetic to real. Faster RCNN and DA-Faster RCNN have similar results to DA-RetinaNet but they have much more false positive detections. Strong-Weak and RetinaNet combined with CycleGAN correctly detect the objects of the first three rows. Strong-Weak is less accurate than RetinaNet but has less false positive detections. DA-RetinaNet combined with CycleGAN perfectly detects artworks in the first four rows with only a misclassification in the fourth rows behind the statue. As can be noted, even these models are not able to detect object in the last two rows. Possible reasons are: 1) bad translation results from synthetic to real, 2) few synthetic object are

Table 3.8: Training times required by the models.

Model	Hours (Days)
RetinaNet (12K iterations)	~ 10 (~ 0.5)
RetinaNet (62K iterations)	~ 65 (~ 3)
DA-RetinaNet	~ 67 (~ 3)
Faster RCNN (62K iterations)	~ 131 (~ 5.5)
DA-Faster RCNN	~ 142 (~ 6)
Strong-Weak	~ 147 (~ 6)
CycleGAN	~ 1470 (~ 61)
CycleGAN + RetinaNet	~ 1535 (~ 64)
CycleGAN + DA-RetinaNet	~ 1537 (~ 64)
CycleGAN + Faster RCNN	~ 1601 (~ 66)
CycleGAN + DA-Faster RCNN	~ 1612 (~ 67)
CycleGAN + Strong-Weak	~ 1617 (~ 67)

not similar to their real counterpart, 3) some synthetic objects are similar to each other (e.g. some books).

3.3.8 Analysis of Computational Resources

Table 3.8 shows the training times required by the algorithms using a single NVIDIA® Tesla® K80. We use the same batch size for each object detector to evaluate the training times. Training CycleGAN for 60 epochs required 61 days in the considered settings. Methods based on feature alignment require from 3 to 6 days depending on the considered object detector. In particular, DA-Faster RCNN, Strong-Weak and DA-RetinaNet have only a small computational overhead given by the presence of the discriminators. However,

even if these methods required less time when compared to CycleGAN, they have limited performance when compared to their counterparts who make use of image-to-image translation (e.g. DA-RetinaNet: 31.04 % vs 58.01 %, Strong-Weak: 25.12 % vs 47.70 %, DA-Faster RCNN: 12.94 % vs 33.20 %). We argue that more attention should be devoted to such approaches in order to minimize training times.

3.3.9 Results on Cityscapes Dataset

To better asses the performance of the proposed method and to understand generalization capability over datasets, we have performed experiments on the Cityscapes dataset [93] [94]. To this aim, we trained RetinaNet and DA-RetinaNet for 50K iteration with a learning rate of 0.0002, batch size of 4 and starting from weights pre-trained on ImageNet. Following [29], we used Cityscapes [93] as source domain and Foggy-Cityscapes [94] as target domain. Both dataset have 2975 images in the training set. We reported results on the 500 images of the validation set. Table 3.9 reports the results obtained by standard object detector architectures and domain adaptation methods based on feature alignment. The table highlights that standard RetinaNet achieves better performance than Strong-Weak and Diversify and Match by about 6%. The proposed DA-RetinaNet increases performance by 4%, 10%, and 24% if compared respectively with standard RetinaNet, Strong Weak and Diversify and Match, and DA-Faster RCNN. However there is still a gap between the best results obtained by the proposed architecture and the result of the Oracle which is obtained training and testing RetinaNet on the Foggy Cityscapes dataset, which suggests that there is still room for improvement.

Table 3.9: Results adaptation between Cityscapes and Foggy Cityscapes dataset. The performance scores of the methods marked with the "*" symbol are reported from the authors of their respective papers.

Model	mAP
Faster RCNN* [29]	20.30%
DA-Faster RCNN* [45]	27.60%
Strong-Weak* [29]	34.30%
Diversify and Match* [30]	34.60%
RetinaNet	40.25%
DA-RetinaNet	**44.87%**
Oracle	53.46%

3.4 Conclusion

We considered the problem of Unsupervised Domain Adaptation for object detection in cultural site. To conduct our study, we created a new dataset consisting of 75244 synthetic images and 2190 real images of 16 artworks, which we publicly release. To better assess generalization of the compared approaches, we have also performed experiment with a dataset related to urban environment. Experiments showed that the proposed DA-RetinaNet method achieves better performance compared to DA-Faster RCNN and Strong-Weak. At the same time, the results obtained by these methods based on feature alignment achieved very poor performance if compared to their counterparts combined with image-to-image translation techniques. DA-RetinaNet performed better than others also when combined with CycleGAN. However, using CycleGAN with this dataset required a high computational training cost. We hope that the proposed dataset will encourage

research on this challenging topic and that the proposed DA-RetinaNet will serve as a strong baseline for future works.

Chapter 4

Unsupervised Multi-Target Domain Adaptation for Object Detection

4.1 Introduction

In this chapter, we investigate the unsupervised multi-target domain adaptation problem. In fact, in real-workly scenarios, object detection algorithms often need to be deployed to different devices, which are generally equipped with different cameras. This constraint further reduces the generalization ability of the object detection methods in real scenarios. Figure 4.1 reports some qualitative results of a standard object detector trained and tested on different domains of images of a cultural sites: a set of synthetic images, real images acquired with an HoloLens device, and a set of images collected with a GoPro. As can be noted, the detection of the artworks works perfectly only if the the training and test set belong to the same data distribution.

Domain adaptation techniques [13] can be used to reduce the domain dif-

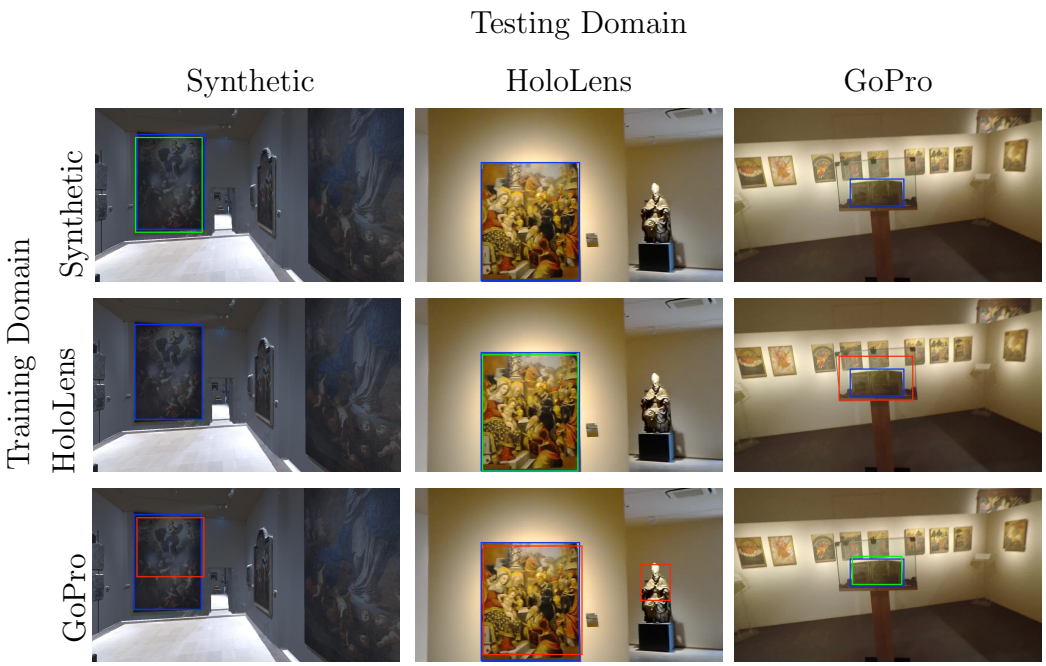


Figure 4.1: Qualitative results of a standard object detector trained and tested on different domains. Blue bounding boxes represent the ground truth, green boxes represent correct detections, whereas red boxes indicate wrong detections (either object localization or classification). The model was trained using the domain indicated in the rows and tested using the domain reported in the columns.

ference between source and target sets. However, in a real scenario, the algorithm should also generalize to images collected using multiple cameras as in the example in Figure 4.1, which may present subtle characteristics capable of affecting model performance. We propose to tackle this problem as a multi-target unsupervised domain adaptation task in which there is a labeled source domain (the synthetic data) and more than one unlabeled target domains (the target images acquired using different cameras). We note that, since target unlabeled images can be acquired with a little effort, the task setup involves a small additional overhead as compared to single-

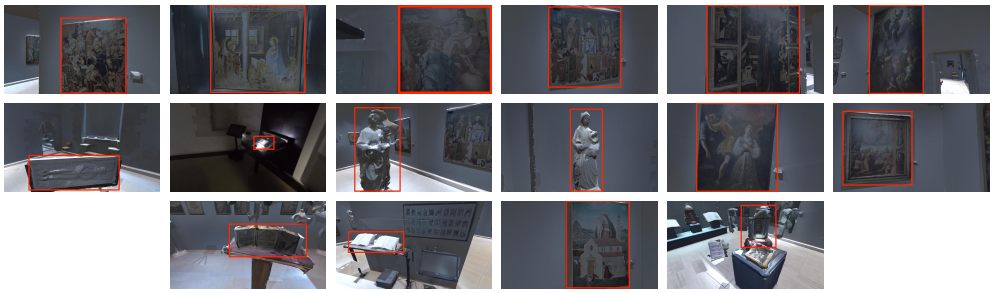
target domain adaptation. We hence investigate whether the presence of more than one target domains can assist the domain adaptation process in the considered settings. To analyze the problem, we introduce a new dataset of both synthetic and real images collected in a cultural site and suitable to study unsupervised multi-camera domain adaptation. We perform experiments to assess the ability of current domain adaptation approaches to generalize across multiple cameras. We hence investigate a generalization of current state-of-the-art methods which is shown to outperform current methods. The proposed method outperforms the results of current state-of-the-art methods by up to +23% mAP. To provide further evidence of the effectiveness of the proposed technique, we conducted an evaluation in an urban environment using the highly regarded Cityscapes dataset [93]. Specifically, we examined the applicability of our approach in challenging scenarios such as foggy [94] and rainy conditions [100].

4.2 Dataset

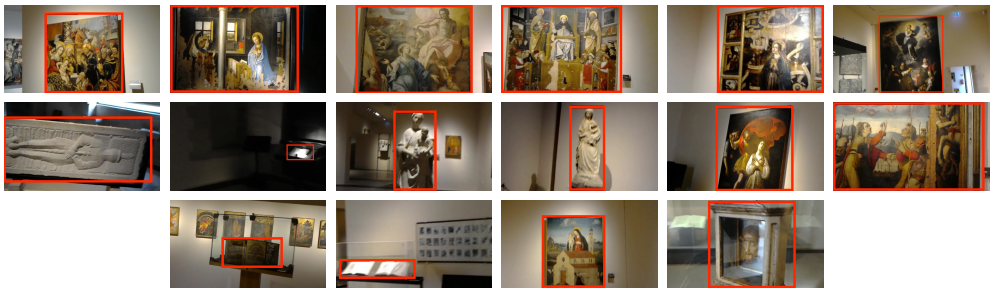
To study the problem, we created a dataset[1] that contains images of 16 artworks included in the cultural site "Galleria Regionale di Palazzo Bellomo[2]". The collection covers different types of artworks, as well as books, sculptures and paintings. We considered three domains: i) synthetic images generated from a 3D model of the cultural site and automatically labeled during the generation process, ii) real images collected by 10 visitors with a HoloLens device and manually labeled, iii) real images collected by the same visitors with a GoPro and manually labeled. Figure 4.2 shows some examples of images belonging to the three domains. As can be noted, synthetic images

[1]The dataset is available at `https://iplab.dmi.unict.it/OBJ-MDA`

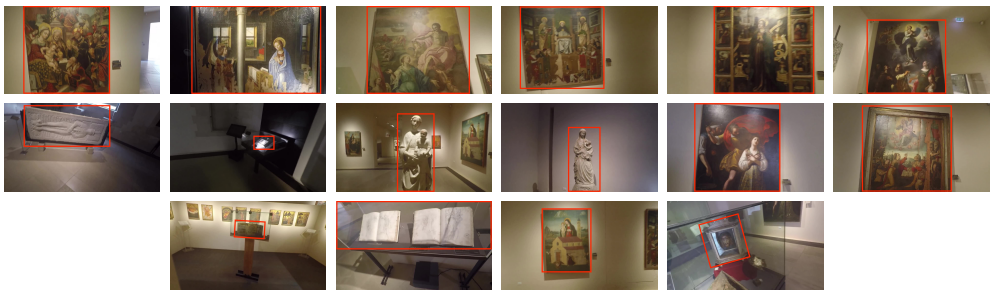
[2]`http://www.regione.sicilia.it/beniculturali/palazzobellomo/`



(a) Synthetic images.



(b) Images acquired with HoloLens.



(c) Images acquired with GoPro.

Figure 4.2: Example of labeled images of the 16 artworks with respect to the three considered domains: (a) Synthetic images, (b) images acquired with HoloLens, (c) images acquired with GoPro.

differ from the real images in style, shapes of the 3D objects (e.g., observe the statues of Figure 4.2) and field of view. Similarly, real images acquired with two the different devices differ only in style and field of view. The three sets of images have been collected as detailed in the following:

- Synthetic labeled images (used as source domain): these images have

Table 4.1: Statistics of the proposed dataset for unsupervised multi-target domain adaptation. The average occupied area (last column) is the average percentage of the image occupied by the bounding boxes of the considered object class.

	Synthetic Domain (Source)		HoloLens Domain (Target)		GoPro Domain (Target)		Total object instances for each class	Average occupied area
Object Instances	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**		
Annunciazione	1301	605	191	69	211	74	2451	42.87%
Libro d'Ore miniato	1628	722	105	30	146	42	2673	8.02%
Lastra tombale di Giovanni Cabastida	2313	1181	200	100	247	114	4155	24.58%
Madonna del Cardillo	2345	1264	106	40	166	66	3987	9.74%
Disputa di San Tommaso	2202	965	100	46	155	67	3535	28.17%
Traslazione della Santa Casa	1904	964	161	46	225	71	3371	22.24%
Madonna col Bambino	2135	1044	119	47	161	46	3552	21.93%
L'immacolata Concezione e Dio Padre in Gloria	2557	1139	77	39	100	54	3966	35.70%
Adorazione dei Magi	1517	478	64	36	69	39	2203	30.35%
Sant'Elena e Costantino e Madonna con Bambino in gloria fra angeli	3285	1031	94	44	153	61	4668	33.72%
Taccuini di disegni	1617	513	59	33	75	39	2336	22.34%
Martirio di S. Lucia	3567	2353	106	36	184	45	6291	22.55%
Volto di Cristo	990	519	25	26	50	36	1646	11.74%
Dipinti di Sant'Orsola	2721	1897	83	69	125	86	4981	30.56%
Immacolata e i santi Chiara, Francesco, Antonio, Abate, Barbara e Maria Maddalena	3824	2424	104	69	187	89	6697	32.36%
Storia della Genesi	927	375	55	14	57	15	1443	22.79%
Total object instances for each split	34833	17474	1649	744	2311	944		

been generated using the tool proposed by [9]. The tool allows to annotate in 3D the position of artworks in the 3D model of a cultural site and simulates an agent navigating the environment while acquiring egocentric images of the observed artworks. The acquired images are automatically labeled by projecting the 3D bounding boxes of the objects onto the generated 2D images. This set contains 75244 images divided in 51284 training images and 23960 test images.

- Target images acquired using a HoloLens: this set of data has been sampled from the work of [96] where data has been manually annotated drawing a bounding box around each of the 16 object to match the same artworks present in the synthetic set. This set contains 2190 images divided in 1502 for the training and 688 for the test;

- Target images acquired using a Gopro: the dataset was created similarly to the previous one HoloLens. The images have been collected by the same visitor which have visited the site wearing both HoloLens and GoPro wearable cameras. This set contains a total of 2707 images splitted into 1911 for the training and 796 for the test.

Table 4.1 shows the distribution of the object instances in the proposed dataset. As can be noted, the HoloLens and GoPro domains have a number of object instances less than ten times smaller than the synthetic domain. The table also highlights that the proposed dataset is challenging for domain adatation for object detection due to the average size of each object. Indeed, the biggest object present in the dataset occupies only the 42.87% of the images' area while the smallest occupies 8.02% of the frame.

4.3 Methods

In this section, we discuss the compared methods and present the proposed one setting the number of the target domains to 2.

4.3.1 Baselines without domain adaptation

We analyze the behaviour of two state-of-the-art object detectors: RetinaNet [42] and Faster RCNN [39]. We train and test both detectors on the target domains to produce "Oracle results" and assess the performance drop observed when the algorithms are trained on synthetic images and tested on the real images of the target domains.

4.3.2 Domain adaptation based on feature alignment

State-of-the-art domain adaptation methods for object detectors commonly consider only one source domain and one target domain. To study whether these state-of-the-art methods can be used to tackle multi-camera domain adaptation, we consider a naive approach which merges the two target domains into a single one. In particular, we considered the following unsupervised domain adaptation methods for object detection: DA-Faster RCNN [45], Strong Weak [29], DA-RetinaNet [101] and CDSSL [48].

4.3.3 Domain adaptation through feature alignment and image to image translation

Feature alignment methods aim to reduce the difference between source and target domains at the feature level without taking into account the difference at pixel level (like style, color, shape etc.) which are present between the source and targets domains. For this reason, we combine feature alignment methods with image to image translation methods to reduce the gap also at the pixel level. For the image to image translation task we used the CycleGAN alorithm [27] to translate synthetic images to real.

4.3.4 Proposed Method

The training of the proposed method comprises three stages that will be discussed in order of execution in the following sections. Each of them contributes to improving the performance of the object detector and works to adapt the two distribution at different levels. Figure 4.3 shows an overview of the general pipeline of the proposed method.

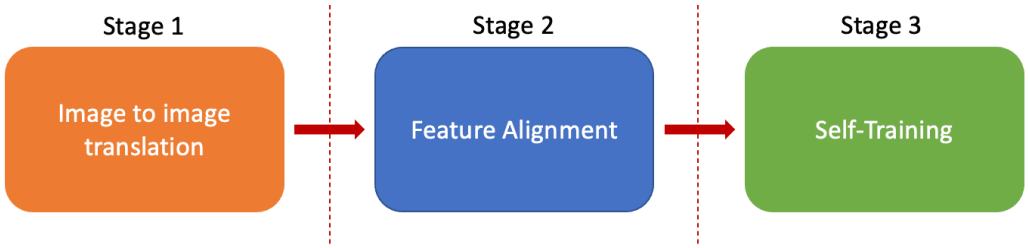


Figure 4.3: Pipeline of the proposed method. In the Stage 1 synthetic domains is translated to the real domains to reduce the gap at pixel level. In the Stage 2 the gap at feature level is reduced using feature alignment method. In Stage 3 an iterative self-training procedure is used to produce pseudo labels for the target domains.

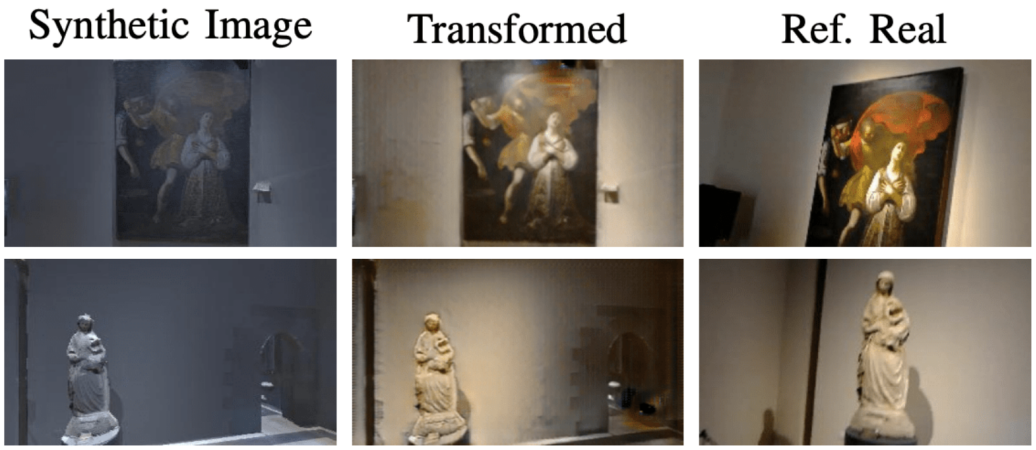


Figure 4.4: Qualitative results obtained using CycleGAN as image to image translation method (Stage 1). Synthetic images (left) are translated to the merged target domains (center). Real images similar to the translated ones are also reported for reference (right).

4.3.4.1 Image to Image Translation

Synthetic images generated from a model acquired using a 3D scanner, such as Matterport [3], differ in general from real images in the style and shape

[3] https://matterport.com/

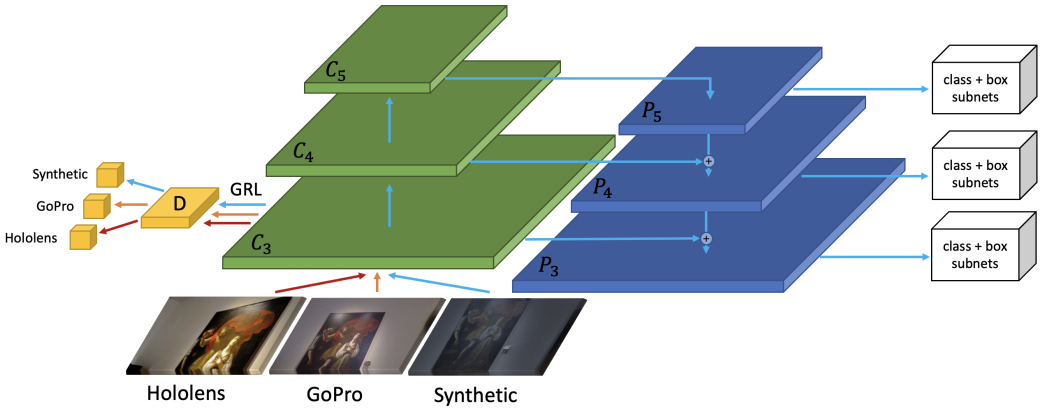


Figure 4.5: Architecture of the proposed MDA-RetinaNet model.

of the object which can affect object detection performance. To reduce this diversity, the first step of our method consists in mitigating the style and shape differences using an image to image translation method. In particular, we used CycleGAN to transform training synthetic images into the real. In the later stages of our pipeline, the object detection model will be trained on the transformed images and tested directly on the real images. This step is optional in our pipeline for two reasons: 1) it can be computationally expensive when the datasets are large; 2) when the target and the source domain are not too similar, this transformation can be not sufficiently accurate. Figure 4.4 shows some qualitative results of this translation. As can be noted, the transformed images look more similar to the real counterpart after the transformation.

4.3.4.2 Feature Alignment

Although image to image translation can be used to reduce the differences in terms of style and shape, the features extracted from the two domains can still be different. For this reason, in this second stage we propose an object detection architecture which jointly adapts the features during the training.

State-of-the-art domain adaptation methods for object detection do not consider the existence of multiple target domains. To take advantage of multiple unlabeled target domains during the training, we propose a model, that we called MDA-RetinaNet, to address the problem of unsupervised domain adaptation for object detection based on adversarial learning [13]. Figure 4.5 shows the architecture of the proposed method which builds on RetinaNet [42]. To reduce the domain gap present at the feature level, we attach a domain discriminator with a gradient reversal layer to the feature map C_3 obtained from the ResNet backbone [99]. In particular, to adapt multiple domains (in this case 1 source and 2 targets in our experiments) we consider a multi-class classifier D which discriminates among all of them. The discriminator has 3 convolutional layers with kernel size equal to 1, followed by a ReLU activation function. Following [13], we place a gradient reversal layer at the input of the discriminator and train the model by minimizing the following loss function:

$$L = L_{class} + L_{box} - \lambda(L_D)$$

where L_{class} and L_{box} are the regression losses of RetinaNet, L_D is the loss of the discriminator module and λ is an hyper-parameter that balances the object detection and domain adaptation losses. This approach differs from standard methods that use a binary classifier used to only discriminate features belonging to the source and target domains, hence ignoring the presence of multiple targets. We hypothesize that, providing a multi-classes discriminator, the model will learn to extract features which are not only indistinguishable across synthetic and real domains, but also indistinguishable across the different real cameras. It is important to highlight that this type of adaptation allow to learn a combination of weights that extract feature maps using the backbone that generalize to the different domains. No adaptation

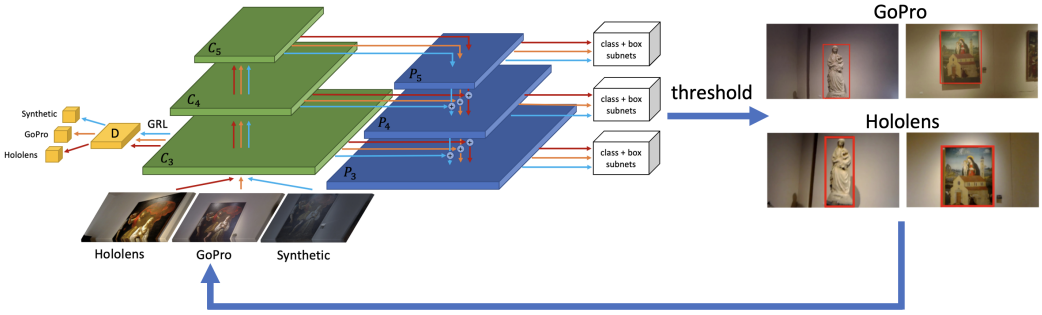


Figure 4.6: Self-training module for MDA-RetinaNet.

is directly enforced for the layers involved in the classification and regression of the bounding boxes (Figure 4.5 white modules).

4.3.4.3 Self-Training

As noted in the previous section 4.3.4.2, the adaptation provided in the second stage is at the level of the features extracted by the backbone, whereas the classification and regression layers which detect the objects are trained only with the synthetic images due to the absence of labels for the real images. To tackle this limitation, we generate pseudo labels for the real images by retraining the model with the predictions on real images above a given confidence threshold produced by the model trained at the second stage of our pipeline (see Figure 4.3). This allows to train MDA-RetinaNet in a supervised way as illustrated in Figure 4.6 by exploiting the obtained pseudo labels. This latest module is trained in an iterative way, gradually increasing the threshold used to generate the pseudo labels to reduce the error of potentially wrong predicted labels. Algorithm 1 reports the complete procedure of the proposed method.

Algorithm 1: Proposed multi-target domain adaptation for object detection algorithm.

Input: $S = \{x_s^n, y_s^n\}$ the source domain, $T = \{T_1, T_2, ...T_D\}$ target domains, $converge = false$;

Step 1: transform the set S into T using CycleGAN;

Step 2: train MDA-RetinaNet using S' and T to adapt the features;

Step 3: set the threshold $t = 0.75$ and produce the pseudo labels y_d^n for each images in each target $T_1, T_2, .., T_D$ using MDA-RetinaNet;

Step 4:

while *!converge* **do**

    train MDA-RetinaNet using y_d^n;

    produce the new pseudo label y_d^n;

    if $t < 0.9$ **then**

        $t = t + 0.05$;

    else

        $converge = true$;

    end

end

Output: B - the set of predicted bounding boxes.

4.3.5 Experimental Settings

All the compared models were trained for 60K iterations using weights pre-trained on ImageNet [97]. We set the learning rate to 0.0002 for the first 30K iterations, then we multiply it by 0.1 for the remains 30K iterations. DA-Faster RCNN, Strong Weak and CDSSL were trained with the same parameters proposed by the authors in their respective papers [45, 29, 48]. The batch size was set to 4 for RetinaNet, 6 for DA-RetinaNet (4 source

and 2 target images, 1 HoloLens and 1 GoPro) and 8 for MDA-RetinaNet[4] (4 source and 4 target images divided in 2 HoloLens and 2 GoPro images). MDA-RetinaNet was implemented using Detectron2 [98]. To reduce the noise of the initial training of the Discriminator D, we adapt the λ hyperparameter following the update rule proposed by [13]. The second stage (Section 4.3.4.2) is performed only one time to produce initial pseudo labels. The Self-Training stage (Section 4.3.4.3) is executed 4 times gradually increasing the threshold used to generate the new pseudo labels which will be used in the next iteration. CycleGAN was trained for 60 epochs using the default parameters.

4.4 Results

This Section reports and analyzes the results of the experimental analysis.

4.4.1 Feature Alignment Results

Table 4.2 reports the results of the feature alignment based models. The first two rows show the results of the baseline Faster RCNN and RetinaNet modules trained with synthetic images and tested on HoloLens and GoPro without any domain adaptation technique. It is worth noting that RetinaNet is less sensitive to the domain gap, obtaining an mAP $\sim$7% higher than Faster RCNN (14.10% vs 7.61% on HoloLens and 30.39% vs 37.13% on GoPro). For this reason, we focused our further experiments considering RetinaNet as backbone for the object detector in our proposed methods. The second group of rows (rows 3-6 of Table 4.2) report the results of state-of-the-art methods adapted for this specific task. In particular, due to the fact that these methods are able to work only with a single target, we merged

[4]code available at https://github.com/fpv-iplab/STMDA-RetinaNet

Table 4.2: Results of baseline and feature alignment methods. S refers to Synthetic, H refers to HoloLens and G to GoPro. ST indicates the self-training procedure.

Model	Source	Target	Test H	Test G
Faster RCNN [39]	S	-	7.61%	30.39%
RetinaNet [42]	S	-	14.10%	37.13%
DA-Faster RCNN [45]	S	H+G	10.53%	48.23%
Strong Weak [29]	S	H+G	26.68%	48.55%
CDSSL [48]	S	H+G	28.66%	45.3%
DA-RetinaNet [101]	S	H+G	31.63%	48.37%
MDA-RetinaNet	S	H, G	34.97%	50.81%
MDA-RetinaNet + ST	S	H, G	**54.36%**	**59.51%**
Faster RCNN [39] (Oracle)	H	-	91.97%	76.88%
Faster RCNN [39] (Oracle)	G	-	68.65%	89.21%
RetinaNet [42] (Oracle)	H	-	92.44%	77.96%
RetinaNet [42] (Oracle)	G	-	69.70%	89.69%

the HoloLens and GoPro datasets into one. The proposed MDA-RetinaNet performs better than the other models and outperforms the best state-of-the-art method, DA-RetinaNet, by ∼3% for HoloLens (34.97% vs 31.63%) and ∼2% for GoPro (50.81% vs 48.37%). The last row shows the results of MDA-RetinaNet combined with the self-training procedure. As the results highlight, this combination allows to increase the performances of ∼23% if compared with DA-RetinaNet (54.36% vs 31.63%) for HoloLens, ∼11% (59.51% vs 48.37%) for GoPro and ∼20% if compared with MDA-RetinaNet without self-training (54.36% vs 34.94%) for HoloLens and ∼9% (59.51% vs 50.81%) for GoPro. Furthermore, the performance gap between HoloLens

Table 4.3: Results of feature alignment methods combined with CycleGAN. H refers to HoloLens while G to GoPro. "{G, H}" refers to synthetic images translated to the merged HoloLens and GoPro domains. ST indicates self-training procedure.

Model	Source	Target	Test H	Test G
Faster RCNN [39]	{G, H}	-	15.34%	63.60%
RetinaNet [42]	{G, H}	-	31.43%	69.59%
DA-Faster RCNN [45]	{G, H}	H+G	32.13%	65.19%
Strong Weak [29]	{G, H}	H+G	41.11%	66.45%
DA-RetinaNet [101]	{G, H}	H+G	52.07%	71.14%
CDSSL [48]	{G, H}	H+G	53.06%	71.17%
MDA-RetinaNet	{G, H}	H, G	**58.11%**	**71.39%**
MDA-RetinaNet + ST	{G, H}	H, G	**66.64%**	**72.22%**
Faster RCNN [39] (Oracle)	H	-	91.97%	76.88%
Faster RCNN [39] (Oracle)	G	-	68.65%	89.21%
RetinaNet [42] (Oracle)	H	-	92.44%	77.96%
RetinaNet [42] (Oracle)	G	-	69.70%	89.69%

and GoPro with this last model it is almost negligible.

4.4.2 Feature Alignment and Image to Image translation Results

Table 4.3 shows the results obtained combining the baseline and feature alignment methods with CycleGAN. The first two rows report the results of Faster RCNN and RetinaNet when trained on synthetic images transformed to the merged HoloLens and GoPro domain. As can be noted, pixel

level domain adaptation allows to significantly increase the performance of Faster RCNN and RetinaNet respectively by about 8% (7.61% vs 15.34%) and 16% (14.10% vs 31.43%) on HoloLens and by about 33% (30.39% vs 63.60%) and 32% (37.13% vs 69.59%) on GoPro, reducing the gap between synthetic and real images. The middle part of the table shows the results of the methods based on feature alignment. Also in this case, MDA-RetinaNet achieves an higher mAP with respect to the best state-of-the-art method, CDSSL, (53.06% vs 58.11% for HoloLens and 71.17% vs 71.39% for GoPro) which further improves if we introduce the self-training procedure (58.11% vs 66.64% for HoloLens and 71.39% vs 72.22% for GoPro). It is worth noting that, with self-training the gap in performances between HoloLens and GoPro is reduced from ∼13% to ∼6% which suggest that the model acquires knowledge from the GoPro images that is useful to detect object in the HoloLens domain. Furthermore, the performance of MDA-RetinaNet with self-training is really close to the performance of the RetineNet oracles when trained with the labeled HoloLens domain and tested on GoPro and vice versa (66.64% vs 69.70% for HoloLens and 72.22% vs 77.96% for GoPro). However, there is still space of improvement if we consider the performances of the oracles trained and tested in their respective domains, which makes proposed dataset still challenging (66.64% vs 92.44% for HoloLens and 72.22% vs 89.69% for GoPro).

4.4.3 Ablation Study

Table 4.4 reports the ablation study of the proposed MDA-RetinaNet model and compares the results with respect to the DA-RetinaNet architecture. We evaluated the models on HoloLens domain, which is more challenging if compared to GoPro, analyzing the impact of the placement of the discrim-

Table 4.4: Ablation study about the impact of each discriminator D_i and comparison between each discriminator D_i placed at C_i and P_i level.

Model	C_3	P_3	C_4	P_4	C_5	P_5	mAP
RetinaNet							14.10%
DA-RetinaNet					✓		15.84%
MDA-RetinaNet					✓		19.54%
MDA-RetinaNet						✓	16.29%
DA-RetinaNet			✓				16.38%
MDA-RetinaNet			✓				19.88%
MDA-RetinaNet				✓			17.01%
DA-RetinaNet	✓						28.61%
MDA-RetinaNet	✓						**34.97%**
MDA-RetinaNet		✓					31.44%
DA-RetinaNet	✓		✓				30.52%
MDA-RetinaNet	✓		✓				34.09%
MDA-RetinaNet		✓		✓			30.85%
DA-RetinaNet	✓		✓		✓		31.04%
MDA-RetinaNet	✓		✓		✓		32.11%
MDA-RetinaNet		✓		✓		✓	30.18%

inator at different levels of the feature map extracted from the RetinaNet backbone (see Figure 4.5). As can be noted, each single discriminator increases the performances of the standard RetinaNet architecture and obtain better performances than DA-RetinaNet. The discriminator attached to the first feature map C_3, allows to achieve better results than the other two discriminators attached to the C_4 and C_5 feature maps. Moreover, considering more than one discriminator to align the feature at different levels does

Table 4.5: Comparison performance considering different threshold.

Model	Threshold	Test H	Test G
MDA-RetinaNet + ST	0.90	47.48%	52.25%
MDA-RetinaNet + ST	0.85 to 90	49.21%	54.90%
MDA-RetinaNet + ST	0.80 to 0.90	52.49%	57.67%
MDA-RetinaNet + ST	0.75 to 0.90	**54.36%**	**59.51%**

lead to obtain better results in our experiments as in the case of the single domain DA-RetinaNet but only decreases the performance. The best combination and optimal number of discriminators was found empirically and, as shown in Table 4.4, it is achieved using only one discriminator at the C_3 level. We hypothesize that considering more discriminators at the same time could unbalance the models training, obtaining features that are aligned but less effective for the main object detection task. In Table 4.4 we also report an ablation study of the impact of each discriminator attached at P_i or at C_i levels. As can be noted, in each case, the performances achieved by the models that use the discriminator at P_i levels are lower than their counterparts which use discriminators at the C_i levels. Table 4.5 shows the results obtained with different linear schedules of the values of the threshold. We noted that, due to the domain gap between source and target domains, it is convenient to use a low threshold in the first iterations of self-training, where a set of initial pseudo-labels is needed, and increasing this threshold to an higher value as training proceeds. Indeed, we achieve best results for using a threshold value starting at 0.75 and ending at 0.9. Table 4.6 reports the results of adapting DA-Faster RCNN [45] and Strong Weak [29] to multiple target domains using the same methodology proposed for MDA-RetinaNet. Specifically, instead of merging the to dataset into one and use the binary

Table 4.6: Comparison between DA-Faster RCNN, Strong Weak and MDA-RetinaNet when modified using multiclass discriminators. S refers to Synthetic, H refers to Hololens and G to GoPro.

Model	Source	Target	Test H	Test G
DA-Faster RCNN [45]	S	H, G	13.79%	48.35%
Strong Weak [29]	S	H, G	29.52%	49.06%
MDA-RetinaNet	S	H, G	**34.97%**	**50.81%**

discriminator proposed by the authors in their papers, we replaced it with our multi classes discriminator and considered the target domains individually instead of merging them. As can be noted, the performances of the other two methods improves by 3-4% if compared with the results of Table 4.2. Nevertheless, the best results are still obtained by the proposed MDA-RetinaNet architecture. These results suggest that using a multi class discriminator instead of a binary discriminator allows to consistently improve performances with different architectures.

4.4.4 Comparison between MDA-RetinaNet and DA-RetinaNet

Table 4.7 compares the results of the proposed MDA-RetinaNet with DA-RetinaNet. It is worth noting that training the model using only one target domain at a time results in worse performance in both domains despite they are very similar. This happens because the model overfits with respect to the considered target domain used for training. Using both domains during training, as the proposed MDA-RetinaNet model does, allows to generalize over both target domains with a single model, which also results in improved performance.

Table 4.7: Comparison between DA-RetinaNet trained using one target set at a time and MDA-RetinaNet. S refers to Synthetic, H refers to Hololens and G to GoPro.

Model	Source	Target	Test H	Test G
RetinaNet [42]	S	-	14.10%	37.13%
DA-RetinaNet [101]	S	H	31.01%	36.60%
DA-RetinaNet [101]	S	G	21.63%	45.86%
MDA-RetinaNet	S	H, G	**34.97%**	**50.81%**

4.4.5 Qualitative Results

Figure 4.7 compares some qualitative detection results obtained by the proposed MDA-RetinaNet with and without Self-Training with respect to RetinaNet baseline (the ground truth is the blue bounding box). RetinaNet fails the detection in many cases. Indeed, it does not detects any artwork or produce a wrong classification and/or regression. MDA-RetinaNet well recognize small and large artworks but fails in the last two rows. MDA-RetinaNet with Self-Training improve the performance of the standard RetinaNet and MDA-RetinaNet with a more accurate detection of the artworks.

4.4.6 Results on Cityscapes Dataset

In order to thoroughly assess the generalization capabilities of the proposed methods across different datasets, we conducted experiments using the widely-used Cityscapes dataset [93], specifically targeting the foggy [94] and rainy conditions as domains of interest. To generate the rainy dataset, we followed the procedure proposed by [100], starting from the original Cityscapes dataset. Each dataset consisted of 2975 training images and 500 validation images. Table 4.8 presents the results obtained on the validation set using

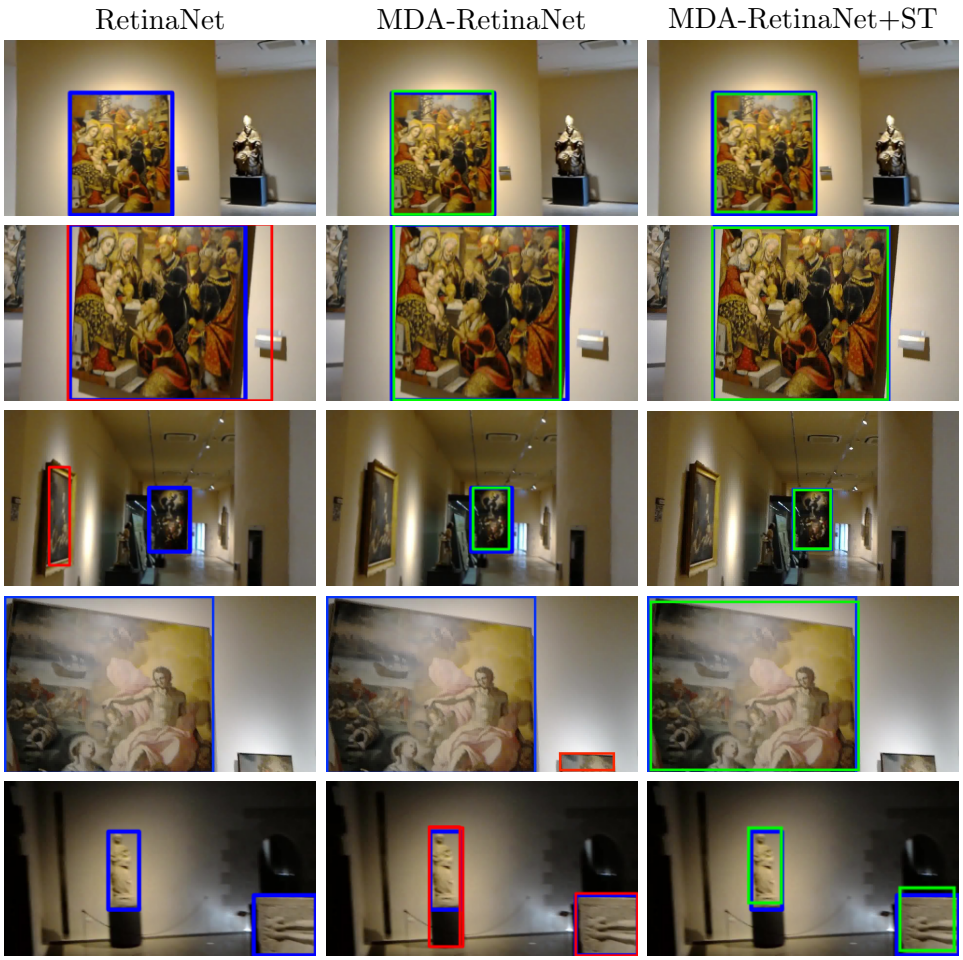


Figure 4.7: Qualitative results of RetinaNet, MDA-RetinaNet and MDA-RetinaNet with self-training (ST). The blue box represents ground truth, the red box indicates a wrong detection (object localization or classification), the green box represents correct detections.

standard object detector architectures and domain adaptation methods based on feature alignment. The table demonstrates that the standard RetinaNet outperforms DA-Faster RCNN, Strong-Weak, and CDSSL by approximately 1.5%. Furthermore, our proposed models, MDA-RetinaNet and (ST)MDA-RetinaNet, exhibit improvements of 1% and 2% respectively when compared to the best-performing model, DA-RetinaNet. Despite these advancements,

Table 4.8: Results adaptation between Cityscapes, Foggy and Rainy Cityscapes dataset.

Model	Fog	Rain
Faster RCNN [29]	23.54%	22.93%
DA-Faster RCNN [45]	30.10%	29.66%
Strong-Weak [29]	38.76%	37.81%
CDSSL [48]	39.34%	38.22%
RetinaNet [42]	40.69%	40.14%
DA-RetinaNet [101]	45.02%	42.90%
MDA-RetinaNet	46.17%	43.79%
STMDA-RetinaNet	**46.91%**	**45.37%**
Oracle	56.75%	55.58%

a discernible gap remains between the performance of the proposed architecture and the Oracle, which represents the performance achieved by training and testing RetinaNet on the Foggy and Rainy Cityscapes datasets. This observation suggests that there is still potential for further improvements and enhancements in our approach.

4.5 Conclusion

We studied the problem of unsupervised multi-camera domain adaptation for object detection in cultural sites. To perform the study, we have collected and publicly released a new challenging dataset with the aim to encourage the community to continue researching on the problem. In order to more comprehensively evaluate the generalization capabilities of the compared approaches, we additionally conducted experiments using a dataset

specifically designed for urban environments. We proposed a new method which combines feature alignment, pixel level and self-training methods that outperforms current state-of-the-art methods.

Chapter 5

Unsupervised Domain Adaptation for Action Recognition

5.1 Introduction

This chapter presents a preliminary study of the problem of unsupervised domain adaptation (UDA) for action recognition, examining both first-person and third-person points of view. The study utilizes the Epic Kitchens 55 dataset for the first-person and the UFC-HMDB datasets for the third-person perspective. To investigate the problem, we select C2D [102], I3D [61], and MViT [69] as representatives of the three main categories of action recognition algorithms (2D CNNs, 3D CNNs, and transformers) with the goal of analyzing the diverse performances associated with each architecture. We specifically focus on four domain adaptation strategies, treating them as individual "blocks" that can be integrated independently or in combination as shown in figure 5.1. The primary objective of this investigation is to gain

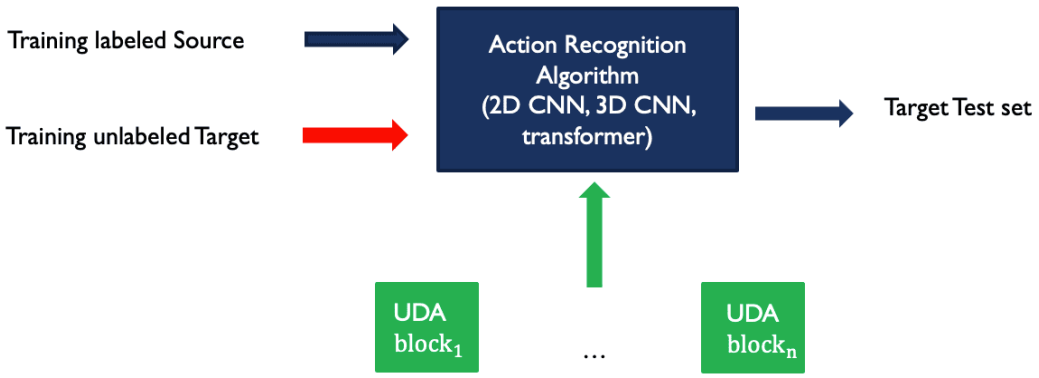


Figure 5.1: Scheme of the integration of one or multiple domain adaptation methods into an action recognition algorithm (2D CNN, 3D CNN or transformers).

valuable insights into the impact of these domain adaptation strategies when applied to different action recognition algorithms. By studying their behavior and effectiveness, we aim to identify the most suitable and effective strategies that lead to improved performance in the challenging domain adaptation setting for both first and third person point of view.

5.2 Methods

In this section, we present the methods used for comparison, including baseline approaches, domain adaptation approaches, and combined domain adaptation strategies.

Baseline Approaches without Adaptation: We establish baseline models without employing any domain adaptation techniques. These baselines include models trained on source images and tested on target images (no adaptation), as well as models trained and tested within the target domain (referred to as "oracle" models, which have access to target domain labels). By exploring these baselines, we can quantify the performance loss due to domain shift. The three action recognition algorithms used for the experi-

ments are C2D [102], I3D [61] and MViT [69]. The results show that higher-performing action recognition models tend to maintain their advantage in the target domain even without domain adaptation, indicating their robustness to domain shift on the evaluated datasets.

Domain Adaptation Approaches: Various methods are employed for unsupervised domain adaptation in action recognition to reduce the domain gap between the source and target domains. The following methods are considered for our analysis: 1) Feature Alignment: this approach employs adversarial learning using the gradient reversal layer [13] to align feature distributions between source and target domains. 2) Temporal Alignment: a temporal module [77] is used to produce relation feature representations, enabling temporal alignment between the source and target domains. 3) Self-Supervised Learning: contrastive learning [103] and pseudo-labeling [104] are utilized as self-supervised learning methods for domain adaptation.

Combined Domain Adaptation Approaches: The effectiveness of combining the individual domain adaptation methods is examined. By evaluating the performance of these combined strategies, we gain insights into potential synergies among them and their overall impact.

5.3 Experimental Settings and Results

In this section, we present and analyze the results obtained from the compared methods. The models were trained using the PyTorch Video [105] implementation, with Adam as the optimizer and a learning rate set to 0.001. All other parameters were set to their default values. We report the top-1 accuracy for action recognition, and for the Epic Kitchens dataset we provide the average accuracy obtained across the three domains (D1, D2 and D3).

5.3.1 Datasets

UCF-HMDB Dataset: The UCF-HMDB dataset combines action classes from two popular third-person action recognition datasets: UCF [106] and HMDB [107]. The dataset contains 12 action classes selected by overlapping them between the two datasets ensuring their presence in both UCF and HMDB. The dataset consists of 3209 videos, with 1438 training videos and 571 validation videos from UCF, and 840 training videos and 360 validation videos from HMDB. For domain adaptation experiments, UFC is used as the source domain, and HMDB is used as the target domain.

Epic-Kitchens55 Dataset: The Epic-Kitchens55 is an egocentric action recognition dataset that captures daily activities in kitchens from a first-person perspective. The dataset contains videos recorded by people wearing head-mounted cameras while performing kitchen-related tasks. The authors of [87] adapted it for domain adaptation tasks by creating three domains based on the eight largest action classes present in the dataset, corresponding to P08, P01 and P22 kitchens. Domain adaptation experiments with Epic-Kitchens aim to assess the models' generalization ability from one participant's kitchen (source domain) to another participant's kitchen (target domain).

5.3.2 Baseline Results

Table 5.1 presents the baseline results of C2D, I3D, and MViT models trained on the source domain and tested on the target domain, along with the oracle results representing upper bound performance achieved within each dataset. As can be observed, all the models drop in performance due to domain shift when compared with the results obtained by the oracle models. Specifically,

Table 5.1: Baseline results of C2D, I3D, and MViT when trained on the source and tested on the target datasets. The table also includes the oracle results, which represent the upper bound performance achieved in each dataset.

Model	EK55	UFC → HMDB
C2D	32.81%	76.70%
I3D	35.49%	80.23%
MViT	**38.98%**	**85.29%**
C2D (oracle)	58.88%	93.44%
I3D (oracle)	60.55%	95.00%
MViT (oracle)	**63.13%**	**96.12%**

on the Epic Kitchens dataset, the performances decrease by approximately 24% compared to the oracle models (32.81% vs 57.88%, 35.49% vs 59.55%, 38.98% vs 62.13%). Similarly, on the UFC → HMDB datasets, the I3D (80.23% vs 95.00%) and MViT (85.29% vs 96.12%) models show less sensitivity to the domain shift compared to the C2D model (76.70% vs 93.44%). These results highlight that action recognition models with superior performance in their source domains maintain an advantage over less performing models even in the presence of domain shift.

5.3.3 Domain Adaptation Results

Table 5.2 presents the performance of the C2D, I3D and MViT models with different domain adaptation methods. The methods are denoted by L_F, L_T, L_{ST} and L_C, representing respectively: feature alignment, temporal alignment, self-training with pseudo labels and contrastive learning. For the Epic Kitchens 55 dataset (EK55), domain adaptation using feature alignment

Table 5.2: Performance of C2D, I3D, and MViT models with domain adaptation methods. L_F, L_T, L_{ST}, and L_C respectively represent feature alignment, temporal alignment, self-training with pseudo labels, and contrastive learning.

Model	L_F	L_T	L_{ST}	L_C	EK55	UFC → HMDB
C2D	✓				34.79%	77.12%
C2D		✓			**35.10%**	**77.26%**
C2D			✓		32.46%	75.90%
C2D				✓	33.38%	77.08%
I3D	✓				38.91%	80.63%
I3D		✓			**39.24%**	**80.81%**
I3D			✓		35.57%	80.36%
I3D				✓	36.22%	80.59%
MViT	✓				40.65%	86.28%
MViT		✓			**41.18%**	**86.72%**
MViT			✓		39.02%	85.37%
MViT				✓	39.55%	85.43%

(C2D with L_F) or temporal alignment results in an accuracy improvement of 34.79% and 35.10% compared to the baseline C2D model without domain adaptation (32.81%). On the other side, applying contrastive learning (C2D with L_C) or self training (C2D with L_{ST}) individually leads respectively minor accuracy gains of 33.38% and a slight drop in performance 32.46%. On the UFC → HMDB datasets, C2D with feature alignment (L_F) achieves an accuracy of 77.12%, while temporal alignment (L_T) and self-training (L_{ST}) show accuracies of 77.26% and 75.90%, respectively. C2D combined with contrastive learning (L_C) attains an accuracy of 77.08%. As can be noted, for all three models (C2D, I3D, and MViT), the domain adaptation meth-

ods have a similar behavior keeping their architectural advantage unchanged regardless of the adaptation method used. The same happens when we compare the results between the first (EK55) and third person dataset (UFC $\rightarrow$ HMDB).

Table 5.3 reports the results of integrating two domain adaptation methods simultaneously into C2D, I3D, and MViT models. The table shows that regardless of the specific combination employed, all methods exhibit an enhancement in their performance compared to the results reported in Table 5.2. For instance, I3D with L_F improves from 38.91% (reported in Table 5.2) to 40.77% when combined with L_{ST}. All the models exhibit a common pattern by achieving their optimal outcomes through the integration of feature alignment (L_F) and temporal alignment (L_T), a trend consistently observed across both datasets.

Table 5.4 reports the results of deploying the top three combined methodologies alongside a comprehensive integration of all UDA methods. As the tabled shows, incorporating three strategies consistently improves the performance of all considered action recognition models compared to Table 5.2 and Table 5.3. For instance, MViT's accuracy increases from 44.36% (when combining L_F and L_T in Table 5.3) to 45.27% with the addition of L_C. However, an observation emerges when considering the third strategy. Contrary to expectations, introducing the third strategy doesn't necessarily lead to superior performance across both first-person and third-person actions. To illustrate, for the I3D model, the most optimal result is attained in first-person actions by employing contrastive learning, while in third-person actions, self-training takes the lead. In the broader scope, it becomes evident that selecting between contrastive learning and pseudo-labeling methods for performance enhancement doesn't exhibit a clear-cut advantage. Addition-

Table 5.3: Performance of C2D, I3D, and MViT models combined with 2 domain adaptation methods at time. L_F, L_T, L_{ST}, and L_C respectively represent feature alignment, temporal alignment, self-training with pseudo labels, and contrastive learning.

Model	L_F	L_T	L_{ST}	L_C	EK55	UFC → HMDB
C2D	✓	✓			**37.17%**	**78.91%**
C2D	✓		✓		36.59%	78.08%
C2D	✓			✓	35.31%	77.43%
C2D		✓	✓		36.98%	78.70%
C2D		✓		✓	36.74%	78.25%
C2D			✓	✓	34.66%	77.31%
I3D	✓	✓			**42.41%**	**81.80%**
I3D	✓		✓		40.77%	81.44%
I3D	✓			✓	39.65%	81.19%
I3D		✓	✓		41.83%	81.63%
I3D		✓		✓	41.56%	81.58%
I3D			✓	✓	38.44%	80.96%
MViT	✓	✓			**44.36%**	**90.02%**
MViT	✓		✓		42.78%	88.47%
MViT	✓			✓	42.59%	88.36%
MViT		✓	✓		43.88%	89.25%
MViT		✓		✓	43.64%	88.90%
MViT			✓	✓	41.22%	87.71%

ally, the application of all four UDA methods to the models doesn't yield significant improvements. This observation could be attributed to the intricate challenge of striking the right balance between multiple losses within

Table 5.4: Performance of C2D, I3D and MViT combined with the top 3 methodology and with the four methods. L_F, L_T, L_{ST}, and L_C respectively represent feature alignment, temporal alignment, self-training with pseudo labels, and contrastive learning.

Model	L_F	L_T	L_{ST}	L_C	EK55	UFC → HMDB
C2D	✓	✓		✓	38.81%	79.95%
C2D	✓	✓	✓		**39.06%**	79.77%
C2D	✓	✓	✓	✓	38.52%	**80.15%**
I3D	✓	✓		✓	**43.70%**	82.58%
I3D	✓	✓	✓		43.39%	**82.84%**
I3D	✓	✓	✓	✓	43.56%	82.69%
MViT	✓	✓		✓	45.27%	91.64%
MViT	✓	✓	✓		**45.43%**	**91.70%**
MViT	✓	✓	✓	✓	45.34%	91.52%

the UDA framework, making complex the optimal adjustments.

5.4 Conclusion

We conducted a preliminary study on unsupervised domain adaptation for action recognition, focusing on both first-person and third-person viewpoints. Our investigation involved the most representative action recognition algorithms from each category: C2D (2D CNN), I3D (3D CNN), and MViT (transformer). The aim was to analyze the efficacy of various UDA strategies across different perspectives and algorithms. By dissecting UDA into four distinct strategies, we delineated their individual and collective influences on model performance. Our experimental results unveiled several noteworthy

observations. Firstly, the combination of multiple UDA methods consistently demonstrated improved performance over employing a single method, indicative of potential synergies between these strategies. Secondly, regardless of the UDA strategy or combination employed, the performance hierarchy of the three action recognition algorithms (2D CNN, 3D CNN, and transformers) remained fairly consistent.

Future work could extend the analysis to other UDA methods, exploring the possibility of combining additional strategies to improve performance in both first and third-person action scenarios.

Chapter 6

Conclusion

In this thesis we tackled the problem of unsupervised domain adaptation for object detection and action recognition tasks, which plays an important role in enhancing the robustness and generalization capabilities of deep learning models in computer vision.

Chapter 1 provided a comprehensive introduction to the unsupervised domain adaptation problem, discussing its applications in various computer vision tasks.

In Chapter 2, we delved into the background concepts, presenting the problem's formulation and an overview of state-of-the-art methods in object detection and action recognition. These foundational insights set the stage for our subsequent contributions.

In Chapter 3, our focus on unsupervised domain adaptation for object detection resulted in the creation of the UDA-CH dataset and the development of the DA-RetinaNet algorithm, specifically tailored to address challenges within cultural heritage and autonomous driving scenarios.

Chapter 4 extended our study to multiple target domains by incorporating GoPro images into the UDA-CH dataset. The (ST)MDA-RetinaNet

model showcased the potential of combining adversarial learning, image-to-image translation, and self-training approaches, demonstrating adaptability in complex scenarios.

In Chapter 5, we explored the action recognition, addressing both first-person and third-person viewpoints. Through a comprehensive analysis of UDA strategies using various algorithms, we uncovered valuable insights into the effectiveness of combining multiple methods to enhance recognition performance.

While our proposed algorithms, datasets, and insights contribute to the field, it is important to acknowledge the dynamic nature of unsupervised domain adaptation. Challenges such as selecting appropriate adaptation strategies, understanding the effects of diverse data distributions, and addressing complex scenarios persist. The individual chapter conclusions provide specific details on our contributions, such as the creation of datasets, the development of the novel algorithms, and the exploration of action recognition strategies. These findings collectively advance the understanding of unsupervised domain adaptation in computer vision. As the field continues to evolve, future research could focus on refining existing algorithms, exploring novel adaptation techniques, and investigating real-world deployment scenarios to enhance the practical applicability of UDA solutions. By releasing the code of the proposed algorithms and datasets, this thesis aims to facilitate further research in the field.

Appendix A

GAN-Driven protection of Medical Imagery against Malicious Tampering

A.1 Introduction

In recent years, advancements in generative models have ushered in a new era of image generation and manipulation, showcasing remarkable capabilities in rendering images increasingly indistinguishable from their original counterparts [27, 108, 109]. This progress, driven by deep learning techniques, has found applications in various domains, from creative artistry [110] to medical imaging [111, 112, 113, 114, 115], among others. However, alongside positive applications, researchers have demonstrated the malicious use of Generative Adversarial Networks (GANs) for tasks such as malware obfuscation [116] and the creation of deepfakes [117].

Within the medical domain, the potential consequences of malicious tampering are critical, as the integrity and authenticity of images can have life-or-

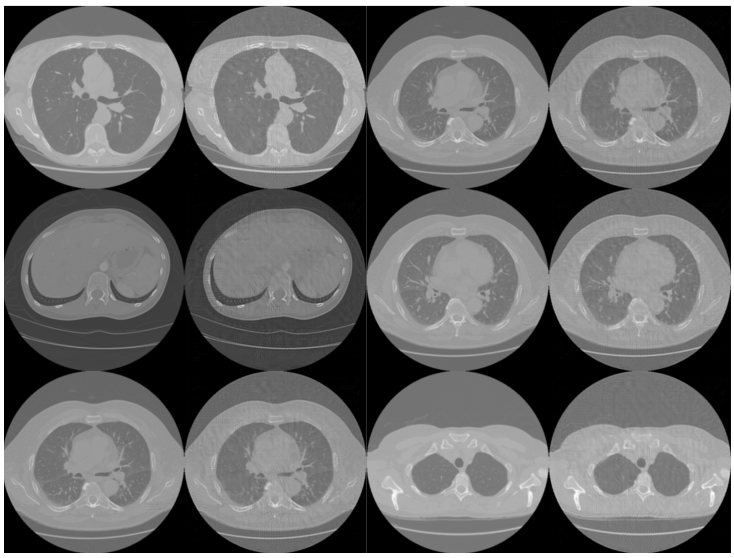


Figure A.1: Qualitative results of TAFIM [123] applied to CT scans. Columns 1 and 3 show input images, while columns 2 and 4 show the protected images. Note the visible artifacts in the protected images, which may pose challenges during analysis by medical experts.

death implications. Image tampering techniques [118] have raised concerns by highlighting the potential for malicious manipulation of medical images, such as computed tomography (CT) scans and radiographs. This introduces a new dimension of cyber attacks, with image manipulation being employed to deceive medical professionals and compromise patient care, potentially leading to misdiagnoses.

To address this challenge, the research community has focused on developing automated detection systems for image manipulation, treating it as a classification task. Various learning-based approaches have shown promise, achieving excellent classification accuracy [119, 120, 121, 122].

Alternatively, another strategy is to prevent manipulations at the source by disrupting manipulation methods' output [124, 125, 123]. The key idea is to disrupt generative neural network models by introducing noise patterns

at a low level, making it more challenging for malicious actors to create convincing forgeries.

In this study, we investigate the problem of image tampering in the medical domain, focusing on the manipulation of CT scans. Building upon the idea presented by the authors of TAFIM [123], we propose MITS-GAN (Medical Imaging Tamper Safe-GAN), an approach based on Generative Adversarial Networks. Our method generates tamper-resistant images, minimizing potential artifacts (Figure A.1) that could pose challenges during the review process by medical experts.

A.2 Related Work

In this section, we briefly review research related to our work.

A.2.1 GAN Applications in Medical Imaging

GANs have significantly contributed to medical imaging by addressing challenges and enhancing the quality and accessibility of medical imagery. They have been employed for tasks such as data augmentation, style translation, and image generation in various medical imaging modalities [126, 127, 128, 129, 115, 114, 113, 112, 111]. Additionally, GANs have found applications in segmentation [130], super-resolution [131], and anomaly detection [132].

A.2.2 Adversarial Attacks

Adversarial attack methods aim to introduce imperceptible changes to images to disrupt neural network feature extraction. Initially applied in classification tasks [133, 134, 135], these methods have been extended to segmentation [136] and detection tasks [137]. Generic universal image-agnostic noise

patterns have been proposed to address the challenge of time-consuming, image-specific pattern optimization [138, 139]. However, such approaches have limitations when applied to generative models.

A.2.3 Image Manipulation Prevention

Preventing image manipulations by disrupting adversarial attack techniques has been explored as an alternative to classification and detection. Methods like disrupting deepfakes [125], nullifying image-to-image translation models [124, 140], and generating image-specific patterns for low-resolution images [141] have been proposed.

A.3 Proposed Method

Our goal is to prevent image manipulation, such as adding or removing tumors in CT scans, by disrupting the CT-GAN [118] architecture. We introduce an imperceptible perturbation that disrupts the CT-GAN's output, making it easier for a human to identify tampered scans.

A.3.1 Method Overview

The proposed method is illustrated in Figure A.2. Given a CT scan X_s and a fixed image-agnostic perturbation shared across the data distribution δ_S, they are concatenated channel-wise and then input into the generator G. The generator applies 5 2D convolutional layers to the perturbation δ_S, with each followed by batch normalization and the ReLU activation function. Subsequently, δ_S is concatenated with the image, and finally, a sequence of a 2D convolution, 3 residual blocks, and a final 2D convolution are applied in succession. The output, denoted as X_P, represents the protected scan and

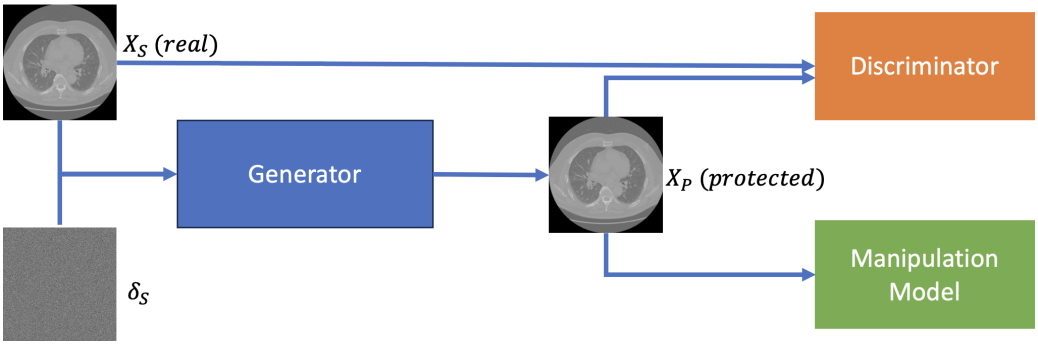


Figure A.2: Model Architecture Overview: The Generator receives the input image X_S and perturbation noise δ_S to produce the protected image X_P. Subsequently, X_P is forwarded to the manipulation model and discriminator.

is fed into the CT-GAN model M and, along with X_S, to the discriminator D. The discriminator D consists of 8 2D convolutional layers, each followed by batch normalization and LeakyReLU activation function. The model is trained using a generative adversarial objective, encouraging the generator to produce protected images similar to the original (unprotected) ones. The model is also augmented with the CT-GAN.

The goal is to optimize the following min-max objective:

$$\min_{G} \max_{D,M} L_g(D, G) + \alpha L_m(G, M) \tag{A.1}$$

where L_g represents the generator and discriminator GAN losses, L_m is mean squared error (MSE) loss computed between the output of the model M and the generator G, and α is the weight that controls the interaction of these losses.

A.4 Results

In this section, we present and analyze the results of the introduced methodologies. Our approach is assessed using the dataset outlined in [142], following

the training editing procedure detailed in [118]. We use respectively 300 and 50 scans for the training set and test set. To evaluate the output quality, we compute the RMSE, PSNR and LPIPS [143] metrics.

A.4.1 Experimental Setup

All models were trained for 20 epochs using an NVIDIA V100. The MITS-GAN [1] architecture, implemented using PyTorch [2], was trained with a batch size of 16, a learning rate set at 0.0002, betas of $[0.5, 0.999]$, and utilizing Adam as the optimizer. For TAFIM, we adopted the configurations suggested by the authors in [123].

A.4.2 Qualitative Results

Figure A.3 shows the qualitative results of the proposed MITS-GAN method compared with TAFIM. MITS-GAN exhibits fewer visible artifacts on the reconstructed images and demonstrates a more robust ability to resist manipulation, accentuating the artifacts introduced when the model attempts to manipulate the selected square. Figure A.4 shows the heatmap obtained by performing a pixel-to-pixel difference between the real image and the protected one. Also in this case, the proposed method generates protected images that are more faithful to the originals than the compared method.

A.4.3 Quantitative Results

Table A.1 reports the results of the considered metrics evaluated between each pair of real-protected and real-protected/tampered on the entire im-

[1] https://github.com/GiovanniPasq/MITS-GAN

[2] https://pytorch.org/

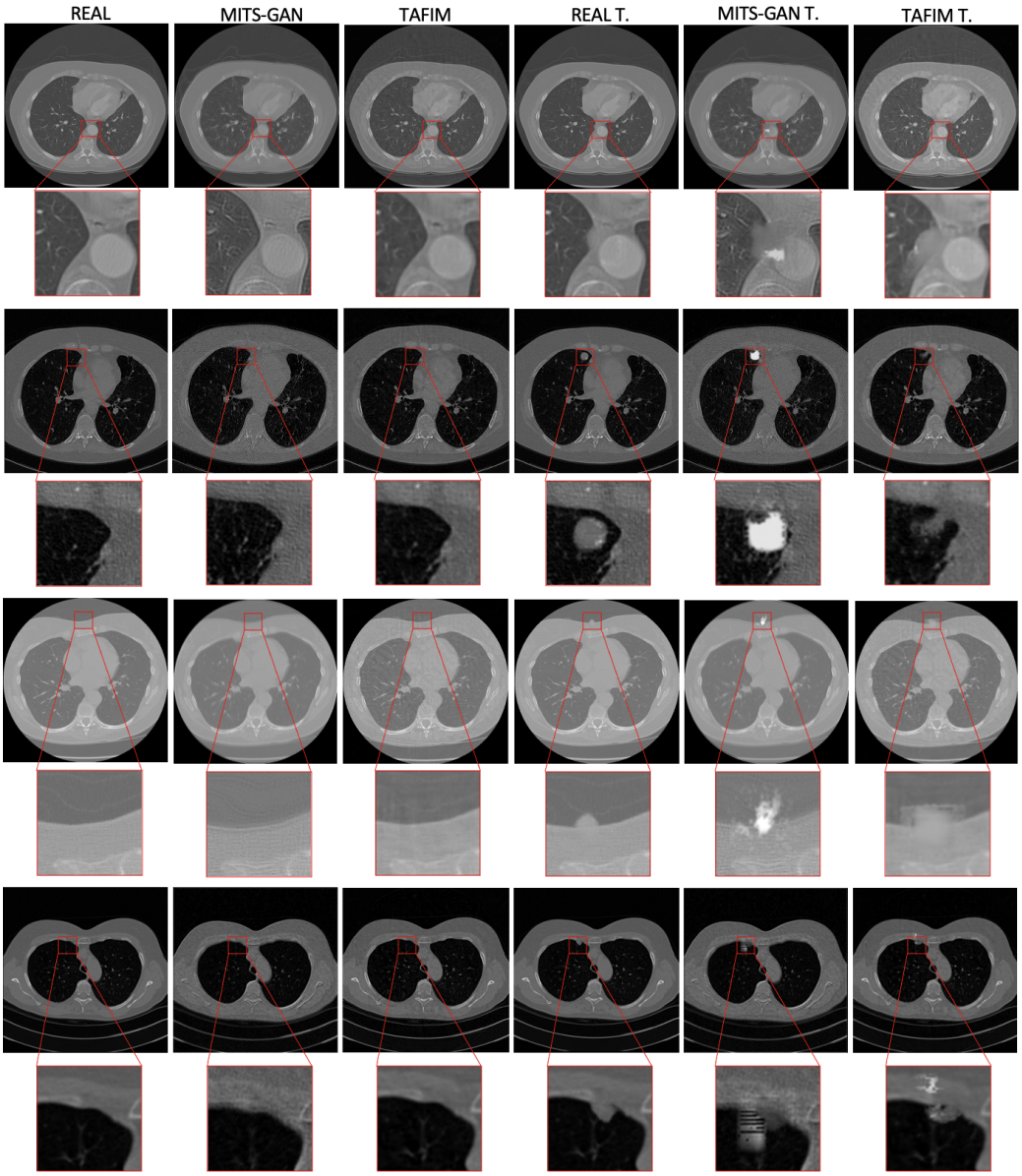


Figure A.3: Qualitative results on the reconstruction task compared with images as manipulation targets.

ages. MITS-GAN has lower RMSE, LPIPS, and higher PSNR values compared to TAFIM, suggesting better reconstruction quality of the images. This advantage is maintained even when considering the images after manipula-

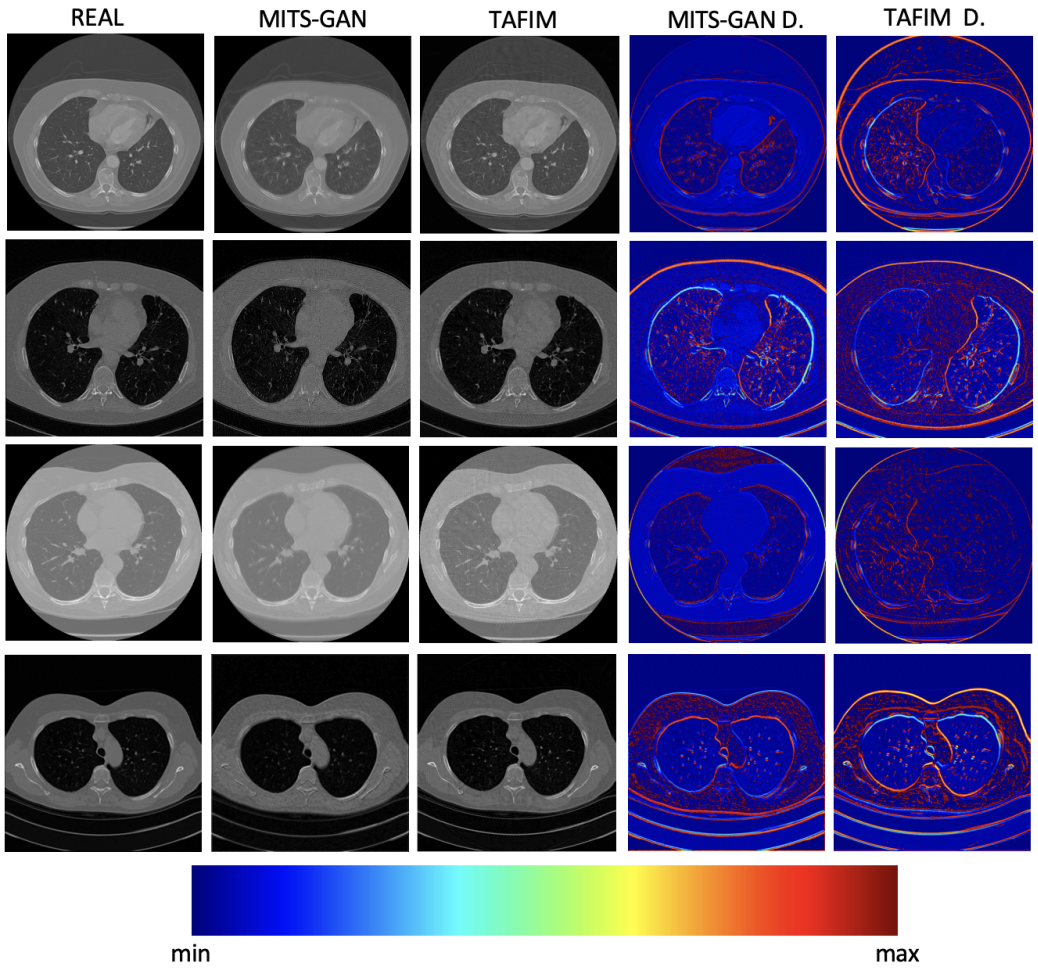


Figure A.4: Heatmap computed between the pairs: real-MITS-GAN and real-TAFIM.

tion. Table A.2 shows the results evaluated on the square part subjected to manipulation. In this case, the metrics favor the proposed method. After manipulation, the output produced by the manipulator model appears to be more damaged than the compared method. This suggests that MITS-GAN produces images with less noise but is more robust to manipulation, generating more visible artifacts when attempting to tamper with an image.

Table A.1: Metric results evaluated between the following pairs on the entire images: real-MITS-GAN, real-TAFIM, real-MITS-GAN tampered, and real-TAFIM tampered. Lower values are better for RMSE and LPIPS, higher for PSNR.

Metric	MITS-GAN	TAFIM	MITS-GAN T.	TAFIM T.
RMSE	169.481	194.943	198.253	233.780
PSNR	27.949	21.702	21.237	21.469
LPIPS	0.170	0.383	0.226	0.391

Table A.2: Metric results evaluated between the following pairs on the tampered square part of the images: real-MITS-GAN, real-TAFIM, real-MITS-GAN tampered, and real-TAFIM tampered. Lower values are better for RMSE and LPIPS, higher for PSNR.

Metric	MITS-GAN	TAFIM	MITS-GAN T.	TAFIM T.
RMSE	50.565	66.061	84.349	79.451
PSNR	26.682	18.854	11.289	18.511
LPIPS	0.372	0.3417	0.591	0.346

A.5 Conclusion

In this work, we introduced MITS-GAN a novel approach utilizing Generative Adversarial Networks to safeguard medical imagery from malicious tampering. The proposed method effectively disrupts manipulations at the source, generating tamper-resistant images with fewer artifacts compared to existing technique. Experimental results demonstrate the superior performance of MITS-GAN, highlighting its potential to enhance the security and integrity of medical scans. As the field evolves, proactive measures like these are crucial to ensure responsible and ethical use of generative models, espe-

cially in high-stakes applications such as healthcare. Further research and collaboration will be key to advancing these methodologies and addressing emerging cyber threats in medical imaging.

Bibliography

[1] Guangchen Luo, Yaping Zhang, Li Zhang, Shengjie Wang, and Wei Hu. Real-time action recognition in video surveillance using deep learning approach. In *International Conference on Smart Computing & Electronic Enterprise*, pages 73–80. IEEE, 2018.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[3] David Fitzgerald, Neil O'Hare, and Ciarán Ó Conaire. Single-shot object detection for augmented reality interaction with physical objects. *IEEE Access*, 6:2151–2162, 2018.

[4] Li Wang and Tao Gu. Human action recognition for healthcare applications. *Journal of Ambient Intelligence and Humanized Computing*, 7(6):881–895, 2016.

[5] Kechao Zhou, Yuqiong Wang, Zhihang Lu, and Honggang Zhang. Mobile augmented reality using object recognition. In *International Conference on Mobile Ad-Hoc and Sensor Networks*, pages 270–276. IEEE, 2014.

[6] Roland Schwarz, Markus Fuchs, and Jürgen Beyerer. Development of a wearable assistance system for visually impaired based on object detection and voice feedback. In *International Conference on Applied Informatics*, pages 331–338. IEEE, 2016.

[7] Congcong Wang, Yan Chen, Chaoyue Liu, Cheng Liu, and Fei Wu. Real-time human activity recognition on smartphones using deep learning. In *International Conference on Information and Automation*, pages 1207–1212. IEEE, 2019.

[8] Alexandre Blanchard, Alexandre Allard, Caroline Lombriser, Mattia Bertschi, Elena Mugellini, and Patrick Salamin. Real-time exercise recognition on wearable sensors. *Sensors*, 18(5):1633, 2018.

[9] Santi Andrea Orlando, Antonino Furnari, and Giovanni Maria Farinella. Egocentric visitor localization and artwork detection in cultural sites using synthetic data. *Pattern Recognition Letters*, 133:17–24, 2020.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio M Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2017.

[11] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. Synthia: A highly detailed synthetic scene generation framework for rgb-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(8):1537–1553, 2016.

[12] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189, 2015.

[14] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[15] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):801–814, 2018.

[16] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[17] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[18] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[19] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *The Journal of Machine Learning Research*, 2, 01 2002.

[20] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer, 2017.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[22] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *CVPR*, 2021.

[23] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964. IEEE, 2019.

[24] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[25] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Advances in neural information processing systems*, pages 1287–1298, 2018.

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.

[27] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[28] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998, 2018.

[29] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.

[30] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019.

[31] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

[32] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In

Proceedings of the IEEE International Conference on Computer Vision, pages 1406–1415, 2019.

[33] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020.

[34] Mohamed Ragab, Zhenghua Chen, Min Wu, Haoliang Li, Chee-Keong Kwoh, Ruqiang Yan, and Xiaoli Li. Adversarial multiple-target domain adaptation for fault classification. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2020.

[35] Le Thanh Nguyen-Meidine, Atif Belal, Madhu Kiran, Jose Dolz, Louis-Antoine Blais-Morin, and Eric Granger. Unsupervised multi-target domain adaptation through knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1339–1347, 2021.

[36] Soumya Varma and M Sreeraj. Object detection and classification in surveillance system. In *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 299–303. IEEE, 2013.

[37] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2847–2854. IEEE, 2012.

[38] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[40] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[41] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[42] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[43] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[45] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[46] Rongchang Xie, Fei Yu, Jiachao Wang, Yizhou Wang, and Li Zhang. Multi-level domain adaptive learning for cross-domain detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[47] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019.

[48] F. Yu, Di Wang, Yinpeng Chen, Nikolaos Karianakis, Pei Yu, D. Lymberopoulos, and X. Chen. Unsupervised domain adaptation for object detection via cross-domain semi-supervised learning. *ArXiv*, abs/1911.07158, 2019.

[49] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6091–6100, 2019.

[50] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 738–746, 2020.

[51] Kazuma Fujii and Kazuhiko Kawamoto. Generative and self-supervised domain adaptation for one-stage object detection. *Array*, 11:100071, 2021.

[52] Chaoqi Chen, Zebiao Zheng, Yue Huang, Xinghao Ding, and Yizhou Yu. I3net: Implicit instance-invariant network for adapting one-stage object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[53] Vidit and M. Salzmann. Attention-based domain adaptation for single stage detectors. *ArXiv*, abs/2106.07283, 2021.

[54] Dayan Guan, Jiaxing Huang, Aoran Xiao, Shijian Lu, and Yanpeng Cao. Uncertainty-aware unsupervised domain adaptation in object detection. *ArXiv*, abs/2103.00236, 2021.

[55] V. Vibashan, Vikram Gupta, Poojan Oza, Vishwanath A. Sindagi, and V. Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *CVPR*, 2021.

[56] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[57] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2017.

[58] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[59] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module

for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[60] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[61] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[62] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.

[63] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[64] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019.

[65] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[66] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[68] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[69] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[70] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021.

[71] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.

[72] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time

attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[73] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.

[74] Arshad Jamal, Vinay P. Namboodiri, Dipti Deodhare, and K. S. Venkatesh. Deep domain adaptation in action space. In *BMVC*, 2018.

[75] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[76] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[77] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6321–6330, 2019.

[78] Yadan Luo, Zi Huang, Zijian Wang, Zheng Zhang, and Mahsa Baktashmotlagh. Adversarial bipartite graph learning for video domain adaptation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 19–27, 2020.

[79] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *European Conference on Computer Vision*, pages 678–695. Springer, 2020.

[80] Aadarsh Sahoo, Rutav Shah, Rameswar Panda, Kate Saenko, and Abir Das. Contrast and mix: Temporal contrastive video domain adaptation with background mixing. *Advances in Neural Information Processing Systems*, 34:23386–23400, 2021.

[81] Peipeng Chen, Yuan Gao, and Andy J Ma. Multi-level attentive adversarial learning with temporal dilation for unsupervised video domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1259–1268, 2022.

[82] Victor G Turrisi da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. Dual-head contrastive domain adaptation for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1181–1190, 2022.

[83] Victor G Turrisi da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. Unsupervised domain adaptation for video transformers in action recognition. *arXiv preprint arXiv:2207.12842*, 2022.

[84] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[85] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[86] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11815–11822, 2020.

[87] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 122–132, 2020.

[88] Xiaolin Song, Sicheng Zhao, Jingyu Yang, Huanjing Yue, Pengfei Xu, Runbo Hu, and Hua Chai. Spatio-temporal contrastive domain adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9787–9795, 2021.

[89] Donghyun Kim, Yi-Hsuan Tsai, Bingbing Zhuang, Xiang Yu, Stan Sclaroff, Kate Saenko, and Manmohan Chandraker. Learning cross-modal contrastive features for video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13618–13627, 2021.

[90] Yunhua Zhang, Hazel Doughty, Ling Shao, and Cees GM Snoek. Audio-adaptive activity recognition across video domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13791–13800, 2022.

[91] Mirco Planamente, Chiara Plizzari, Emanuele Alberti, and Barbara Caputo. Domain generalization through audio-visual relative norm alignment in first person action recognition. In *Proceedings of the*

IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1807–1818, 2022.

[92] Lijin Yang, Yifei Huang, Yusuke Sugano, and Yoichi Sato. Interact before align: Leveraging cross-modal knowledge for domain adaptive action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14722–14732, 2022.

[93] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[94] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[95] Giovanni Maria Farinella Giovanni Pasqualino, Antonino Furnari. Unsupervised domain adaptation for object detection in cultural sites. In *International Conference on Pattern Recognition (ICPR)*, 2020.

[96] Francesco Ragusa, Antonino Furnari, Sebastiano Battiato, Giovanni Signorello, and Giovanni Maria Farinella. Ego-ch: Dataset and fundamental tasks for visitors behavioral understanding using egocentric vision. *Pattern Recognition Letters*, 131:150–157, 2020.

[97] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[98] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[99] 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.

[100] Vishwanath A Sindagi, Poojan Oza, Rajeev Yasarla, and Vishal M Patel. Prior-based domain adaptive object detection for hazy and rainy conditions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 763–780. Springer, 2020.

[101] Giovanni Pasqualino, Antonino Furnari, Giovanni Signorello, and Giovanni Maria Farinella. An unsupervised domain adaptation scheme for single-stage artwork recognition in cultural sites. *Image and Vision Computing*, 107:104098, 2021.

[102] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[103] Mamatha Thota and Georgios Leontidis. Contrastive domain adaptation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2209–2218, 2021.

[104] Giovanni Pasqualino, Antonino Furnari, and Giovanni Maria Farinella. A multi camera unsupervised domain adaptation pipeline for object detection in cultural sites through adversarial learning and self-training. *Computer Vision and Image Understanding*, page 103487, 2022.

[105] Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, and Christoph Feichtenhofer. PyTorchVideo: A deep learning library for video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. `https://pytorchvideo.org/`.

[106] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[107] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011.

[108] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[109] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[110] Sakib Shahriar. Gan computers generate arts? a survey on visual arts, music, and literary text generation using generative adversarial network. *Displays*, 73:102237, 2022.

[111] Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Chest x-ray generation and data augmentation for

cardiovascular abnormality classification. In *Medical imaging 2018: Image processing*, volume 10574, pages 415–420. SPIE, 2018.

[112] Jelmer M Wolterink, Tim Leiner, and Ivana Isgum. Blood vessel geometry synthesis using generative adversarial networks. *arXiv preprint arXiv:1804.04381*, 2018.

[113] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

[114] Camilo Bermudez, Andrew J Plassard, Larry T Davis, Allen T Newton, Susan M Resnick, and Bennett A Landman. Learning implicit brain mri manifolds with deep learning. In *Medical imaging 2018: Image processing*, volume 10574, pages 408–414. SPIE, 2018.

[115] Cheng-Bin Jin, Hakil Kim, Mingjie Liu, Wonmo Jung, Seongsu Joo, Eunsik Park, Young Saem Ahn, In Ho Han, Jae Il Lee, and Xuenan Cui. Deep ct to mr synthesis using paired and unpaired data. *Sensors*, 19(10):2361, 2019.

[116] Weiwei Hu and Ying Tan. Generating adversarial malware examples for black-box attacks based on gan. In *International Conference on Data Mining and Big Data*, pages 409–423. Springer, 2022.

[117] Bobby Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *Calif. L. Rev.*, 107:1753, 2019.

[118] Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. {CT-GAN}: Malicious tampering of 3d medical imagery using deep learning.

In *28th USENIX Security Symposium (USENIX Security 19)*, pages 461–478, 2019.

[119] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva. Id-reveal: Identity-aware deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15108–15117, 2021.

[120] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5001–5010, 2020.

[121] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311. IEEE, 2019.

[122] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 384–389. IEEE, 2018.

[123] Shivangi Aneja, Lev Markhasin, and Matthias Nießner. Tafim: Targeted adversarial attacks against facial image manipulations. In *European Conference on Computer Vision*, pages 58–75. Springer, 2022.

[124] Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. Disrupting image-translation-based deepfake algorithms with adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 53–62, 2020.

[125] Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 236–251. Springer, 2020.

[126] Lei Bi, Jinman Kim, Ashnil Kumar, Dagan Feng, and Michael Fulham. Synthesis of positron emission tomography (pet) images via multi-channel generative adversarial networks (gans). In *Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment: Fifth International Workshop, CMMI 2017, Second International Workshop, RAMBO 2017, and First International Workshop, SWITCH 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, 2017, Proceedings 5*, pages 43–51. Springer, 2017.

[127] Avi Ben-Cohen, Eyal Klang, Stephen P Raskin, Michal Marianne Amitai, and Hayit Greenspan. Virtual pet images from ct data using deep convolutional networks: initial results. In *Simulation and Synthesis in Medical Imaging: Second International Workshop, SASHIMI 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 10, 2017, Proceedings 2*, pages 49–57. Springer, 2017.

[128] Avi Ben-Cohen, Eyal Klang, Stephen P Raskin, Shelly Soffer, Simona Ben-Haim, Eli Konen, Michal Marianne Amitai, and Hayit Greenspan. Cross-modality synthesis from ct to pet using fcn and gan networks for improved automated lesion detection. *Engineering Applications of Artificial Intelligence*, 78:186–194, 2019.

[129] Qi Dou, Cheng Ouyang, Cheng Chen, Hao Chen, and Pheng-Ann

Heng. Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. *arXiv preprint arXiv:1804.10916*, 2018.

[130] Zhongyi Han, Benzheng Wei, Ashley Mercado, Stephanie Leung, and Shuo Li. Spine-gan: Semantic segmentation of multiple spinal structures. *Medical image analysis*, 50:23–35, 2018.

[131] Rohit Gupta, Anurag Sharma, and Anupam Kumar. Super-resolution using gans for medical imaging. *Procedia Computer Science*, 173:28–35, 2020.

[132] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[133] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[134] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[135] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[136] Volker Fischer, Mummadi Chaithanya Kumar, Jan Hendrik Metzen, and Thomas Brox. Adversarial examples for semantic image segmentation. *arXiv preprint arXiv:1703.01101*, 2017.

[137] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017.

[138] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 2755–2764, 2017.

[139] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[140] Chin-Yuan Yeh, Hsi-Wen Chen, Hong-Han Shuai, De-Nian Yang, and Ming-Syan Chen. Attack as the best defense: Nullifying image-to-image translation gans via limit-aware adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16188–16197, 2021.

[141] Qidong Huang, Jie Zhang, Wenbo Zhou, Weiming Zhang, and Nenghai Yu. Initiative defense against facial manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1619–1627, 2021.

[142] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[143] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.