IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 25, 2021, accepted March 6, 2021, date of publication March 15, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066041

Adversarial Reconstruction Loss for Domain Generalization

IMAD EDDINE IBRAHIM BEKKOUCH[1,2], DRAGOŞ CONSTANTIN NICOLAE[3], ADIL KHAN[2], (Member, IEEE), S. M. AHSAN KAZMI[4], ASAD MASOOD KHATTAK[5], (Senior Member, IEEE), AND BULAT IBRAGIMOV[2,6]

[1]Sorbonne Center for Artificial Intelligence, Sorbonne University, 75005 Paris, France
[2]Institute of Data Science and Artificial Intelligence, Innopolis University, 420500 Innopolis, Russia
[3]Institutul de Cercetări pentru Inteligenta Artificiala "Mihai Draganescu," Academia Romana, 050711 Bucharest, Romania
[4]Networks and Blockchain Laboratory, Innopolis University, 420500 Innopolis, Russia
[5]College of Technological Innovations, Zayed University, Abu Dhabi, United Arab Emirates
[6]Department of Computer Science, University of Copenhagen, 2100 Copenhagen, Denmark

Corresponding author: Adil Khan (a.khan@innopolis.ru)

This work was supported in part by the Foundation for Basic Research (RFBR) under Project 19-37-51034, and in part by the Zayed University Research Incentive Fund under Grant R19096.

ABSTRACT The biggest fear when deploying machine learning models to the real world is their ability to handle the new data. This problem is significant especially in medicine, where models trained on rich high-quality data extracted from large hospitals do not scale to small regional hospitals. One of the clinical challenges addressed in this work is magnetic resonance image generalization for improved visualization and diagnosis of hip abnormalities such as femoroacetabular impingement and dysplasia. Domain Generalization (DG) is a field in machine learning that tries to solve the model's dependency on the training data by leveraging many related but different data sources. We present a new method for DG that is both efficient and fast, unlike the most current state of art methods, which add a substantial computational burden making it hard to fine-tune. Our model trains an autoencoder setting on top of the classifier, but the encoder is trained on the adversarial reconstruction loss forcing it to forget style information while extracting features useful for classification. Our approach aims to force the encoder to generate domain-invariant representations that are still category informative by pushing it in both directions. Our method has proven universal and was validated on four different benchmarks for domain generalization, outperforming state of the art on RMNIST, VLCS and IXMAS with a 0.70% increase in accuracy and providing comparable results on PACS with a 0.02% difference. Our method was also evaluated for unsupervised domain adaptation and has shown to be quite an effective method against over-fitting.

INDEX TERMS Computer vision, deep learning, domain adaptation, domain generalization, transfer learning.

I. INTRODUCTION

Deep learning (DL) and Convolutional neural networks (CNN) empowered the computer vision field to be used in many situations efficiently and provide very promising results. Nowadays, all of our smart phones use facial recognition as an option for authentication with Federated Learning [1], and all new self-driven cars [2] are based mainly on a combination of deep CNNs for road image processing. This massive adoption raises the bar for computer vision systems to be more robust to edge cases and generalizes well in unforeseen situations. As useful as DL techniques are, deploying them and using them on real-world data brings some problems that we don't commonly see while working on toy datasets or training data in general [3], even if it was taken from previous users of the system. As powerful as they are, Deep Convolutional Networks showed a huge dependency problem on the data set they were trained on, commonly known as over-fitting [4]. This problem (called domain-shift [5] or concept drift [6]) is mainly due to the fact that the training data set (Source domain) comes from a different distribution than the deployment data (target dataset), resulting in a decrease in the performance of the model [7], largely due to the fact that the latent distribution extracted by the encoders

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

for both domain don't overlap, this can also be confirmed by using several Manifold Learning [8], [9] techniques as Bekkouch *et al.* showed [10] by reducing the dimensions of the rich latent space into a lower dimensionality and visualizing the distributions of both domains. Manifold Learning and domain generalization (deep learning in general) are both similar on many levels since they both reduce the input shape and learn an underlying structure in high dimensional data. The main difference between them is the ability for deep learning based feature extraction to include the class information in the latent space that is easily interpretable by a deep learning classifier unlike manifold learning methods which are mostly unsupervised or lack the easy integration with other deep learning components.

Such changes in real life can occur from very simple things like a change in image resolution or the brightness of the pictures or even changes in the background. As Fig 1 shows, the horse was misclassified as an Arabian Camel by ResNet mostly because of the sand and Arabian architecture in the background, which the Local Interpretable Model-agnostic Explanations (LIME) [11] algorithm (used to interpret the decisions of black-box models per sample [12]) confirms by showing the pixels on which the ResNet relied on to make the decision. The same can be found in Fig 2 where a horse painting was misclassified as a macaw parrot because of the resemblance between their colors. Such problems are unavoidable in real datasets, which created a new field in transfer learning named Domain Generalization (DG).

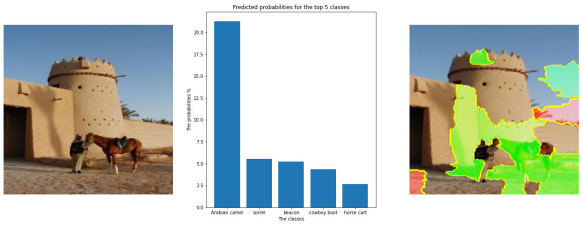


FIGURE 1. A horse wrongly predicted as an Arabian camel by ResNet, because of the surroundings. The left part is the LIME interpretation of the ResNet decision.

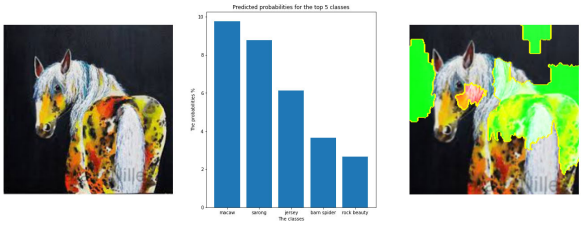


FIGURE 2. A horse wrongly predicted as a macaw parrot by ResNet, because of different colors (painting). The left part is the LIME interpretation of the ResNet decision.

DG can be also seen as a generalized case of the over-fitting problem, in the sense that the model is learning the data and not the task, even though in DG cases the model performs very well on the source test data, unlike traditional over-fitting scenarios. Domain Generalization (DG) [13] is a sub-field of Transfer Learning (TL) [14] that aims to solve the aforementioned problem by combining multiple data sources to train a more resilient model in hopes of generalizing to unseen domains. DG assumes the existence of multiple sources of data D_i^s (e.g. Photo, Art Paintings, and Cartoon) that are used for the same task T_i^s (e.g. classifying images of animals), and a target domain D^t (e.g. Sketches of the same classes of animals) that is harder to work with (harder to label or to collect). Most DG methods provide an extension to a closely related field, Domain Adaptation (DA) [10], [15] which often uses one source domain and one target domain to solve the domain shift problem. At the time of training, DA assumes the availability of target domain data but can be classified according to the presence of labels in the target domain in three key ways: Supervised [16], Unsupervised [10], and Semi-Supervised DA [17]. DG differs from DA in the fact that we do not have access to the target data nor its labels at training phase. Therefore, DG aims at building a model that can generalize well to unseen domains rather than generalizing to a single known domain.

Researchers have approached the problem of domain gaps and their consequences in many ways. One traditional yet very commonly used technique is to treat this problem as an over-fitting problem and use regularisation techniques to help the model (parametric models) generalize well [18], [19]. Many techniques have proven to be useful in the case of deep neural networks such as learning rate decay, dropout [18], batch normalisation [19], L_1, L_2 regularisation [20] and Shakeout [21]. Although these techniques were proven effective to help the model generalize well within the same data set and achieve higher test accuracy, however, it is not the most effective method for DG. Hence, we need to develop new methods that are both effective for over-fitting and for DG problems.

Recent approaches for DG are commonly neural-network-based and are separated into two main types: one-for-all and one-for-each. The former uses all source domains and learns a common model that works for all of them hoping it would generalize to future domains [22] whereas the latter approach (one-for-each), trains a different branch for each source domain. Next, at evaluation, we measure the closeness of each source domain to the target image and only consider the output of the corresponding classifier [23].

In this article, we deal with the case of one-for-all DG in its largest definition given its applicability and speed increase over the one-for-each type. We implemented a new DG method that can generalize from multiple source domains to an unknown target domain, from one domain to another, and from one domain to itself, making this method easily applicable in many real world scenarios where the CNN or the neural networks in general show signs of over-fitting and dependency on the underlying distribution of the training data.

Similar to JiGen [22], who trains a jigsaw puzzle solver over the images to help the encoder better learn the internal

structure, our approach belongs to the one-for-all category of DG approaches, focusing on how to use the training data more effectively to help the model learn better features in an unsupervised manner. In contrast to JiGen, the proposed model uses an Encoder, a Decoder, and a classifier to forget specific features of the data and not to learn it better. Unlike traditional Auto-Encoders that are trained to reconstruct the input, by training a Decoder to reconstruct the images and training the encoder in an adversarial way against the reconstruction loss, we force the Encoder to neglect the domain-specific details and only forward the information required for classification.

As proven by our experimental results on single source DG, our technique can also be helpful as a measure against overfitting. Our approach uses pure deep learning based methods that can be run easily on GPUs, making it simpler to train and quicker to converge, unlike most other DG methods that add a huge computational burden such as JiGen (to make the jigsaw puzzle).

In short, this article presents a new DG system based adversarial auto encoders by training the encoder to extract only classification needed information and remove all the style details noise, which achieves state-of-the-art efficiency in various scenarios for Domain Generalization, Domain Adaptation and Overfitting without adding a huge computational burden, making it more applicable to real-world scenarios and easily incorporated into more complex architectures. We evaluated our method against the state of the art deep learning methods based on five primary datasets and 13 sub-datasets and showed that our method outperforms most of them on all tasks.

II. RELATED WORKS

The field of transfer learning has witnessed a great deal of research interest, especially domain adaptation and domain generalization as two sub-fields of TL. Hence we will present some of the most prominent works in both fields. Furthermore, since our method is based on the use of a robust adversarial loss function, we will also briefly discuss works related to designing adversarial loss functions and reconstruction losses for neural networks in different problems.

A. DOMAIN ADAPTATION

Domain Adaptation has been one of the most active research areas in the last few years, and has been approached in both traditional Machine learning ways and more sophisticated Deep Learning based techniques. The deep Learning techniques that were applied on DA varied a lot but they all aimed at achieving two properties for the latent space of the input: (i) extract features from the data of both domains that can be used by a classifier to get good accuracy i.e Category Informative Latent Space, and (ii) make the latent spaces of both domains harder to tell apart i.e Domain Invariant Latent Space. For this purpose many researchers have used Generative models to generate images from both domains aiming at finding a mapping between domains that allows the model to reduce the domain gap [24]. Only the discriminating portion of the Generate Adversarial Network has been used to formulate a minimization-maximization competition between the feature extractor (Encoder) and the domain discriminator that showed more promising results and faster convergence [10], [25].

B. DOMAIN GENERALIZATION

Domain Generalization is less explored as topic than Domain Adaptation, but the ability to access multiple source domains allowed for more innovation and creative techniques. Most DG methods primarily fall into two main streams: (i) Calculating the similarity between and target image and possible source domains and then this information is used later to either combine or select a certain classifier to use for this sample as in BSF [26]. (ii) Combining the source domains in a way that allows the model to learn domain invariant characteristics that can generalize well to unseen domains, one of the state of the art techniques attempts to learn domain agnostic representation by rearranging the input images and asking the network to solve it as a puzzle [22]. While it has proven to be very successful, it faces a risk as different groups will share the same sub-components but are connected together differently.

C. ADVERSARIAL & RECONSTRUCTION LOSSES

Using Convolutional Auto Encoders while Pre-Training CNN classifiers is considered one of the best practices when the dataset is too small or when the labels are too sparse [27]. In order to assist with the absence of labeled data, this task leverages the availability of unsupervised data under reconstruction loss. They are also used widely used for outlier detection [28]–[30], novelty detection [31], auto-drawing for with RNNs [32] and Open-Set Recognition [33]. Reconstruction loss is also used for domain adaptation by jointly learning a shared encoding representation for: i) supervised classification ii) unsupervised reconstruction of unlabeled data [34], this way the encoder learns to extract information from the target dataset too making it more familiar with it. This idea goes exactly against ours where our goal is to maintain the latent space empty of any style information that can reduce the performances of the classifier. Adversarial losses, which are at the heart of most recent developments in Computer Vision and Generative models, are another very useful type of loss functions [35]. Adversarial losses allow us to define an unwanted situation and go the other way around it. It has been employed in GANs to generate new images similar to the real ones by detecting the differences between them and working to reduce them. In the same manner, [24] has used it to generate images in both domains, whereas [10] has defined the problem of the domains being distinguishable and trained the encoder on the opposite of it, which allowed it learn more domain agnostic representations of the images. Another interesting approach was separating source and target domains from the adversarial losses by only applying to one domain only where [36] applied it target dataset and kept the source

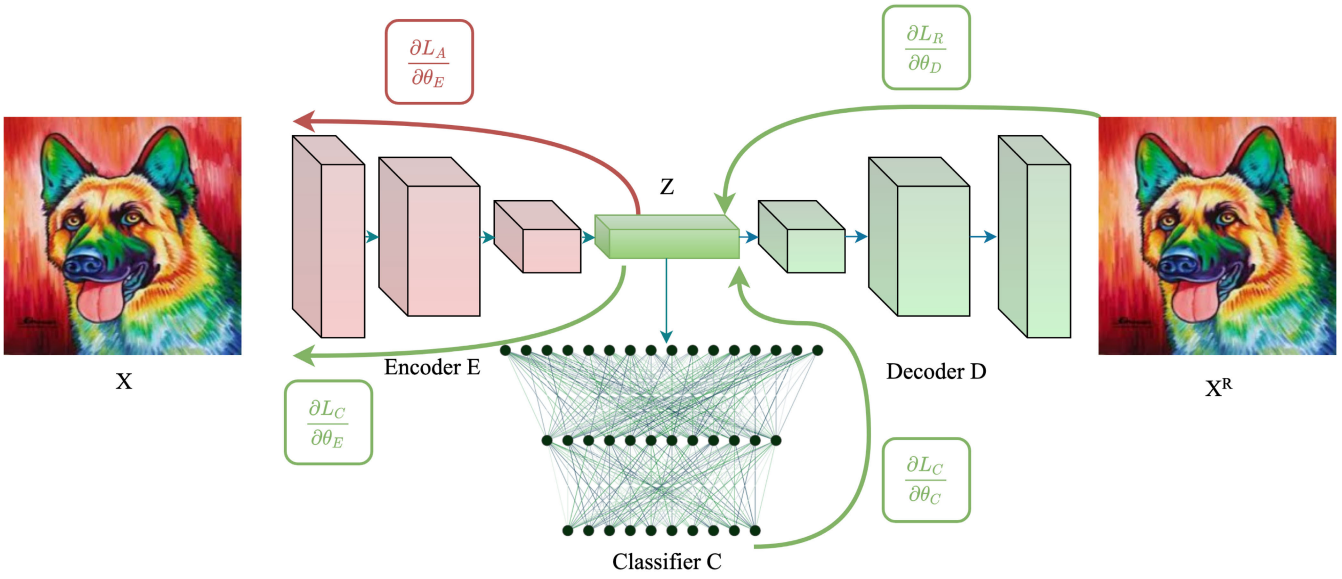


FIGURE 3. Model Architecture: The Encoder generates latent representation z which is used by the Decoder to reconstruct the input using L_R and by the Classifier to classify the sample using L_C. The encoder is trained on the classification L_C and adversarial L_A losses.

domain the same whereas [37] applies it to the source domain and keeps the target domain fixed.

III. METHODS

We explain the approach of Adversarial Reconstruction Loss for Domain Generalization and the motivation behind it in this section. We base our approach on the premise that for the same problem, deep neural networks cannot generalize to different domains because they are too dependent on their training domain. In other words, the CNN encoder portion is learning features that are helpful for prediction but also for extracting other domain-specific features that restrict the model's ability to handle unseen data. The CNN (Enoder) part of the models is responsible for the feature extraction; our main assumption is that the feature extractor extracts two types of information. Type 1 is the class-informative, which helps make the decisions and the classification, whereas type 2 is the misleading background noise. Thus, we characterize the model's ability for generalizing to unseen datasets by its ability to forget the data's peculiarities, symbolizing how much of the input has been overlooked or neglected by the encoder.

We illustrate the Encoder's ability to sustain low-level image information despite the fact that the only loss we used for the training was the classification loss. Figure 4 explains the amount of information the Encoder preserves even after applying extreme input alterations.

After training an Encoder plus a Classifier setup on MNIST, the images were reconstructed based on a frozen Encoder and newly trained Decoder. These findings on the test dataset support our hypothesis that even though we train the encoder for classification only, it retains numerous input features from its source data.

A. DOMAIN GENERALIZATION

As with all DG methods, our technique requires S source datasets (domains) and at least one target dataset (domain). N_i is used to represent the ith source dataset's sample size, such that $X_i^s = \{(x_{i,j}^s, y_{i,j}^s)\}_{j=1}^{N_i}$, where $x_{i,j}^s$ references the jth sample of the ith source dataset and $y_{i,j}^s$ is its corresponding label. Moreover, we denote M as the target domain's sample size with $X^t = \{(x_j^t, y_j^t)\}_{j=1}^{M}$, where x_j^t is the jth sample from the target dataset and y_j^t is its label, the t is used to distinguish between source and target domains.

The three main components of our model are: Encoder, Decoder, and a classifier, as shown in Figure 3. The central part of the model and our point of focus is the Encoder E(.) with its weights θ^E, which maps the input samples x into the latent embedding space z. These features are commonly known as the images' latent representation.

The Classifier C(.) with weights θ^C, is a feed forward neural network and the whole classification model is the combination of the encoder and the classifier which is represented with the function $f_c = e \circ c$, where $e : \mathcal{X} \longrightarrow \mathcal{Z}$ is the encoder function that maps the images into feature vectors and $c : \mathcal{Z} \longrightarrow \mathcal{Y}$ is the classification function operating on the latent space.

The last part of our method is the Decoder D(.), which will not be included in the final model since it is not part of the inference process. Its weights are denoted as θ^D and we use it to reconstruct the input samples given their latent space representation such that the reconstruction function $f_d = e \circ d$ where $d : \mathcal{Z} \longrightarrow \mathcal{X}$.

Each component of the architecture is trained with a different combination of losses, starting with the Classifier which is trained by minimizing the classification error

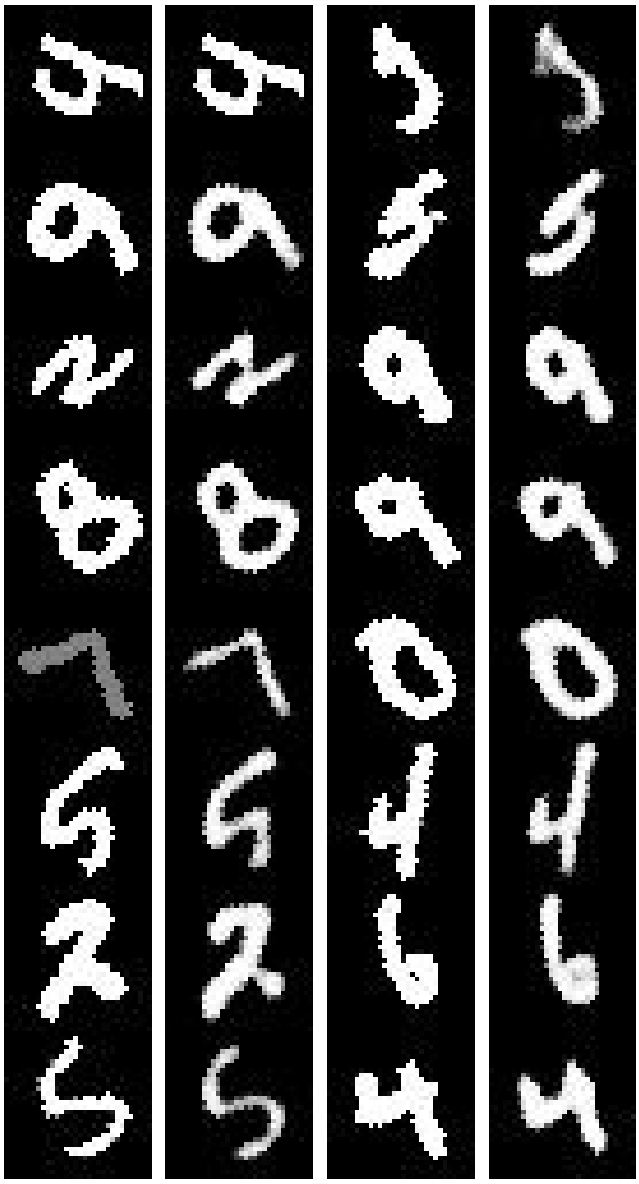


FIGURE 4. Reconstructed images formed by training a decoder on a model (Encoder+Classifier) trained only for classification. Reconstructed on the left, Input image on the right.

(cross entropy loss) $H(.,.)$

$$\mathcal{L}_c(\theta^E, \theta^C) = \sum_{i=1}^{S} \left(\sum_{x_i^s \in X_i^s} H(C[E(x_i^s)], y_i^s) \right) \tag{1}$$

The decoder's weights are updated to reduce the reconstruction Loss (Mean Squared Error) between input sample x and the reconstructed image $\hat{x}$ even though it doesn't have access to the input, it does that by mapping the latent space into a data sample.

$$\mathcal{L}_R(\theta^D) = \sum_{i=1}^{S} \left(\sum_{x_i^s \in X_i^s} \left\| D[E(x_i^s)] - x_i^s \right\|^2 \right) \tag{2}$$

Algorithm 1 Domain Generalization With Adversarial Reconstruction Loss

Input: X^s — Source domain images.
Y^s — Source domain image labels.
generalizing_epochs — NB epochs 1
pretraining_epochs — NB epochs 2
α — The learning rate
β — Balancing factor - hyperparameter
Output: θ^E — Weights of the encoder
θ^C — Weights of the classifier

```
// Start Pre-training the Model
```
for $i \leftarrow 1$ **to** *generalizing_epochs* **do**
  for $j \leftarrow 1$ **to** *nb_batches* **do**
    Sample a batch of source images
    $(x^j_{1s}, y^j_{1s}), (x^j_{2s}, y^j_{2s}), \ldots, (x^j_{Ns}, y^j_{Ns})$;
    $\theta^E = \theta^E - \alpha \frac{\partial L_C}{\partial \theta^E}$ Equation 1 ;
  end
end

```
// Start the Generalization process
```
for $i \leftarrow 1$ **to** *pretraining_epochs* **do**
  for $j \leftarrow 1$ **to** *nb_batches* **do**
    Sample a batch of source images
    $(x^j_{1s}, y^j_{1s}), (x^j_{2s}, y^j_{2s}), \ldots, (x^j_{Ns}, y^j_{Ns})$;
    $\theta^D = \theta^D - \alpha \frac{\partial L_R}{\partial \theta^D}$ Equation 2 ;
    $\theta^C = \theta^C - \alpha \frac{\partial L_C}{\partial \theta^C}$ Equation 1 ;
    $\theta^E = \theta^E - \alpha \frac{\partial (L_A + \beta L_C)}{\partial \theta^D}$ Equation 1, 3 ;
  end
end
return θ^E, θ^C

Our method's crucial element is that the reconstruction loss $\mathcal{L}_R$ will not be used to update the encoder's weights directly. Nevertheless, the encoder will be trained on both the classification loss and the adversarial of the reconstruction Loss:

$$\mathcal{L}_A(\theta^E) = -\sum_{i=1}^{S} \left(\sum_{x_i^s \in X_i^s} \left\| D[E(x_i^s)] - x_i^s \right\|^2 \right) \tag{3}$$

In computer vision, the initialization of the model's weights using an auto-encoder architecture and learning features useful for reconstructing the input is considered a standard best practice; and assumed to help build better classifiers using fewer data [38]–[41]. We propose to take in the opposite route, enabling the Encoder to update its weights under the classification loss and skipping the structure, shape, and other information that overfits the network.

The step by step process of the training is described in Algorithm 1.

1) EXTENSION TO UNSUPERVISED DOMAIN ADAPTATION

Our method is easily generalisable to the Unsupervised Domain Adaptation setting. Given the unsupervised nature of the Adversarial Reconstruction Loss, we can always add

more samples without labeling which will help the model generalize even better. We also add in this setting a separation loss that operates on the output of the encoder similar to Linear Discriminant Analysis (LDA). The optimization goal is to maximize the between-class variability (making different classes further apart from each other in the latent space) and minimize the within-class variability (making samples from the same class close together). Our separability loss is defined as follows:

$$\mathcal{L}_{sep}(\theta^E) = \left(\frac{\sum_{i\in Y}\sum_{z_{ij}\in Z_i} d(z_{ij}, \mu_i)}{\sum_{i\in Y} d(\mu_i, \mu)}\right) \times \lambda_{BF}$$
$$\lambda_{BF} = \frac{\min_i |Y_i^t|}{\max_i |Y_i^t|} \tag{4}$$

where Z_i is the set of all the latent representations of both source and target domains, that belongs to class i. For the target domain classes, we used the pseudo-labels that are produced with a high level of confidence from the classifier since we assume that the target data has no labels for training. μ_i is the mean of all latent representations with label i, such that $\mu_i = mean(Z_i)$, whereas μ is the mean of all the latent representations for both source and target $\mu = mean(Z)$. $d(.,.)$ is the distance function used to measure the dissimilarity between the latent vectors. λ_{BF} is a normalizer since the behavior of this loss is very fluctuating in cases where the batch doesn't contain a large enough amount for each class, and it represents the ratio between the number of least represented pseudo-labeled target samples $\min_i |Y_i^t|$ and the number of the most represented ones $\max_i |Y_i^t|$.

2) EXTENSION TO OVER-FITTING

Over-fitting arrives when a model has learned the training data too well. It is very common with strong models such as neural networks and decision trees. A number of techniques for combating over-fitting in neural networks exist such as reducing the model size, reducing the input data's dimensions, regularization (L1, L2), dropouts, and batch normalization, yet most of them constrain the model from actually learning category informative features.

Our technique although made for DG, can be easily applied in the case of single source datasets and contrarily to other over-fitting techniques, ours allows the model to learn as deep as possible without letting it over-fit on the style of the training data. Our method is not exclusive with other techniques, but it should be used along the side of most of the previously mentioned techniques since they are considered to be the best practice for the training process.

IV. ANALYSIS

Our Adversarial Reconstruction Loss method provided outstanding performances compared to other states of the art methods on several experiments using different datasets. This section is split into four main parts; the first one is the dataset, where we present the five primary datasets and their 13 sub-datasets. The second part is the main results section, where we compare our model against several Domain Generalization methods on four benchmarks. The third and last parts are related to unsupervised domain adaptation and over-fitting results.

A. DATASETS

To explore our Method's effect on the domain generalization problem and its related issues (UDA, overfitting), we analyze five datasets extensively chosen in the field. The first one is **MNISTR**; the Rotated MNIST dataset is an alteration to the popular digits classification dataset MNIST. The different domains of RMNIST are created via rotating images by 15 degree increments: 0, 15, 30, 45, 60, and 75 (referred to as $M_0, \ldots, M_{75}$). We employ a leave-one-out situation at the training phase, signifying that we will have five source domains and one remaining for the target. Nevertheless, the data has an identical test/train split as the primary MNIST; therefore, there is no overlap between train and test samples of the different domains. Next, we use the **MNIST-SVHN-USPS** Street View House Numbers (SVHN) which is a real-world image dataset for digit recognition commonly used with MNIST for domain adaptation tasks. SVHN is obtained from house numbers in Google Street View images and is a little bit more challenging because of many side artifacts in it and the inclusion of color. US Post Office Zip Code Data (USPS) Handwritten Digits has 7291 train and 2007 test images. The images are 16*16 grayscale pixels which make them similar to MNIST but less complex. This combination of datasets is used both for Domain Generalization and Unsupervised Domain Adaptation. **PACS** dataset is a new benchmark challenge dataset for object classification which covers seven object classes (person, elephant, dog, house, giraffe, horse, and guitar) spread across four different domains (Photo, Art Paintings, Cartoon, and Sketches), producing a more tough predicament for our models. Hence, we start with a pre-trained imagenet model, namely AlexNet. **VLCS** dataset is commonly used in Domain Generalization settings as a benchmark for performance evaluation on multi-class object recognition tasks. VLCS is an abbreviation of the four datasets that make it up: PASCAL **V**isual Object Classes 2007 (V) [42], **L**abelMe (L) [43], Caltech (C) [44], and SUN09 (S) [45]. It was created by combining the five common classes between its sub-datasets, which are: Birds, Cars, Dogs, Chairs, Person. For evaluation purposes, we use the same setup as the previous works [13], [22], [46], [47] by using pre-extracted DeCAF6 features (4096-dimensional vector) and performing a leave-one-domain-out validation by randomly splitting each domain into 70% training and 30% testing. We also use a two fully connected layer neural network inputting to two fully connected layers with sizes of 1024 and 128 respectively with ReLU activation. **IXMAS** is a cross view action recognition dataset containing eleven different human actions that are recorded by five cameras in different positions. We aim to build an action detector that works regardless of the angle of view. We follow the same experimental setup as [23], [47], [48] by using

the same Dense trajectory input features and excluding the irregularly performed actions by only keeping the first five actions (check watch, cross arms, scratch head, sit down, get up) performed by the six actors (Alba, Andreas, Daniel, Hedlena, Julien, Nicolas). Each camera position is treated as a separate data domain named (0,1,2,3,4). Following the previous works, we generate a 4-source domain generalization task (leave-one-domain-out). **Skin lesion dataset** is a combination of 7 public datasets for skin lesion detection collected from different equipments. The main dataset is HAM10000 [49] which is used as part of the source data of all experiments following the setup of [50], [51]. The other datasets are Dermofit (DMF) [52], Derm7pt (D7P) [53], MSK [54], PH2 [55], SONIC (SON) [54], and UDA [54]. All the datasets contain 7 common lesions which are melanoma (mel), melanocytic nevus (nv), dermatofibroma (df), basal cell carcinoma (bcc), vascular lesion (vasc), benign keratosis (bkl), and actinic keratosis (akiec). Following [51] we split the data into training (50%), validation (20%) and testing set (30%) in a stratified manner. In each experiment we choose one of the secondary datasets as a target domain and keep HAM10000 and the other dataset for the source domains. We use a pretrained Resnet18 as the backbone of our model for fair comparison with the other methods. **Hip MR scan Landmark detection HML** dataset is a 3D dataset of 423 3D Magnetic resonance scans of the hip area for 114 patients [56]. The dataset contains 12 landmarks annotated by doctors for diagnosis of several pathologies such as Hip dysplasia and impingement syndrome. The dataset contains three domains which are the different MRI sequences (T1 weighted, T2 weighted and PD weighted). All three modalities are needed for correct indentification of early signs of hip abnormalities. However there is no guarantee that all three of them will be available at a specific hospital. The challenge is therefore to mitigate the problem of missing sequences and ensure higher rates of abnormality defections. For each experiment we use two source domains and the remaining one is the target. We split the data into 80% training 5% validation (For hyper-parameter tuning) and 15% testing in a stratified manner according to the pathologies for the patients and the patient IDs don't overlap between sets. We use a pretrained Resnet18 as the backbone for our model and decoder of 3 layers.

B. DOMAIN GENERALIZATION RESULTS

1) DIGIT CLASSIFICATION: RMNIST

For the task of digit classification, we assessed our model's performance versus numerous state of the art deep learning methods in domain generalization which are: MTAE [57], CAE [58], BSF [26], UDS [46], PSSO [59], AFLAC [60]. We were inspired to pursue this method after conducting experiments on the MNIST dataset to understand domain dependency better. Therefore, our model performs significantly better on this dataset than all the current state of the art, as Table 1 clearly shows our model's performance exceeds all the other models on average and is ranked at least first or second in each experiment.

TABLE 1. Domain Generalization for digit classification: RMNIST. The average accuracy over 20 runs of the model. We represent each experiment by the name of its target dataset.

Method	0	15	30	45	60	75	mean
CAE [58]	72.1	95.3	92.6	81.5	92.7	79.3	85.5
MTAE [57]	82.5	96.3	93.4	78.6	94.2	80.5	87.5
PSSO [59]	**94.2**	82.5	96.3	93.4	78.6	80.5	87.5
UDS [46]	84.6	95.6	94.6	82.9	94.8	82.1	89.1
BSF [26]	85.6	95.0	95.6	95.5	95.9	84.3	92.0
AFLA [60]	89.3	**98.8**	**98.3**	93.3	**97.4**	88.1	94.2
ARL (ours)	89.5	97.2	97.3	**98.1**	96.7	**89.4**	**94.7**

The reported results are the averaged over 20 runs of the model with the learning rate set to 0.003, *generalizaing_epochs* $=$ 50, *pretraining_epochs* $=$ 100, and the balancing factor set to $\beta = 0.1$. Our method outperformed all other methods on average providing more consistent results than others especially on the extreme case of 75 degrees, where we had 1.33% accuracy increase over the second best method AFLAC. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 5 hours and 46 min. The time needed to train the models for classification only without our loss is 2 hours and 18 mins.

In order to fully understand what our technique achieves we regenerated the experiment from Fig 4 but with adversarial reconstruction loss used for the training of the model. So our experiment goes as follows, We train the Encoder by the adversarial reconstruction loss and the classification loss as described in Algorithm 1 and after convergence, we re-train a new decoder on the latent space of the MNIST dataset without changing the encoder weights. After it converges, we evaluate the results on the test data with extreme rotations to see if the same effects from the previous experiment Fig4 still holds. We inferred that the results in Fig 5 are definitely different in this case where most of the reconstructions appear to be centered and without rotation, unlike their respective original inputs. Furthermore, we can see that most of the specific details in the pictures tend not to appear in the reconstructed images. We can also easily see that all the reconstructions have the same class as their input. Proving that the aim of our method was actually achieved and that the learned features don't contain information about the specific details of the input yet they are still useful for classification.

2) OBJECT RECOGNITION 1: VLCS

We use the same experimental setup as the deep learning works we compare our model with [13], [22] with our learning rate being $\alpha = 0.003$, *generalizaing_epochs* $=$ 550, *pretraining_epochs* $=$ 300, and the balancing factor is set to $\beta = 0.15$. We found the values of our hyper parameter with 10-fold cross validation hyper parameter tuning and we report the average test accuracy over 20 experiments.

TABLE 2. VLCS results for Domain Generalization.

Source	Target	Deep All (Base line)	MTAE [57]	MMD-AAE [60]	CCSA [46]	MLDG [61]	Epi-FCR [47]	MASF [16]	JiGen [22]	Ours
L,C,S	V	68.67	63.90	67.70	67.10	67.7	67.1	69.14	**70.62**	68.93
V,CS	L	63.10	60.13	62.60	62.10	61.3	64.3	64.90	60.90	**65.12**
V,L,S	C	92.86	89.05	94.40	92.30	94.4	94.1	94.78	96.93	**96.97**
V,L,C	S	64.11	61.33	64.40	59.10	65.9	65.9	**67.64**	64.30	65.78
Average		72.19	68.60	72.28	70.15	72.3	72.9	74.11	73.19	**74.2**

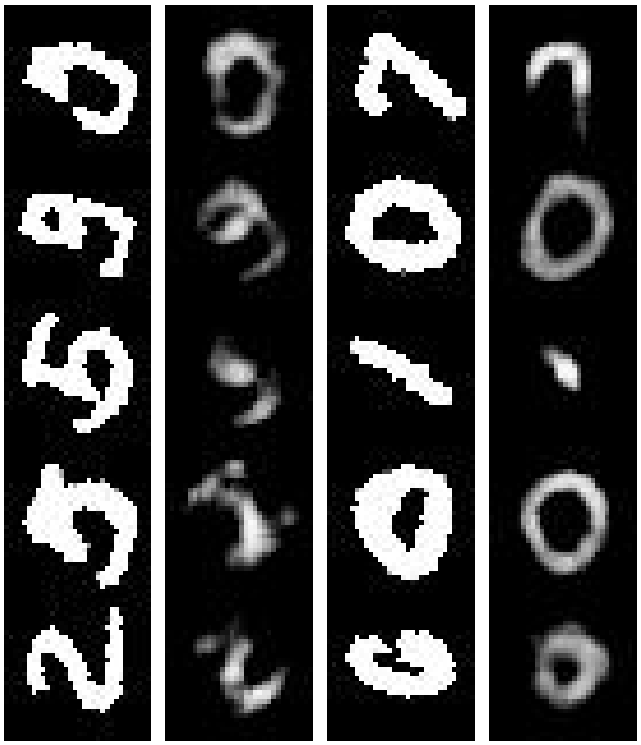


FIGURE 5. Reconstructed images formed after applying our ARL Generalization and training a new decoder to reconstruct the input images. Reconstructed on the right, Input image on the left.

TABLE 3. AlexNet PACS dataset results for Domain Generalization.

Method	Photo	Art	Cartoon	Sketches	Mean
IRDCD [62]	82.9	61.2	63.8	57.51	66.7
deeper [63]	89.50	62.86	66.97	57.51	69.21
MetaGen [61]	88.00	66.23	66.88	58.96	70.01
SSO [59]	87.9	66.8	69.7	56.3	70.2
BSF [64]	90.2	64.1	66.8	60.1	70.3
MetaReg [65]	91.07	**69.82**	70.35	59.26	72.62
JiGen [22]	89.00	67.63	**71.71**	**65.18**	**73.38**
ARL (ours)	**92.1**	66.42	68.87	63.07	72.62

TABLE 4. Cross-view action recognition results (accuracy. %) on IXMAS dataset for Domain Generalization. Best result in bold.

Source	0,1,2,3	0,1,2,4	0,1,3,4	0,2,3,4	1,2,3,4	Ave.
Target	4	3	2	1	0	
LRE-SVM	75.8	86.9	84.5	83.4	92.3	84.6
CCSA [46]	75.8	92.3	94.5	91.2	**96.7**	90.1
MMD	79.1	94.5	95.6	93.4	**96.7**	91.9
DANN	75.0	94.1	97.3	95.4	95.7	91.5
MLDG	70.7	93.6	97.5	95.4	93.6	90.2
CrossGrad	71.6	93.8	95.7	94.2	94.2	89.9
MetaReg	74.2	94.0	96.9	97.0	94.7	91.4
AGG	73.1	94.2	95.7	95.7	94.4	90.6
Epi-FCR	76.9	**94.8**	**99.0**	**98.0**	96.3	93.0
Ours	**79.4**	94.2	98.4	97.1	96.5	**93.1**

As Table 2 shows, our model achieves the best performance on two experiments out of 4 and is quite competitive in the rest being second and third, whereas on average it achieves the best results. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 8 hours and 12 min. The time needed to train the models for classification only without our loss is 4 hours and 12 mins.

3) OBJECT RECOGNITION 2: PACS

We followed the same protocol as the previous deep learning papers, by using the same train/test/validation splits for a fair comparison and the same model sizes and pre-trained weights. Our learning rate is $\alpha = 0.003$, $generalizaing\_epochs = 150$, $pretraining\_epochs = 500$, and the balancing factor is set to $\beta = 0.25$.

PACS object recognition dataset provides a much more challenging setting due to its big image resolution, small sample size, and the notable variation among the domains. Nevertheless, our method outperformed most of the state of the art as table 3 shows. Furthermore, it gave near-perfect results on the Photo target domain being the best at this experiment. Overall, our model performed very well and was ranked 2nd after JigSaw with similar performances as the MetaReg model. Even though our model did not rank first, it is still more applicable in real-world scenarios, given its training speed and simplicity.

We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for 2 hours and 37 min per experiment. The time needed to train the models for classification only without our loss is 1 hours and 28 mins.

4) ACTION RECOGNITION:IXMAS

IXMAS is a human action dataset with 5 actions and 5 different domains. We train on 4 domains and test on the last one. We report the average accuracy of over 20 runs. We use one hidden layer network with 2000 hidden neurons

TABLE 5. Skin Lesion results for Domain Generalization. The bolded experiment is the best and underlined in the second.

Target	DeepAll	MASF [16]	MLDG [61]	CCSA [46]	LDDG [50]	ARL (ours)
DMF	0.2492±0.0127	0.2692±0.0146	0.2673±0.0452	0.2763±0.0263	**0.2793±0.0244**	0.2789±0.0137
D7P	0.5680±0.0181	0.5678±0.0361	0.5662±0.0212	0.5735±0.0227	0.6007±0.0208	**0.6461±0.0319**
MSK	0.6674±0.0083	0.6815±0.0122	0.6891±0.0167	0.6826±0.0131	**0.6967±0.0193**	0.6830±0.0172
SON	0.8613±0.0296	0.9204±0.0227	0.8817±0.0198	0.9045±0.0128	**0.9272±0.0117**	0.9184±0.0218
PH2	0.8000±0.0167	0.7833±0.0101	0.8016±0.0096	0.7500±0.0419	0.8167±0.0096	**0.8453±0.0239**
UDA	0.6264±0.0312	0.6538±0.0196	0.6319±0.0284	0.6758±0.0138	0.6978±0.0110	**0.7418±0.0402**
Avg	0.6287	0.6460	0.6396	0.6438	0.6697	**0.6855**

as the previous works did. Our learning rate is $\alpha = 0.01$, *generalizaing_epochs* $= 50$, *pretraining_epochs* $= 150$, and the balancing factor is set to $\beta = 0.1$. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 6 hours and 08 min. The time needed to train the models for classification only without our loss is 3 hours and 02 mins.

From our experimental results in Table 4 we see that our model is very competitive with the state of the art having the best average accuracy and if most experiments either being the best or the 2nd best.

5) SKIN LESION

Skin Lesion dataset is an image classification dataset used to benchmark the knowledge transfer abilities of several models. It contains 7 classes and 7 domains (1 primary and 6 secondary). For each of the experiments of Table 5 we use one of the 6 secondary dataset as the target data and the rest as the source. We report the results of the average of 5 runs on each experiment and take the results as mentioned in their original papers. We use the same experimental setup as the state of the art methods used for comparison, by training a Resnet 18 as our base classifier and using mirror of their encoder as our decoder component. Our learning rate is $\alpha = 0.003$, *generalizaing_epochs* $= 75$, *pretraining_epochs* $= 150$, and the balancing factor is set to $\beta = 0.2$. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 7 hours. The time needed to train the models for classification only without our loss is 4 hours.

As the results on Table 5 show, all the DG techniques can outperform the DeepAll methods (which trains on all the source domains using only the classification loss) which is the expected behaviour. The best method on average is ours with a significant marge of 1.58%. Our paper provides the best results on 3 out of 6 experiments followed by LDDG and MASF. We can see that the results of the different methods are overall consistent with the difficulty of the domain gaps, where they provide good results on datasets such as PH2 and SON, and fail on datasets such as DMF.

6) PELVIC LANDMARK DETECTION

The pelvic Landmark Detection [56] dataset is a 3D MR scans dataset manually annotated by expert doctors.

TABLE 6. Pelvic Landmark detection results for Domain Generalization. The bolded experiment is the best.

Source	T1, T2	T1, PD	T2, PD	Avg
Target	PD	T2	T1	
Baselines				
Deep All	0.8153	0.9360	0.8813	0.8875
Theoretical Max	0.9174	0.9821	0.9468	0.9487
Models				
JiGen [22]	0.8690	0.9332	0.9214	0.9078
Epi-FCR [47]	0.8814	**0.9420**	0.9253	0.9162
ARL (ours)	**0.8973**	0.9261	**0.9338**	**0.9190**

It contains 12 landmarks with 423 3D scans of size ranging from 350*350*42 to 370*370*128 for the x, y, and z axes respectively. The dataset contains 3 different domains which represent the different MR squences used for each scan: T1, T2, and PD. We report the results of the average of 20 runs on each experiment. We use a Resnet 18 as our base classifier and using mirror of their encoder as our decoder component. Our learning rate is $\alpha = 0.01$, *generalizaing_epochs* $= 50$, *pretraining_epochs* $= 150$, and the balancing factor is set to $\beta = 0.03$. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 12 hours. The time needed to train the models for classification only without our loss is 8 hours.

We compared our model against two of the state of the art methods in Domain Generalization which are JiGen and Epi-FCR. Our model and all compared DG models outperform the deep all baseline as shown in Table 6 and Fig 6. Our method outperforms both of them but with a small margin against Epi-FCR which outperforms our method on the $T1, PD \longrightarrow T2$ experiment. Our method still outperforms both methods on the two other experiments.

C. UNSUPERVISED DOMAIN ADAPTATION

In the case where the unlabeled target images exist during the training (Unsupervised Domain Adaptation), we add an extra loss to our model which is the Separability loss 4. We explore the effects of this loss along with the performance of our model on two challenging scenarios, MNIST-USPS-SVHN dataset and the PACS data.

1) DIGIT CLASSIFICATION: MNIST-USPS-SVHN

This is the most common benchmark for domain adaptation tasks and UDA specifically. Hence we follow the same

TABLE 7. Digit Recognition Benchmark on the MNSIT-USPS-SVHN dataset for Unsupervised Domain Adaptation. Each experiment name follows source_domain - target_domain naming convention. ARL-sep is used to reference to our method + the seperability loss and ARL is used to reference our model without it. The "-" notation is used for experiments where the results have not been reported in previous works.

Method	UB	LB	JAN [70]	Gen2Adpt [69]	MCD [68]	I2I [67]	TarGAN [66]	DupGAN [24]	TPN [71]	TripNet [10]	ARL-sep	ARL
SVHN - MNIST	98.97	62.19	78.4	92.4	93.6	90.1	98.1	92.46	93.0	94.70	**98.7**	93.81
MNIST - USPS	95.02	86.75	84.4	92.8	90.0	**98.8**	93.8	96.01	92.1	97.63	98.3	97.12
USPS - MNIST	98.96	75.52	83.4	90.8	88.5	97.6	94.1	**98.75**	94.1	97.94	97.14	95.31
$SVHN_E$ - MNIST	98.97	73.67	-	-	-	-	-	96.42	-	98.57	**98.76**	97.13

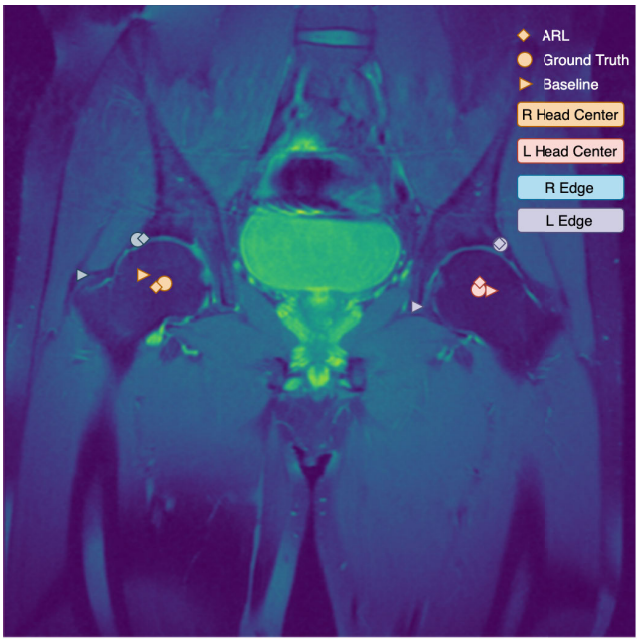


FIGURE 6. Comparison of the deep all baseline versus our ARL models on the task of landmark detection. Only four out of twelve landmarks are shown. The 3D landmarks were projected into a coronal MR cross-section for better visibility.

experimental setup as [10], [24]. We compare our results against first the two baselines (Upper Bound UB, and Lower Bound LB) which represent the accuracy of training and testing on the target dataset, and the accuracy of training on the source dataset only without access to the target dataset (not even unlabeled images), respectively. We also compare it against several of the state of the art deep learning methods in the field such as TripNet [10], DuplexGan [24], TarGan [66], Image2Image [67], Maximum Classifier Discrepancy [68], Generate to adapt [69], Joint Adaptation Networks [70] and Transferrable Prototypical Networks [71].

Our learning rate is $\alpha = 0.01$, $generalizaing\_epochs = 250$, $pretraining\_epochs = 200$, and the balancing factor is set to $\beta = 0.15$. Table 7 shows that our method outperforms most of the current state of the art techniques in 2 out of 4 experiments and ranked 2nd in the other two being only a few 0.05% away in the MNIST-USPS experiment. We can also see that our ARL-sep model outperforms our ARL model on all experiments, demonstrating the efficiency of the separability loss, yet it is also worth mentioning that the ARL model alone performed nicely being only 1.18% behind ARL-sep in the MNIST - USPS. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 1 hours and 32 min. The time needed to train the models for classification only without our loss is 0 hours and 31 mins.

2) PACS - MULTI-SOURCE DOMAIN ADAPTATION

Multi-source Domain Adaptation is a subset of DA where we have multiple source domains with labels but they are treated as one source, and a target domain either with or without labels. We are focused on the unsupervised case where the target domain is only available with images. Our method is unsupervised at its core making it easily applied in such case. To verify our assumptions we make the same experimental setup as other deep learning methods such as JiGen [22], DDiscovery [72], and Dial [73] by using ResNet18 [74] as our base model (Encoder + Classifier), whereas our Decoder is built as the mirror of the Encoder. We compare our method against all of the previous models and against a ResNet18 only model as our lower baseline. Our learning rate is $\alpha = 0.003$, $generalizaing\_epochs = 350$, $pretraining\_epochs = 500$, and the balancing factor is set to $\beta = 0.1$. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 12 hours. The time needed to train the models for classification only without our loss is 8 hours and 12 mins.

The results in Table 8 summarize the outcome of this experiment, where the provided accuracies show that our method ARL-sep is superior to the other techniques on average and on two out of four of the experiments which are Photo target domain and the more difficult task of Cartoon target domain. We can also see that even though the ARL only model isn't outperforming the other methods but it still way better than the baseline with a 8.78% increase in accuracy on average and a maximum of 11.64% accuracy increase on the Sketches dataset.

TABLE 8. Multi-source Unsupervised Domain Adaptation results on PACS datasets obtained as average over five runs for each experiment.

PACS-DA	photo	art paint.	cartoon	sketches	Avg.
ResNet 18 [74]	92.9	74.7	72.4	60.1	75.0
Dial [73]	97.0	87.3	85.5	66.8	84.2
DDiscovery [72]	97.0	**87.7**	86.9	69.6	85.3
JiGen [22]	97.9	84.8	81.1	**79.1**	85.7
ARL-sep	**98.3**	86.1	**87.6**	73.4	**86.3**
ARL	96.5	82.9	83.9	71.7	83.7

TABLE 9. Accuracy results of different models on digit classification datasets MNIST-USPS-SVHN and MNISTR for the Over-fitting scenario. The best model is bolded and the second best is underlined.

Method	OF	WT	T-ARL	O-ARL	F-ARL-sep
MNIST	63.74	98.97	*99.31*	94.73	**99.54**
USPS	72.41	95.02	**98.12**	96.41	*97.93*
SVHN	58.46	94.97	*97.85*	92.9	**98.14**
Avg.	64.87	96.32	*98.42*	94.68	**98.53**
MNISTR					
0	63.74	98.97	*99.31*	94.73	**99.54**
15	60.13	96.64	**98.07**	91.93	*97.17*
30	68.52	98.03	*98.69*	92.86	**99.05**
45	68.24	98.14	*98.83*	94.33	**99.29**
60	65.05	97.12	*97.41*	92.74	**98.17**
75	62.48	*97.59*	97.43	93.42	**97.84**
Avg.	64.69	97.748	*98.29*	93.33	**98.51**

D. OVER-FITTING

Over-fitting problems have been explored ever since the start of neural networks. Given the strong ability of neural nets to remember and memorize data samples. To evaluate the efficiency of our method on this problem we make the following setting, Train a model longer than it needs to force it to over fit, and then see if adding our loss can help bring it back from the over-fitting scenario, we refer to this model as (O-ARL).

We compare our method against several baselines: (i) Over-fitted model (OF), (ii) Well trained model (WT), (iii) model trained with ARL only from the start (T-ARL), and (iv) model fine-tuned with ARL-sep (F-ARL-sep). We perform this experiment on several benchmarks for digit classification which are: MNIST, SVHN, USPS, MNISTR-0, ..., MNISTR75. For each one of these experiments we used a different set of Hyper-parameters which are all mentioned in Table 10. We use the same experimental setup as [75]. We trained our model on a Tesla V100 SXM2 32 GB with a server with 64 cores and 80G of ram, for a total of 14 hours and 52 min. The time needed to train the models for classification only without our loss is 2 hours and 12 mins.

Table 9 shows the results of our over fitting experiments. The most obvious conclusion we can make is that the F-ARL-sep model, which was first trained on the data and then fine tuned with both the Adversarial Reconstruction Loss and the Separability loss, outperforms all the other models in most cases specifically the models that suffer from over-fitting OF and those who are well trained WT proving that our method is quite good for increasing model's performances and accuracy even on the same data domain. We can also see that O-ARL model which was used on top of an over-fitted OF model was able to help the model go back to performing good even though it was not as good as F-ARL-sep but it still gave an increase of 29.81% in accuracy on average. We also see that the T-ARL model which is trained from the beginning on the ARL loss was as rigid as O-ARL and even better than WT model in most of the cases.

TABLE 10. Hyper-parameters for the over fitting experiments on digit classification Table 9. G-epochs is generalizing epochs and PT-epochs is pretraining-epochs.

Hyper-paramter	α	G-epochs	PT-epochs	β
MNIST	*0.01*	*50*	100	*0.2*
USPS	*0.01*	*50*	*100*	*0.15*
SVHN	*0.003*	*250*	500	*0.15*
MNISTR				
0	*0.01*	*50*	100	*0.2*
15	*0.007*	*100*	*200*	*0.25*
30	*0.007*	*100*	250	*0.15*
45	*0.003*	*250*	500	*0.1*
60	*0.003*	*250*	500	*0.15*
75	*0.003*	250	*500*	*0.1*

FIGURE 7. Comparison of different models on the task of digit classification on MNIST for the over-fitting scenario. The accuracy results are reported as the average of 5 experiments with the best hyper-parameters. OF is the over-fitted model, which is used by O-ARL as the initial start for solving the over-fitting problem. WT is the well trained model, T-ARL is the model which is trained from the start with ARL, and F-ARL-Sep is the WT model and fine-tuned with both ARL and sep loss4.

We also confirm our findings through Figure 7 where we show the behaviour of our different losses and how they influence the testing accuracy of the model on the MNIST dataset. We can easily notice that the over-fitted models always go up and then quickly decreases in performance as shown with OF chart, which is continued using the O-ARL chart which drops the performance in the first few epochs but then quickly starts giving positive outcome on the model's performance approaching results provided by the WT models. We can also notice that the WT models achieve better than our models in the first few epochs where as our models (F-ARL-Sep and T-ARL) improve slower but with enough epochs they exceed the WT performances.

V. CONCLUSION

We proposed a simple but effective task agnostic method for Domain Generalization and Unsupervised Domain

Adaptation that is based on the assumption that models extract two types of information, class informative -useful- and style information -harmful-. Our method pushes the model to forget the style information while keeping the class informative part of the input which leads to high performance increase on several Object detection and classification benchmarks for DG and UDA. Our method also showed a great effect in fixing over-fitted models as shown by the experimental results. Moreover, the proposed method shows great promise of wide applicability since it is implemented orthogonally to other models and hence can be applied to different problems such as facial recognition without having to change the underlying algorithms.

REFERENCES

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[2] M. Dikmen and C. Burns, "Trust in autonomous vehicles: The case of tesla autopilot and summon," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1093–1098.

[3] D. Baylor *et al.*, "TFX: A tensorflow-based production-scale machine learning platform," in *Proc. KDD*, Aug. 2017, pp. 1387–1395.

[4] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, Jan. 2004.

[5] A. Tsymbal, "The problem of concept drift: Definitions and related work," *Comput. Sci. Dept., Trinity College Dublin*, vol. 106, no. 2, p. 58, 2004.

[6] V. M. A. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. A. P. A. Batista, "Challenges in benchmarking stream learning algorithms with real-world data," 2020, *arXiv:2005.00113*. [Online]. Available: http://arxiv.org/abs/2005.00113

[7] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Apr. 2014.

[8] F. Luo, H. Huang, Y. Duan, J. Liu, and Y. Liao, "Local geometric structure feature for dimensionality reduction of hyperspectral imagery," *Remote Sens.*, vol. 9, no. 8, p. 790, Aug. 2017.

[9] G. Shi, H. Huang, and L. Wang, "Unsupervised dimensionality reduction for hyperspectral imagery via local geometric structure feature learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1425–1429, Aug. 2020.

[10] I. E. I. Bekkouch, Y. Youssry, R. Gafarov, A. Khan, and A. M. Khattak, "Triplet loss network for unsupervised domain adaptation," *Algorithms*, vol. 12, no. 5, p. 96, May 2019, doi: 10.3390/a12050096.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," in *Proc. ICML Workshop Hum. Interpretability Mach. Learn.*, 2016, pp. 1–5.

[13] Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 6450–6461.

[14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[15] E. Batanina, I. E. I. Bekkouch, Y. Youssry, A. Khan, A. M. Khattak, and M. Bortnikov, "Domain adaptation for car accident detection in videos," in *Proc. 9th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2019, pp. 1–6.

[16] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5715–5725.

[17] H. Daumé, III, A. Kumar, and A. Saha, "Frustratingly easy semi-supervised domain adaptation," in *Proc. Workshop Domain Adaptation Natural Lang. Process.*, 2010, pp. 53–59.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, pp. 1–11, Feb. 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[20] A. Y. Ng, "Feature selection, L_1 vs. L_2 regularization, and rotational invariance," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 78.

[21] G. Kang, J. Li, and D. Tao, "Shakeout: A new approach to regularized deep neural network training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1245–1258, May 2018.

[22] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2229–2238.

[23] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5400–5409.

[24] L. Hu, M. Kan, S. Shan, and X. Chen, "Duplex generative adversarial network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1498–1507.

[25] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[26] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci, "Best sources forward: Domain generalization through source-specific nets," *CoRR*, vol. abs/1806.05810, pp. 1–6, Jun. 2018. [Online]. Available: http://arxiv.org/abs/1806.05810

[27] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Computer Vision—ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 597–613.

[28] K. Yakovlev, I. E. I. Bekkouch, A. M. Khan, and A. M. Khattak, "Abstraction-based outlier detection for image data," in *Proc. Intell. Syst. Appl. Conf. (IntelliSys)*, in Advances in Intelligent Systems and Computing, London, U.K., vols. 1 and 1250, K. Arai, S. Kapoor, and R. Bhatia, Eds. Springer, Sep. 2020, pp. 540–552, doi: 10.1007/978-3-030-55180-3_40.

[29] A. R. Rivera, A. Khan, I. E. I. Bekkouch, and T. S. Sheikh, "Anomaly detection based on zero-shot outlier synthesis and hierarchical feature distillation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 16, 2020, doi: 10.1109/TNNLS.2020.3027667.

[30] B. I. Ibrahim, D. C. Nicolae, A. Khan, S. I. Ali, and A. Khattak, "VAE-GAN based zero-shot outlier detection," in *Proc. 4th Int. Symp. Comput. Sci. Intell. Control (ISCSIC)*. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 1–5, doi: 10.1145/3440084.3441180.

[31] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6822–6833.

[32] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," 2015, *arXiv:1502.04623*. [Online]. Available: http://arxiv.org/abs/1502.04623

[33] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4016–4025.

[34] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," *CoRR*, vol. abs/1607.03516, pp. 1–21, Jul. 2016. [Online]. Available: http://arxiv.org/abs/1607.03516

[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. NIPS*, 2014, pp. 2672–2680.

[36] J. Yang, H. Zou, Y. Zhou, Z. Zeng, and L. Xie, "Mind the discriminability: Asymmetric adversarial domain adaptation," in *Computer Vision—ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 589–606.

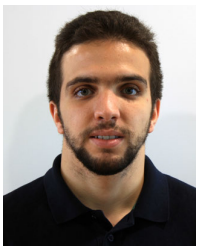
[37] J. Yang, H. Zou, Y. Zhou, and L. Xie, "Towards stable and comprehensive domain alignment: Max-margin domain-adversarial training," in *Proc. ICLR*, 2020, pp. 1–12.

[38] Z. Li, L. Qu, Q. Xu, and M. Johnson, "Unsupervised pre-training with seq2seq reconstruction loss for deep relation extraction models," in *Proc. Australas. Lang. Technol. Assoc. Workshop*, 2016, pp. 54–64.

[39] L. Le, A. Patterson, and M. White, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 107–117.

[40] M. F. Ferreira, R. Camacho, and L. F. Teixeira, "Autoencoders as weight initialization of deep classification networks for cancer versus cancer studies," 2020, *arXiv:2001.05253*. [Online]. Available: http://arxiv.org/abs/2001.05253

[41] R. Xie, J. Wen, A. Quitadamo, J. Cheng, and X. Shi, "A deep auto-encoder model for gene expression prediction," *BMC Genomics*, vol. 18, no. 9, p. 845, Nov. 2017.
[42] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[43] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, May 2008.
[44] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, p. 178.
[45] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 129–136.
[46] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," *CoRR*, vol. abs/1709.10190, pp. 1–11, Sep. 2017. [Online]. Available: http://arxiv.org/abs/1709.10190
[47] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," *CoRR*, vol. abs/1902.00113, pp. 1–11, Jan. 2019. [Online]. Available: http://arxiv.org/abs/1902.00113
[48] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 628–643.
[49] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, no. 1, Dec. 2018, Art. no. 180161.
[50] H. Li, Y. Wang, R. Wan, S. Wang, T.-Q. Li, and A. C. Kot, "Domain generalization for medical imaging classification with linear-dependency regularization," 2020, *arXiv:2009.12829*. [Online]. Available: http://arxiv.org/abs/2009.12829
[51] C. Yoon, G. Hamarneh, and R. Garbi, "Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification," in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds. Cham, Switzerland: Springer, 2019, pp. 365–373.
[52] L. Ballerini, R. B. Fisher, B. Aldridge, and J. Rees, "A color and texture based hierarchical K-NN approach to the classification of non-melanoma skin lesions," in *Color Medical Image Analysis*. Dordrecht, The Netherlands: Springer, 2013, pp. 63–86, doi: 10.1007/978-94-007-5389-1_4.
[53] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, "Seven-point checklist and skin lesion classification using multitask multimodal neural nets," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 2, pp. 538–546, Mar. 2019.
[54] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. K. Mishra, H. Kittler, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)," *CoRR*, vol. abs/1710.05006, pp. 1–5, Oct. 2017. [Online]. Available: http://arxiv.org/abs/1710.05006
[55] T. Mendonca, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira, "PH2—A dermoscopic image database for research and benchmarking," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 5437–5440.
[56] I. E. I. Bekkouch, T. Aidinovich, T. Vrtovec, R. Kuleev, and B. Ibragimov, "Multi-agent shape models for hip landmark detection in MR scans," *Proc. SPIE*, vol. 11596, pp. 153–162, Feb. 2021, doi: 10.1117/12.2580862.
[57] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," *CoRR*, vol. abs/1508.07680, pp. 1–9, Aug. 2015. [Online]. Available: http://arxiv.org/abs/1508.07680
[58] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. ICML*, 2011, pp. 833–840.
[59] H. Wang, Z. He, Z. C. Lipton, and E. P. Xing, "Learning robust representations by projecting superficial statistics out," *CoRR*, vol. abs/1903.06256, pp. 1–16, Mar. 2019. [Online]. Available: http://arxiv.org/abs/1903.06256
[60] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Adversarial invariant feature learning with accuracy constraint for domain generalization," *CoRR*, vol. abs/1904.12543, pp. 1–17, Apr. 2019. [Online]. Available: http://arxiv.org/abs/1904.12543
[61] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.
[62] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Domain generalization via invariant representation under domain-class dependency," in *Proc. ICLR*, 2018, pp. 1–13.
[63] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5542–5550.
[64] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci, "Best sources forward: Domain generalization through source-specific nets," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1353–1357.
[65] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "MetaReg: Towards domain generalization using meta-regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 998–1008.
[66] F. Lv, J. Zhu, G. Yang, and L. Duan, "TarGAN: Generating target data with class labels for unsupervised domain adaptation," *Knowl.-Based Syst.*, vol. 172, pp. 123–129, May 2019.
[67] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.
[68] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.
[69] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8503–8512.
[70] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," *CoRR*, vol. abs/1605.06636, pp. 1–10, May 2016. [Online]. Available: http://arxiv.org/abs/1605.06636
[71] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," *CoRR*, vol. abs/1904.11227, pp. 1–9, Apr. 2019. [Online]. Available: http://arxiv.org/abs/1904.11227
[72] M. Mancini, L. Porzi, S. R. Bulò, B. Caputo, and E. Ricci, "Boosting domain adaptation by discovering latent domains," *CoRR*, vol. abs/1805.01386, pp. 1–10, May 2018. [Online]. Available: http://arxiv.org/abs/1805.01386
[73] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Just DIAL: Domain alignment layers for unsupervised domain adaptation," *CoRR*, vol. abs/1702.06332, pp. 1–11, Feb. 2017. [Online]. Available: http://arxiv.org/abs/1702.06332
[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, pp. 1–12, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03385
[75] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," *CoRR*, vol. abs/1204.3968, pp. 1–4, Apr. 2012. [Online]. Available: http://arxiv.org/abs/1204.3968



IMAD EDDINE IBRAHIM BEKKOUCH received the B.S. degree in computer science from Abdelhamid Mehri Constantine 2 University, Algeria, in 2018, and the M.Sc. degree in data science from Innopolis University, Russia. He is currently pursuing the Ph.D. degree with the Sorbonne Center of Artificial Intelligence, Paris, France. He is currently working as a Research Assistant with the Institute of Artificial Intelligence and Data Science, Innopolis University. His research interests include domain adaptation, computer vision, and deep learning.

DRAGOŞ CONSTANTIN NICOLAE received the B.S. degree in economics and the M.Sc. degree in econometrics from Academia de Studii Economice din Bucuresti, Romania, in 2004 and 2006, respectively, and the M.Sc. degree in data science from the City, University of London, in 2010. He is currently pursuing the Ph.D. degree with the Romanian Academy Research Institute for Artificial Intelligence ''Mihai Dragulescu.'' His research interests include machine learning, computer vision, and deep learning applied in natural language processing.

ADIL KHAN (Member, IEEE) received the B.S. degree in information technology from the National University of Sciences and Technology, Pakistan, in 2005, and the M.Sc. and Ph.D. degrees in computer engineering from Kyung Hee University, South Korea, in 2011. He is currently an Associate Professor with the Institute of Data Science and Artificial Intelligence, Innopolis University, Russia. His research interests include machine learning and deep learning.

S. M. AHSAN KAZMI received the master's degree in communication system engineering from the National University of Sciences and Technology (NUST), Pakistan, in 2012, and the Ph.D. degree in computer science and engineering from Kyung Hee University (KHU), South Korea. He is currently an Assistant Professor with the Institute of Information Security and Cyber Physical System, Innopolis University, Innopolis, Russia. His research interests include applying analytical techniques of optimization and game theory to radio resource management for future cellular networks. He received the Best KHU Thesis Award in engineering in 2017 and several best paper awards from prestigious conferences.

ASAD MASOOD KHATTAK (Senior Member, IEEE) received the M.S. degree in information technology from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2009, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2012. He is currently an Associate Professor with the College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates, that he joined in August 2014. He has worked as a Postdoctoral Fellow with the Department of Computer Engineering, Kyung Hee University, where he later joined as an Assistant Professor. He is currently leading three research projects, collaborating in four research projects, and has successfully completed five research projects in the fields of data curation, context-aware computing, the IoT, and secure computing. He has authored/coauthored more than 150 journal and conference papers in highly reputed venues. He has been serving as a Reviewer, a Program Committee Member, an Organizer, and a Guest Editor for many workshops, conferences, and journals. He has delivered keynote speeches, invited talks, guest lectures, and has delivered short courses in many universities. He and his team have secured several national and international awards in different competitions. He has been appointed as an ACM Distinguish Speaker.

BULAT IBRAGIMOV received the Ph.D. degree in electrical engineering from the University of Ljubljana, in 2014. He was a Postdoctoral Fellow with Stanford University, from 2016 to 2018, and the Senior Research Scientist of Auris Health, from 2018 to 2019. He is currently an Assistant Professor of machine learning with the University of Copenhagen and the Lead Research Scientist of Innopolis University. His research interests include the use of machine learning for abdominal radiotherapy planning, musculoskeletal pathology quantification, and lung pathology diagnosis. He is a winner of multiple international competitions on medical image analysis and computer-aided diagnosis.