Improving Cross-dataset Generalization in Image Classification with Contrastive Representation Learning

by

Najmeh Saffar

A Thesis Submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, MB, Canada

ABSTRACT

Regular monitoring of marine wildlife is essential for rapid detection of changes in the marine ecosystem allowing for adaptive strategies. However, the manual analysis of large volumes of underwater images taken by cameras is highly time-consuming. Deep learning techniques have been adopted in marine wildlife for the automatic classification of underwater photos to accelerate image analysis. However, water quality varies at different locations, depths, and acquisition times during data collection. This, along with differences in other acquisition parameters, leads to datasets with idiosyncratic footprints and, therefore, limited generalization of the trained deep learning model to other sets of images different from the training set. As a result, more work is required toward improving the cross-dataset generalization of deep learning models. In our research, we started by assessing dataset biases' impact on cross-dataset generalization in the classification of beluga whale images from empty underwater image frames. We used three underwater image datasets with varying image acquisition profiles: a dataset of good water quality photos, moderately bad water quality photos, and a dataset of images with both the horizon and water in the same frame. Then, we investigated two frameworks to improve cross-dataset generalization. One attempts to unlearn dataset-specific information for explicitly handling the dataset bias problem. The other uses a contrastive loss for learning a representation by contrasting the images with beluga whales against the images with empty frames regardless of their dataset membership. We conducted an exhaustive evaluation of proposed deep learning architectures and compared performance using cross-dataset approaches with traditional architectures. The supervised contrastive approach outperforms the other architectures. To the best of our knowledge, this was the first use of contrastive settings to implicitly address the dataset bias problem.

ACKNOWLEDGEMENTS

I would like to express my special appreciation to my advisor, Dr. Ahmed Ashraf, for his immense knowledge, invaluable patience, motivation and continuous support of my Master's study and research in my field of interest, deep learning. His thoughtful feedback and suggestions assisted me in all the research and writing of this thesis. Without his guidance and dedicated involvement, I would not have been able to accomplish my Master's thesis.

Besides my advisor, I would like to thank my co-advisor, Dr. Shehroz Khan, for his insightful comments and encouragement. I also appreciate my thesis committee, Dr. Pradeepa Yahampath and Dr. Matt Khoshdarregi, for their fantastic comments and suggestions during the defence. I take this opportunity to acknowledge the University of Manitoba for providing me the International Graduate Student Entrance Scholarship. Part of my graduate studies were supported by Dr. Ashraf's Startup Grant (provided by UM) and the Discovery Grant (provided by NSERC).

My sincere thanks also go to the Assiniboine Conservancy Park research group. It was an honour for me to work with these wonderful people, Ashleigh Westphal, Sarah Falconer, C-Jae Breiter, and Stephen Petersen. They provided me with valuable datasets, which formed a perfect case study for demonstrating the applicability of my research. The collaboration with the Assiniboine Conservancy Park was supported by Mitacs through the Mitacs Accelerate program.

Lastly, I am forever grateful to my family and friends for their love, support, and encouragement throughout my pursuit of education. I also deeply appreciate my only companion in life, Hossein, who has accompanied me through all the hardships and pleasant moments of my life.

DEDICATION

To my dear parents and my lovely sister

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
SVM	Support Vector Machine
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic Curve
VGG	Visual Geometry Group
SGD	Stochastic Gradient Descent
Adam	Adaptive Movement Estimation
HIHO	Half-in Half-out
LODO	Leave One Dataset Out
MSE	Mean Squared Error
NLLLoss	Negative Log Likelihood Loss
BCE	Binary Cross Entropy
SupCon	Supervised Contrastive Learning
AdvRep	Adversarial Representation

Chapter 1

Introduction

1.1 General Introduction

Deep learning is a subset of machine learning and AI techniques that enables computers to understand and represent the world as a hierarchy of concepts. As a representation-learning method, it allows a machine-learning system to transform the raw data (such as the image pixels) into multiple levels of representation (feature vectors), from more low levels to progressively higher levels of abstraction (e.g., starting from the presence of edges to more complex local shapes and textures). (Lecun, Bengio, and Hinton 2015; I. Goodfellow, Bengio, and Courville 2016). A deep learning network is essentially a multi-layer neural network that successively applies transformations to extract useful feature representations (each layer accepts, as input, the output from the previous layer). Two main types of deep learning techniques are supervised and unsupervised learning. In unsupervised learning, we directly learn from the input data without knowing about the category of the given input. Thus, we can only group similar data points. While, in supervised learning, we have a labelled dataset, and we can learn a function that maps the input to the labels (Bishop 2006).

Classification is one of the most common settings in supervised deep learning. In a classification task, we train a model to decide which class label each datapoint comes from. It has three subcategories: binary classification, where each sample takes only one label out of two classes; multi-class classification, where each sample takes only one label out of multiple classes;

multi-labelled classification, in which we can assign multiple labels concurrently out of all the existing classes to each sample (Krizhevsky, Sutskever, and Hinton 2017; Kim et al. 2019). In the process of training a model, we try to learn a function (classifier) that can classify the given input to several class labels in the best possible way. To learn that function, we need to evaluate the model iteratively and change the network's parameters (weights) accordingly. For this purpose, we must define a loss function to evaluate the difference between the predicted output and the actual label. Over the iterations, we try to minimize that loss function with respect to the choice of the network's parameter. There are two different optimization approaches for minimizing the loss: non-gradient-based (genetic algorithms, simulated annealing, etc.) and gradient-based (Ahmadianfar, Bozorg-Haddad, and Chu 2020). Non-gradient-based algorithms usually converge to a global optimum, while gradient-based algorithms usually achieve a local optimum. The gradient-based methods are the most widely used in deep learning algorithms, especially in tasks such as image classification, for which a large amount of data is needed to better generalize the model to the visual world. Gradient descent is one of the gradient-based methods. The computation of this method is based on the backpropagation learning algorithm. First, we calculate the gradients of the loss function with respect to each weight individually, and then we update the network's weights in response to the gradients. This process is done iteratively to minimize the loss function to its local minimum (I. Goodfellow, Bengio, and Courville 2016).

Convolutional Neural Networks (CNN) represent a class of deep neural networks that have shown state-of-the-art results for imaging inputs compared to multi-layer perceptron neural networks for a number of applications (Krizhevsky, Sutskever, and Hinton 2017). The commonly used type of CNN consists of three primary layers: convolutional layer, pooling layer and fully connected layer. Convolutional layers are responsible for capturing the important features of the images starting from low-level features in the first layers to high-level ones in later layers. Therefore, each convolutional layer has a weight matrix kernel (filter) smaller than the given input. It applies linear convolutional operations by calculating dot products multiple times between its filter and different filter-sized patches of the input. Then, it passes the convolved input through a nonlinear activation function before giving it to the next layer. The advantage of having filters with smaller sizes is being able to build a deeper network. As a result, we enhance the network's representation powers to implement more nonlinear functions while having less number of

parameters for each layer. Pooling layers can help to reduce the dimension of convolved features by taking the average or the maximum value from the portion of the image. The advantage of using a pooling layer is reducing the computational power to process the given input and suppressing the noise by dimensionality reduction, especially for the max pooling. After having a number of convolutional layers and pooling layers, we flatten the matrix of the last layer and feed it into a traditional multi-layer perceptron called fully connected layers. The output of the last fully connected layer has the same number of nodes as the number of classes. The number must be one if we have a binary classification. A nonlinear function follows every convolutional layer and fully connected layer. The most common one for all the layers is the rectified linear unit (ReLU) except for the last layer, for which we usually use the sigmoid or softmax function (Patil and Rane 2021).

We have various CNN architectures depending on the application, such as U-net (Ronneberger, Fischer, and Brox 2015) for image segmentation, or AlexNet (Krizhevsky, Sutskever, and Hinton 2017), VGG-16 (Simonyan and Zisserman 2015), ResNet50 (He et al. 2016), etc. for image classification. AlexNet has the fewest number of layers (five convolutional layers and three fully connected layers) compared to the other two architectures, VGG-16 and ResNet-50, while it uses larger receptive fields (Figure 1.1). VGG-16 has thirteen convolutional layers and three fully connected layers (Figure 1.2). ResNet-50 is the deepest network compared to the other two examples with forty-nine convolutional layers and one fully connected layer. This network is known as a residual network (Figure 1.3). A deeper network increases the efficiency of learning a more complex function; however, as a network goes deeper, performance will eventually drop due to vanishing gradient problems. Skip connection has been introduced in residual networks to address this.

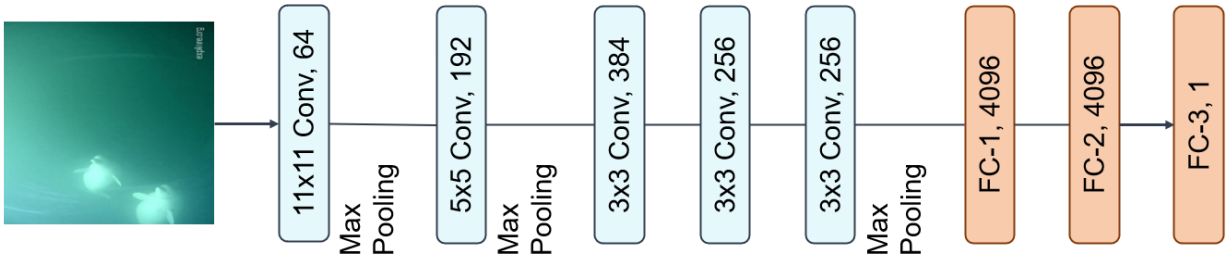

Figure 1.1. Schematic of the AlexNet convolutional neural network architecture.

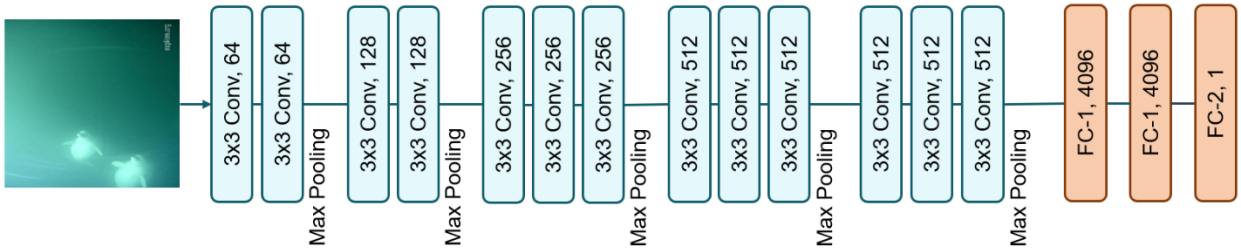

Figure 1.2. Schematic of a VGG-16 convolutional neural network architecture.

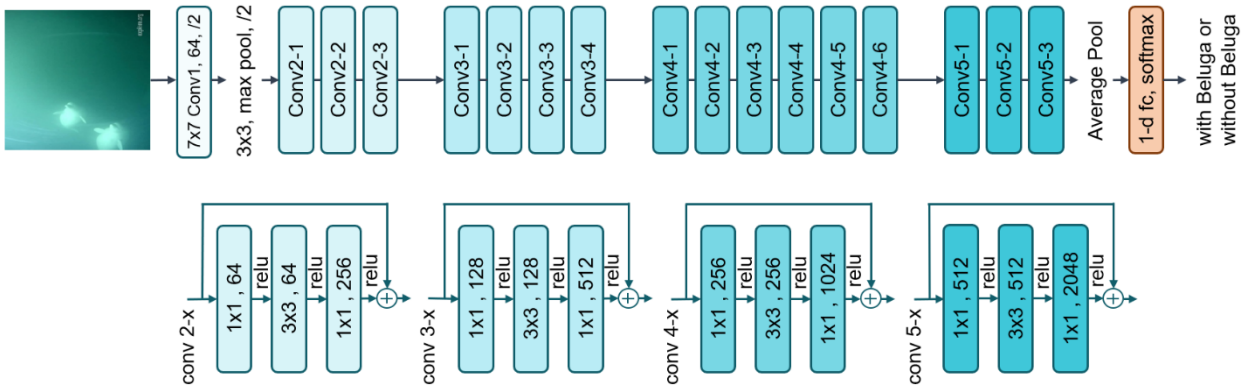

Figure 1.3. Schematic of a ResNet-50 architecture with skip connections, adding the input of each convolution block to its output.

Unlike the CNNs that are commonly used for spatial data such as images, recurrent neural networks (RNN) are other types of deep neural networks designed to analyze temporal, sequential data such as video, audio, text, etc. In sequential data, it is essential to consider the order of the data as each part of the data is related to the previous or the next data. For this purpose, special types of RNNs were introduced for sequential data, such as LSTM and GRU, that can remember the sequences or have control of what information to keep or throw out during the training process (Alex Graves 2012; Cho et al. 2014).

Deep learning models can either be trained for a task from scratch, or a pre-trained model on a large publicly available dataset, such as ImageNet, can be utilized to improve new models' training accuracy and speed. The latter approach is referred to as transfer learning (Ribani and Marengoni 2019), in which the extracted features of the images based on the pre-trained network can be transferred to the new task and do not have to be learned again. If the model is not pre-

trained, more labelled data is necessary for generalization. Though massive computation is required, giving more labelled data to these networks will help minimize the overfitting. Accordingly, the overall performance of the model for unseen data improves. Nevertheless, supposing data acquisition is challenging, data augmentation is suggested as a tool to better generalize the model (Krizhevsky, Sutskever, and Hinton 2017).

In general, access to large and diverse training data sets is a key requirement for training deep neural networks. Even though significant efforts are typically made to create a dataset representative of the real world, studies show that all datasets end up narrowing the real world to their own specific closed worlds to some extent (Torralba and Efros 2011). As a result, different datasets bring along their characteristic and idiosyncratic footprints, which prevent models trained on one dataset from generalizing to examples from different datasets. These dataset's footprints and their impact on cross-dataset generalization are referred to as the dataset bias problem (Torralba and Efros 2011; A. Khosla et al. 2012; Tommasi et al. 2013). Therefore, we need to develop methods to improve cross-dataset generalization by either unlearning dataset membership information or mitigating the dataset bias's impact on our classifiers.

The advances in AI and deep learning have increased our capability to solve problems in a variety of applications. One of the domains in which we can apply deep learning techniques is for monitoring wildlife. With the rapid improvement of camera and data storage technology, the wildlife image datasets have increased in size and quality. Even though these datasets provide opportunities to explore research questions and address them, they bring challenges such as including empty and non-target images. Manual image processing prevents us from utilizing these datasets efficiently. Therefore, deep learning models can be developed to increase data processing efficiency using CNNs to sort images containing the object of interest from empty frames. Applying deep learning algorithms can significantly reduce the labour and time costs of manual data processing, which can eventually allow researchers to focus their resources on more challenging tasks.

The presence of the biases in the image data collection can be due to several sources, such as changes in image capturing devices and environment changes, among others. These variations

limit the generalizability of trained deep learning models to new and different datasets. Since we aim to transfer the learned knowledge from one dataset to the other more efficiently, we can alleviate the bias by having the training focus on the related features of the variable of interest instead of the dataset-specific factors (Kim et al. 2019; Yamashita et al. 2019). One way to investigate the presence of biases within datasets is to train a classifier to predict the dataset membership of the images. If the image data collections contain biases, the performance of the dataset membership classifier should be better than chance. In other words, if the datapoints carry dataset-specific footprints, it should be easy to build a classifier to detect which dataset a datapoint comes from. The next step is evaluating the impact of biases in detection and classification tasks. If the biases affect the cross-dataset generalization of the classifiers, different frameworks and architecture should be investigated for improving the cross-dataset performances. We discuss the following techniques in this study to address the mentioned issue of the image data collection in training deep neural networks:

- Following the line of work by Madras et al.(Madras et al. 2018), we developed a model to learn a latent representation from the images such that the representation is capable of reconstructing the input image if combined with the sensitive attribute (dataset membership). The model contains one autoencoder, a classifier and an adversarial network. The encoder maps the input image to a latent representation, and the decoder reconstructs the input image from the representation and the sensitive variable. The classifier predicts the label from the representation, and the adversary predicts the dataset membership from the representation. This new representation aims to lose any information that can identify whether the image belongs to the specific dataset while retaining as much information as possible needed for the classification task.

- We investigate a contrastive approach for implicitly handling the dataset bias problem. Contrastive representation learning is a technique for improving the performance of the classification tasks by contrasting samples against each other to learn similar and dissimilar images. The contrastive paradigm started off with representation learning in a purely self-supervised fashion with no access to labels (Chen et al. 2020; Henaff et al. 2020; He et al. 2019; Wu et al. 2018). Recently, supervised contrastive methods have also been proposed,

such as the work by (P. Khosla et al. 2020; Kopuklu et al. 2021). In a supervised setting, they leverage label information to build a representation in a contrastive setup. In particular, we learned an embedding space using a contrastive loss, such that all the samples with the positive labels, regardless of their dataset membership, get pulled together and pushed apart from those with negative labels. We demonstrate that this approach can address the issue of the dataset bias problem by introducing a new loss function that leads to a better margin for testing the trained model on different datasets from the training data. The contrastive approach improves cross-generalization more in comparison to traditional methods and the work by (Madras et al. 2018).

1.2 Background

The presence of biases in image data collection has become a well-known issue in the computer vision community. In 2011, the seminal work by Torralba et al. (Torralba and Efros 2011) aimed to raise awareness in the object recognition research community about the critical issue of the built-in biases of the datasets and how they adversely affect the performance of the detection and classification tasks. To demonstrate the problem, they randomly sampled 1000 images from the training portion of each of the twelve widely used recognition datasets to train a classifier to play a game called "Name That Dataset!". Interestingly, the classifier performed with a classification accuracy of 39%, notably better than chance (1/12 = 8%). They also did the same experiments with more training data to show that the classification accuracy could be increased without immediate saturation. By visualizing the confusion matrix grouped by similarity, they demonstrated that each dataset had its distinctive signature (Figure 1.4). To confirm the idea, they attempted to alleviate the biases by isolating specific objects of interest from the images of five datasets. Surprisingly, the classifier still was able to separate the datasets with an accuracy of 61%, better than a 20% chance. Their experiments went on to show that biases inadvertently continue to persist in datasets despite efforts to minimize them. The paper suggested that a clear understanding of the types and sources of bias could result in developing better datasets while minimizing each type of bias. A good-quality dataset can be employed later to build algorithms that can perceive the visual world (Torralba and Efros 2011).

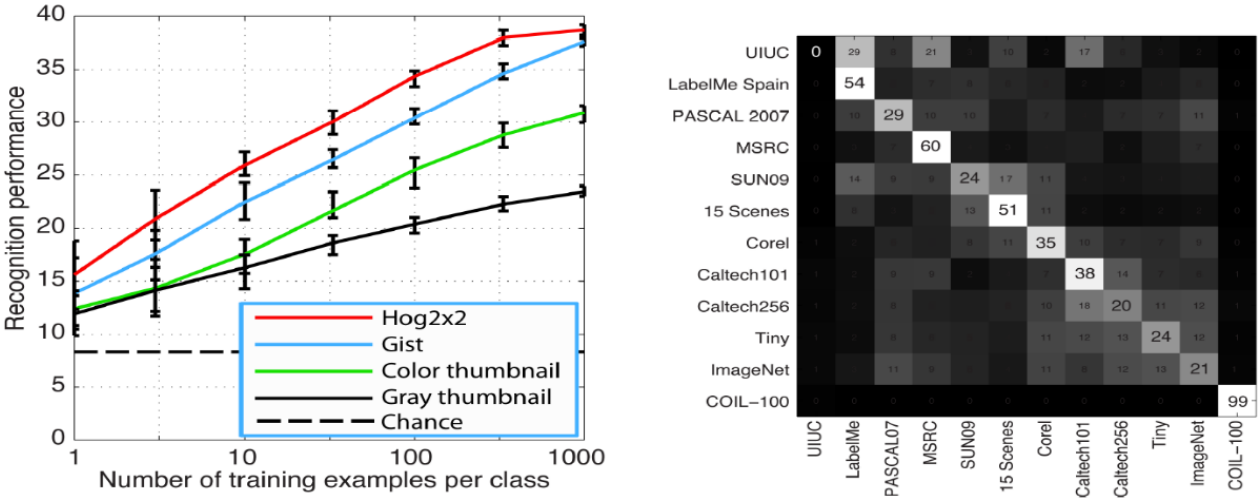

Figure 1.4. "Name That Dataset" game. Left: classification performance as a function of log scale of dataset size. Right: confusion matrix (Torralba and Efros 2011) (permission granted).

Since it is practically difficult to keep datasets entirely unbiased, one does not necessarily increase the generalization ability of an algorithm by adding more training data with biased data points, as shown in (Torralba and Efros 2011). Experiments in (Torralba and Efros 2011) have demonstrated the existence of different types of bias (e.g. selection bias, capture bias, and negative set bias) in popular image datasets. Therefore, there is a need for methods which still aim to minimize cross-dataset performance drop by still using datasets which are not entirely free of biases. One of the first works toward this end was by Khosla et al. (A. Khosla et al. 2012), proposing an algorithm for undoing the dataset's bias to mitigate its adverse effects and thus evaluating the algorithm with respect to cross-dataset generalization performance. They noted that despite various biases in each dataset, images of each dataset are biased samples of a more general dataset called the visual world. Hence, they proposed a discriminative framework for learning a support vector machine (SVM) classifier (Cortes and Vapnik 1995) using images of multiple datasets and decomposing the SVM weight into visual world vector common among all datasets and the dataset-specific bias vectors (Figure 1.5). According to their study, models based on the learned common weight vector were shown to perform well on a new dataset when the target task was the same as the source task. Additionally, they demonstrated that the learned bias vector indicated membership of the images to a particular dataset by training a classifier using the bias vectors to predict each image belongs to which dataset (A. Khosla et al. 2012).

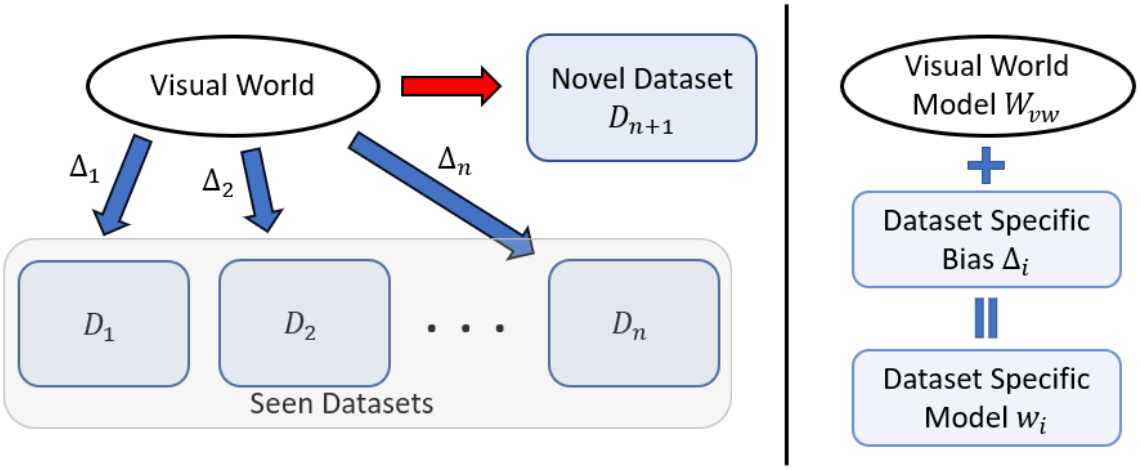

Figure 1.5. Datasets that are sampled from the visual world. One of them is used for a test set and has not been seen by the model. The model jointly learns a visual world vector and the bias vector (A. Khosla et al. 2012).

Another study was done by Tommasi et al. (Tommasi et al. 2013), which introduced an algorithm called Multi-task Unaligned Shared Knowledge Transfer (MUST), similar in spirit to work by (A. Khosla et al. 2012). They improved the cross-dataset generalization performance by learning an image representation that decomposes into two parts, one specific to each dataset and the other shared between all the datasets. Unlike the study (A. Khosla et al. 2012), they generalized the dataset bias problem to multi-class instead of binary. They mentioned that analyzing one class at a time and considering the remaining set of classes as a negative class (in datasets with more than one class) implies the task of binary classification, but this definition of "what an object is not" is intrinsically biased. They exploited the valuable knowledge of the datasets and demonstrated cross-dataset generalization of the MUST algorithm in multi-label classification tasks via a one-dataset-out strategy. Additionally, it was noted that common information in the shared space could be transferred to assist a new task and learn only the remaining private part of the new dataset. This approach is called transfer learning, where the target domain's feature space is different from the source feature space. If the source and target domain have the same feature space but different distributions, the approach can be referred to as domain adaptation, a subcategory of transfer learning (Sun, Shi, and Wu 2015). Domain adaptation can be classified into three groups depending on whether the target data have labels or not: unsupervised (set of unlabeled target examples), semi-supervised (set of labelled target examples) or supervised (all the

examples are labelled) (Saenko et al. 2010). Dataset bias is defined as a particular domain shift problem where we have different datasets with common classes (Tommasi et al. 2013). In another paper by Tommasi et al. (Tommasi et al. 2017), the authors look deeply at dataset bias as a domain shift problem in the CNN-based features field. They demonstrated how dataset bias limits the generalization of the trained models across different datasets and how the issues arising from the dataset bias can be addressed.

The study by Zemel et al. (Zemel et al. 2013) introduced an algorithm for fair classification. In this paper, fairness is defined as an optimization problem of finding a representation of the input with respect to the classification task which has the necessary information as well as possible (i.e., information about the individual's attributes) while ignoring any information about membership in the protected subgroup (e.g., race or gender). Based on the new representation, they made a fair classification. They showed positive results using their algorithm on three datasets. For one of the datasets, they classified the bank account holders into credit classes good or bad. They described each person by 20 attributes and considered age as the sensitive attribute. For this setup, the decision was made based on the individual's attributes except for age. They noted that the degree to which the system succeeded in ignoring the age information was evaluated by building a classifier that learns the age from the new representation. The other advantage of their approach is benefiting the intermediate representation for other classification tasks referred to as transfer learning.

Another work has been done by Ashraf et al. (Ashraf et al. 2018), inspired by the methods in the previous work (Zemel et al. 2013). They proposed a method for medical imaging classification on multiple datasets that learns a latent representation in which the data points are indiscernible in terms of the dataset membership information while allowing classification with respect to the variable of interest. Their method to unlearn dataset membership is referred to as handling dataset bias for generalizing the trained model well to the other dataset. They extended the previous work to multi-class problems instead of binary and modified the objective function to deal with any number of datasets.

The risk of biases towards certain groups, specifically in sensitive fields such as medical diagnosis, has brought attention to fair representation learning to mitigate the impact of biases. In this learning technique, the original domain of features is mapped to a latent domain, upon which the information about sensitive attributes got forgotten. A paper by Madras et al. (Madras et al. 2018) is motivated by the idea of fair classification using adversarial learning to learn a fair representation that guarantees performance on metrics of group fairness. Furthermore, they demonstrated that using their model for fair transfer learning under certain conditions is possible.

1.3 Underwater Wildlife Datasets

Three datasets of underwater images were studied for cross-dataset generalization, including a dataset of good water quality photos, bad water quality photos, and photos that contained images with both the horizon and water in the same frame (henceforth half-in and half-out (HIHO) dataset). All these underwater images were collected between 2016-2021 to track and monitor beluga whales and their marine ecosystem in the Churchill 130 River estuary near Churchill, Manitoba, Canada. The Churchill River estuary has a complete freeze-thaw cycle, so the water quality in the estuary changes based on the influx of water in the spring melt. In 2017 there was a large flood in the spring, which brought an influx of water into the estuary. This resulted in a dataset with bad water quality images, mostly with the murky background in 2017 and 2018 due to suspended sediment and silt in the water column. The sediment comes from the erosion of the riverbanks and surrounding areas in high flood years. In 2016, 2019 and 2021, it was a regular thaw cycle in the spring (without flooding), so the water quality was much clearer resulting in higher water quality images. Photos from these years made up the good water quality photo dataset. The HIHO dataset was collected in 2020, the first year that Polar Bears International (the organization that runs the Beluga Boat) used a different boat for video collection. Because of this, the camera was not mounted deep enough in the water resulting in the "half-in and half-out" photos as the boat hit swells. In some of these photos, part of the boat is within the picture. These three datasets thus represent three very different acquisition profiles, although they aim to capture images of more or less the same set of objects (underwater marine life). As such, they provide a

very useful testbed for assessing an algorithm's intrinsic capability of detecting the objects of interest rather than aligning to specific acquisition conditions and biases of particular datasets.

All the images were extracted from an underwater camera that captured video footage below the surface. The video was subsampled at a rate of one frame every three seconds (2016-2020) or one frame per second (2021) to get the images. Then all the images were classified by participants on Beluga Bits, a citizen science project hosted by Zooniverse (Zooniverse.org). The photos were classified by citizen scientists based on quality, whether a beluga is present, and content. Each image was classified by a minimum of 10 participants. The photo datasets were created by aggregating the citizen scientist's responses and selecting images from the target years with a minimum of 80% agreement on the presence or absence of a beluga. The number of images for each dataset was written in Table 1.1. Extracting frames from the video produces a large number of images that do not contain species of interest. This lack of beluga images typically reduces the level of participation of citizen scientists for further analysis. Developing CNNs to sort frames that contain beluga whales from empty (just water) images can increase data processing efficiency and maintain participant interest. However, the water quality each year might be different. Consequently, the trained classifier on one dataset might not perform well once tested on different water quality images. Accordingly, we addressed this issue using the three available datasets discussed above.

Table 1.1. The number of images for each available dataset

Datasets	Good water quality photos list	Moderately bad water quality photos list	HIHO photos list
Number of photos	6000	3140	3804

1.4 Thesis Organization

The remainder of the thesis is organized as follows: In chapter two, we quantify the prevalence of dataset bias in our underwater datasets by attempting to build a three-class dataset membership classifier to test if each dataset leaves specific footprints on its images. In chapter three, we investigate the potential negative impact of dataset bias by assessing the performance of traditional classifiers in a leave-one-dataset-out sense. In chapter four, we proposed an adversarial approach to build an unbiased representation that reduces the adverse impact of biases in the images. In chapter five, a supervised contrastive learning technique is used to improve the trained model's cross-dataset generalization to a different dataset but with the same variable of interest. In the last chapter, we conclude our work and discuss possible future work directions.

Chapter 2

Dataset Bias Detection

2.1 Dataset Membership Classification

If there are no dataset-specific biases, given images from different datasets but having the same context and objects, it should be difficult to train a dataset membership classifier. As a result, one way of quantifying the presence of dataset biases is to train a classifier to distinguish what dataset a date point comes from (Torralba and Efros 2011; Ashraf et al. 2018). We did the experiments based on three balanced sets of underwater wildlife datasets; each set contains some photos with at least one beluga whale, while some photos do not have beluga whales. The number of images with and without beluga is shown in Table 2.1. All the images from three datasets of D_1, D_2 and D_3 were selected to train and test a three-class classifier telling each image belongs to which dataset regardless of whether they include beluga whales. If the classifier performs better than chance (1/3 = ~33.3%) on both training folds, we can infer that dataset membership information (dataset biases) exists in each frame. We used cross-validation to evaluate the generalization of our trained models. Cross-validation involves dividing the dataset into partitions (commonly referred to as folds), wherein multiple models are trained and assessed by letting different folds assume the role of training and validation sets (Bishop 2006). We carried out the two-fold cross-validation (Figure 2.1) by randomly shuffling the images from all three datasets into two subsets of equal size, designated as F_0 (first fold) and F_1 (second fold). We then trained deep learning models on F_0 and validated it using F_1, followed by training on F_1 and validating on F_0.

Table 2.1. Number of images with and without beluga whales in three datasets of D_1, D_2 and D_3

Datasets	Good water quality photos list (D_1)	Moderately bad water quality photos list (D_2)	HIHO photos list (D_3)
Images with belugas	3,000	1,570	1,902
Images without belugas	3,000	1,570	1,902
Total	6,000	3,140	3,804

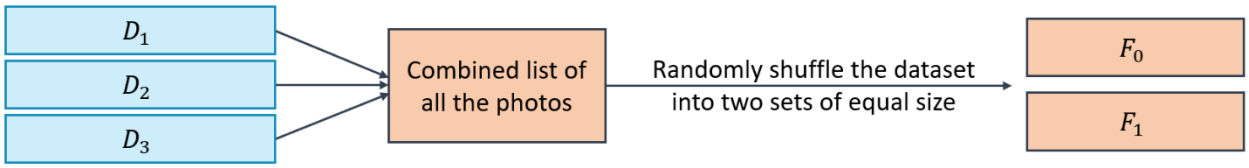

Figure 2.1. Two-fold cross-validation based on the images from all three available datasets.

We employed a convolutional neural network VGG-16 (Simonyan and Zisserman 2015) for our three-class dataset membership classifier. Moreover, we incorporated attention mechanisms to further improve the performance (Jetley et al. 2018) (Figure 2.2). Attention mechanisms in deep neural networks are a class of methods through which the neural network can learn to pay attention to certain parts of the image based on the context and the task. We can gain insight into where the model focuses when it predicts the class label of the photo frames by looking at their attention estimator output.

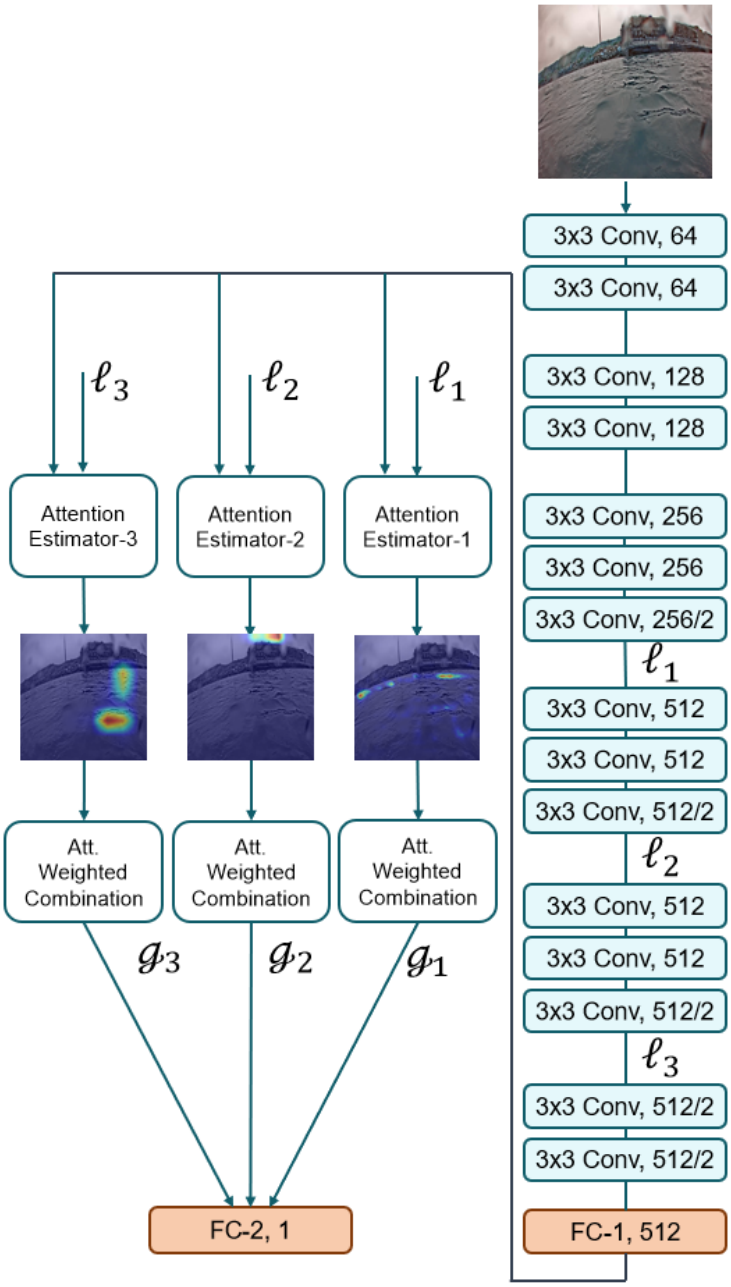

Figure 2.2. The architecture of VGG-16 with the attention module. The attention estimator masks shown above demonstrate that the neural network learns to pay attention to those regions of the image which are distinguishable in each dataset.

2.2 Results and Discussion

The dataset membership classifier was trained on both fold F_0 and F_1 and tested on F_1 and F_0 respectively. The accuracies on each fold are reported in Table 2.2, which are significantly better than chance (1/3 = ~33.3%). The confusion matrix of the classifier, when tested on F_1 is in Table 2.3, and the confusion matrix, when tested on F_0 are in Table 2.4. We noticed a pronounced diagonal in confusion matrices, indicating that each dataset possesses a unique, identifiable signature which we call dataset membership information. We can gain insight into where the model focuses when making predictions by looking at the attention estimator output of the image frames, paying particular attention to images the network struggles to identify its membership to a particular dataset or images assigned to a dataset with a high probability (Figure 2.3). As such, the results of these experiments suggest a strong presence of dataset-specific biases in the images.

Table 2.2. Results summary for the dataset membership classifier. The dataset is divided into two folds where the training process is performed on one of the folds and tested on the other fold. Reported are the class accuracy (%) of all three datasets (D_1, D_2 and D_3) and the overall accuracy (%).

Architecture	Training Fold	Class Accuracy of D_1 (%)	Class Accuracy of D_2 (%)	Class Accuracy of D_3 (%)	Overall Accuracy (%)
VGG-16	F_0	95.16	82.99	96.58	92.62
VGG-16	F_1	97.66	89.42	97.00	95.47

Table 2.3. Confusion matrix for the dataset membership classifier (trained on F_0 and tested on F_1)

		Predicted			
		D_1	D_2	D_3	Total
Actual	D_1	2855	57	88	3000
	D_2	231	1303	36	1570
	D_3	49	16	1837	1902
	Total	3135	1376	1961	6472

Table 2.4. Confusion matrix for the dataset membership classifier (trained on F_1 and tested on F_0)

		Predicted			
		D_1	D_2	D_3	Total
Actual	D_1	2930	17	53	3000
	D_2	122	1404	44	1570
	D_3	48	9	1845	1902
	Total	3100	1430	1942	6472

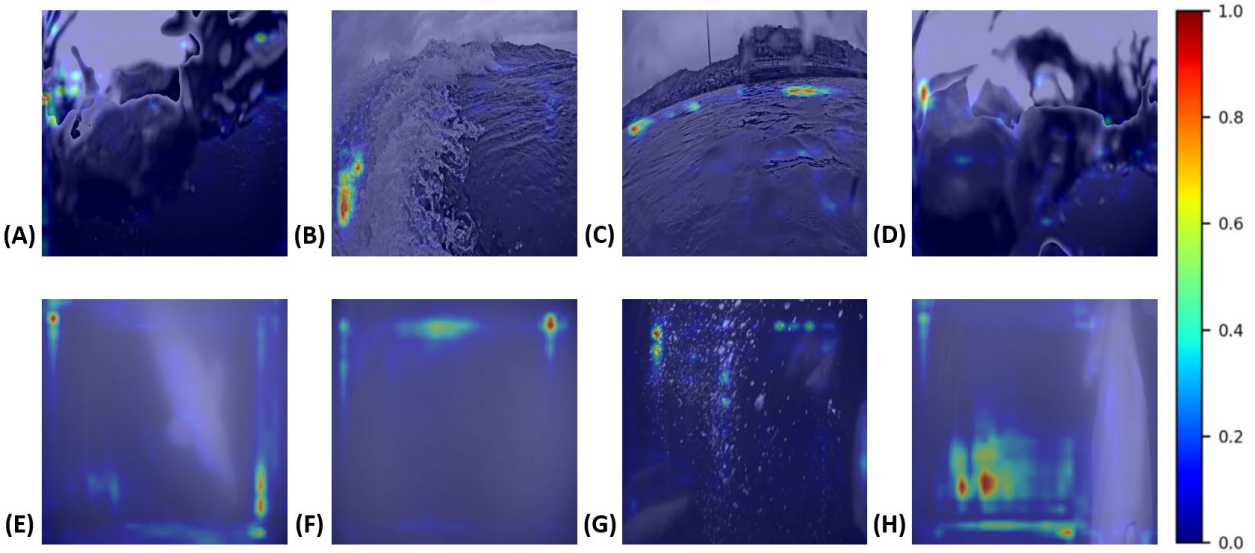

Figure 2.3. Panel A-D are the example frames that are correctly assigned to D_3 dataset with high probability. We can observe that part of these images is above sea level. That confirms the fact that D_3 contains a significant number of images above the sea level compared to the other two datasets. Panel E-H are the example frames that are correctly assigned to D_1 dataset with high probability. These are the images under the water. The colour bar defines regions with varying attention values, where the blue and red ends of the spectrum signify lower and higher attention values, respectively.

Chapter 3

Baseline using Traditional Methods

3.1 Introduction

In this chapter, we establish a baseline using a traditional CNN approach and assess both within-dataset and across-dataset performance for the task of detecting beluga whales in an image frame. As stated before, biases can show up due to a variety of sources, such as changes in cameras, acquisition settings, ocean environment, time of the day, time of the year, etc. Those variations may limit the generalization of trained deep learning models to other sets of images compared to when the same image data collection is used for training and testing. In the previous chapter, we demonstrated that each of our three available image data collections does carry dataset-specific biases in their image frames. For within-dataset performance, the training and testing subsets come from the same data collection. For cross-dataset performance, images from one dataset are held back as the testing set while the model is trained on the other two datasets, and this process is repeated three times, wherein each time, the held back dataset is changed. This type of strategy is referred to as the leave-one-dataset-out (LODO) approach. Throughout the thesis, we follow a LODO strategy for quantifying the cross-dataset generalization of our models. The procedure of evaluating deep learning models is to fit and assess them on training data, then verify that the model has good skills on a test dataset. The training and the test data might not come from the same image data collection. Therefore, each may contain biases due to the variations such as changes in cameras, ocean environment, time of the day, time of the year, etc. Those variations may limit the generalization of trained deep learning models to other sets of images compared to

when the same image data collection is used for training and testing. In previous chapters, we demonstrated that each of our three available image data collections holds dataset membership information (biases) in their image frames. In this chapter, we investigated if these biases affected the performance of the trained model on a held-back collection of images.

3.2 Experiment

For within-dataset evaluation, we select the training and testing sets from the same dataset to observe the performance of our initial classifiers. To this end, we split each of our three datasets (D_1, D_2, D_3) into two sets D_{ia} and D_{ib} with equal size, as shown in Figure 3.1. For each dataset, we learned two individual classifiers. One was trained on the first half of the dataset (D_{ia}) and tested on the second half (D_{ib}), and the other was done in reverse.

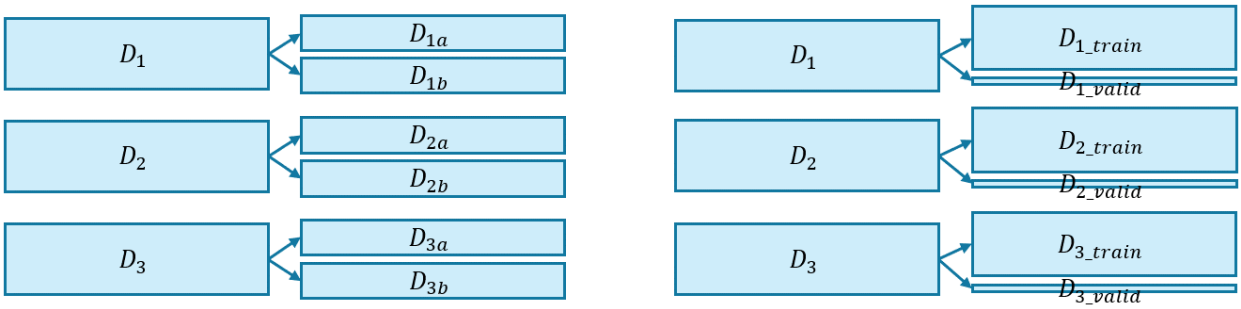

Figure 3.1. Left: two-fold cross-validation with 50:50 ratio. Right: splitting the datasets into a 90:10 ratio to set aside 10% of the data for validation purposes.

In cross-dataset evaluation, we considered a validation set for tuning the parameters of the classifiers (e.g., optimizer, learning rate, batch size, epochs, etc.). Every dataset is split into a 90:10 ratio such that 90% of the data (D_{i_train}) reserved for the training process and 10% of the data (D_{i_valid}) set aside for validation purposes. The training and testing data in all the cases are demonstrated in Figure 3.1. We followed a LODO approach as follows: for each LODO experiment, learning was done on the basis of two datasets, while testing was done on the left-out dataset. For testing, we used both halves of the left-out dataset, D_{ia} *and* D_{ib}, individually to make the result of cross-dataset evaluation comparable to the case of within-dataset evaluation. All

possible cases for within-dataset and cross-dataset evaluations are represented in Table 3.1 and Table 3.2, respectively. Since our focus in this study was not to work on enhancing the within-dataset performance, we did not consider a validation set for within-dataset evaluation.

Table 3.1. Within-dataset evaluation

Within-dataset (trained on the 1st fold)		Within-dataset (trained on the 2nd fold)	
Train Set	Test Set (Tested on 2nd fold)	Train Set	Test Set (Tested on 1st fold)
D_{1a}	D_{1b}	D_{1b}	D_{1a}
D_{2a}	D_{2b}	D_{2b}	D_{2a}
D_{3a}	D_{3b}	D_{3b}	D_{3a}

Table 3.2. Cross-dataset evaluation

Cross-dataset			
Train Set	Validation Set	Test Set (Tested on 2nd fold)	Test Set (Tested on 1st fold)
D_{2_train} - D_{3_train}	D_{2_valid} - D_{3_valid}	D_{1b}	D_{1a}
D_{1_train} - D_{3_train}	D_{1_valid} - D_{3_valid}	D_{2b}	D_{2a}
D_{1_train} - D_{2_train}	D_{1_valid} - D_{2_valid}	D_{3b}	D_{3a}

We used a VGG-16 network with a binary cross-entropy loss to establish a traditional baseline. The VGG models were trained using a stochastic gradient descent (SGD) optimizer (Kingma and Ba 2015) with 100 epochs, an initial learning rate of 0.001 and a weight decay of 1e6. The models are trained on 4 Quadro RTX 8000 GPUs with a batch size of 64 (Bishop 2006).

We implemented our code in PyTorch for all experiments and used the same hardware platform with an Intel Core i9 10th generation processor with 256 GB DDR4 RAM and 48 GB NVIDIA Quadro RTX 8000 GPU.

3.3 Results and Discussion

We evaluated the models using two performance metrics: classification accuracy and ROC-AUC (Receiver Operating Characteristic Curve – Area Under the Curve). We used a confusion matrix (Table 3.3) for our binary classification to calculate two types of accuracy: overall accuracy and per-class accuracies. Overall accuracy measures how many samples, both beluga and non-beluga samples, were classified correctly. We can calculate it by dividing the number of samples in the test set that were predicted correctly by the total number of the dataset (Eq. (1)). Per-class accuracy is measured for both positive (beluga images) and negative (non-beluga images) classes. We calculated the per-class accuracy of positive class (sensitivity or recall or true positive rate) by dividing the number of correct positive predictions by the total number of positives (Eq. (2)) and of negative class (specificity or true negative rate) by dividing the number of correct negative predictions divided by the total number of negatives (Eq. (3)).

Table 3.3. Confusion matrix of binary classification. TP and TN are the numbers of beluga and non-beluga images, respectively, that are classified correctly. FN and FP are the numbers of beluga and non-beluga images, respectively, that are classified wrongly.

		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

$$Total\ ACC = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$PerClass\ ACC\ (Positive\ Class) = SN = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{2}$$

$$PerClass\ ACC\ (Negative\ Class) = SP = \frac{TN}{TN + FP} = \frac{TN}{N} \tag{3}$$

$$False\ Positive\ Rate = \frac{FP}{TN + FP} = \frac{FP}{N} \tag{4}$$

ROC curve, as shown in Figure 3.2, is produced by plotting the TPR (True Positive Rate) (Eq. (2)) against the FPR (False Positive Rate) (Eq. (4)) for different thresholds on the binary classifier's output (scores). The scores are the predicted probability of the samples. Typically, we consider 0.5 as a threshold to classify a sample as positive if its score is above the threshold and

as negative if its score is below the threshold. The accuracy metric can be calculated by classifying the samples on that specific threshold. However, the ROC curve is not dependent on the threshold and classifies the samples based on any threshold between 0 and 1. The area under the ROC curve (AUC) measures how good a classifier performs independently from the threshold. The perfect classifier is when all the samples are correctly classified. This way, we have an AUC equal to one.

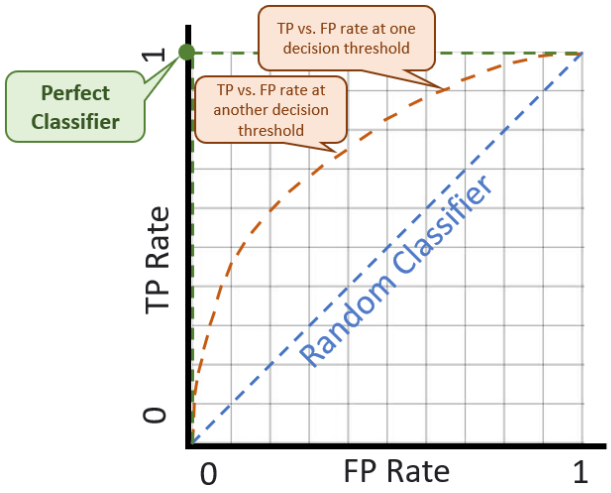

Figure 3.2. ROC-AUC plot: The blue dotted line represents a classifier that is not better than random guessing. The green dot represents a perfect classifier

We reported the accuracies and the AUCs of the trained classifiers for both within-dataset and cross-dataset evaluations on each fold in Tables 3.4 and 3.5.

Table 3.4. Within-dataset evaluation (For the first three rows, tested on the 2nd fold and for the second three rows, tested on the 1st fold)

Within-dataset Evaluation				
Train Set	Test Set	Class Accuracies	Overall Accuracy	AUC
D_{1a}	D_{1b}	[95.73% 92.68%]	94.23%	0.9851
D_{2a}	D_{2b}	[79.20% 96.21%]	87.51%	0.9707
D_{3a}	D_{3b}	[97.56% 96.87%]	97.21%	0.9957
D_{1b}	D_{1a}	[93.50% 93.63%]	93.56%	0.9824
D_{2b}	D_{2a}	[91.91% 93.15%]	92.54%	0.9706
D_{3b}	D_{3a}	[98.54% 98.83%]	98.68%	0.9972

Table 3.5. Cross-dataset evaluation (For the first three rows, tested on the 2nd fold and for the second three rows, tested on the 1st fold)

Cross-dataset Evaluation				
Train Set	Test Set	Class Accuracies	Overall Accuracy	AUC
D_{2_train} - D_{3_train}	D_{1b}	[93.49% 78.80%]	86.26%	0.9390
D_{1_train} - D_{3_train}	D_{2b}	[73.59% 93.61%]	83.37%	0.9322
D_{1_train} - D_{2_train}	D_{3b}	[21.31% 97.28%]	59.62%	0.7496
D_{2_train} - D_{3_train}	D_{1a}	[94.38% 76.95%]	85.53%	0.9314
D_{1_train} - D_{3_train}	D_{2a}	[72.35% 94.39%]	83.63%	0.9339
D_{1_train} - D_{2_train}	D_{3a}	[20.75% 96.92%]	58.51%	0.7509

When tested on the left-out dataset, the performance of the classifiers in all LODO experiments drops as compared to the within-dataset performance on the corresponding test set. This further demonstrates the presence of dataset bias and that it can significantly impact the performance when evaluated on a left-out dataset. The degradation in the across-dataset performance might be due to changes in the image quality of the datasets or the views and angles of the photos taken. We have the same trend on both folds, confirming the obtained results. In the third experiment, when the left-out dataset is the HIHO photos list (D_3), the performance degrades more noticeably compared to the other two experiments. Poor generalization across datasets might be because of the nature of D_3 as it is most different from the other two datasets.

The last linear layer's output of our convolutional neural network, known as scores, can be provided for all the given images from the testing set. We plotted the bar graph (histogram) of the frequency distribution of the scores by splitting them into small equal-sized bins to compare the overlap of the scores in within-dataset evaluation against cross-dataset evaluation. Figures 3.3, 3.4 and 3.5 are the histogram plots for the 2nd fold of different testing sets. Figures 3.6, 3.7 and 3.8 are the histogram plot for the 1st fold of the testing sets. The histograms on both folds show that scores overlap more in cross-dataset evaluations than when the training and testing samples are from the same dataset explaining the incidence of more confusion in classification for cross-dataset experiments.

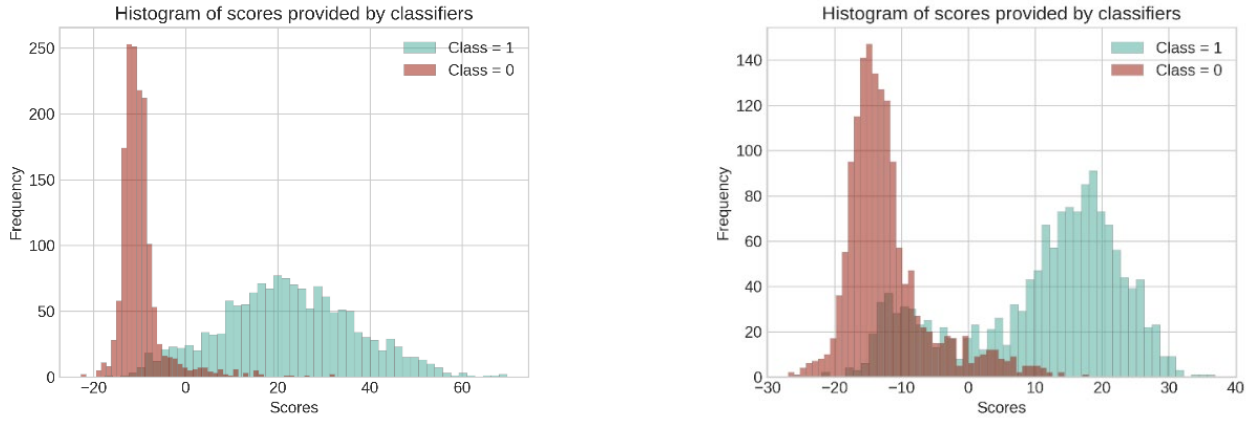

Figure 3.3. Histogram plot for the frequency distribution of scores provided by the classifier (when tested on the 2nd fold of good water quality photos list (D_1)). Left: within-dataset evaluation. Right: cross-dataset evaluation.

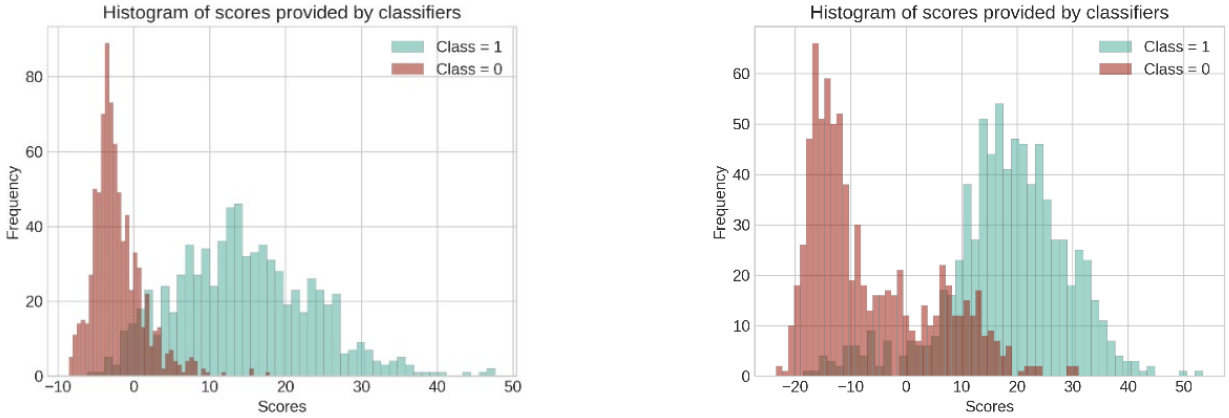

Figure 3.4. Histogram plot for the frequency distribution of scores provided by the classifier (when tested on the 2nd fold of moderately bad water quality photos list (D_2)). Left: within-dataset evaluation. Right: cross-dataset evaluation.

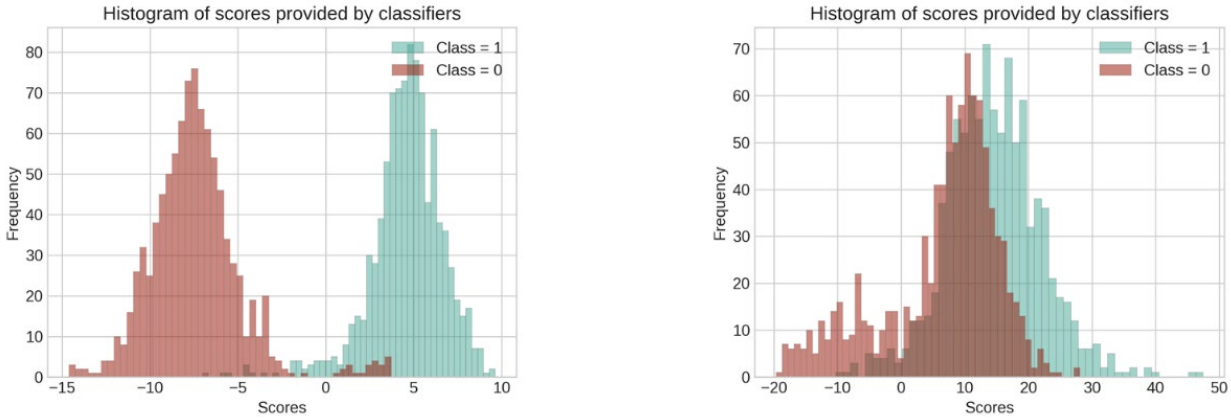

Figure 3.5. Histogram plot for the frequency distribution of scores provided by the classifier (when tested on the 2nd fold of HIHO photos list (D_3)). Left: within-dataset evaluation. Right: cross-dataset evaluation.

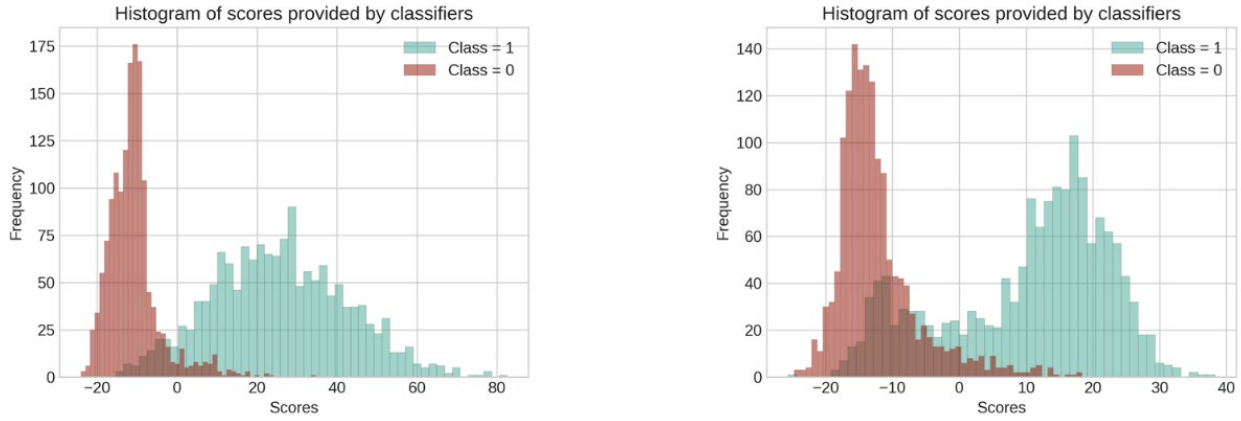

Figure 3.6. Histogram plot for the frequency distribution of scores provided by the classifier (when tested on the 1st fold of good water quality photos list (D_1)). Left: within-dataset evaluation. Right: cross-dataset evaluation.

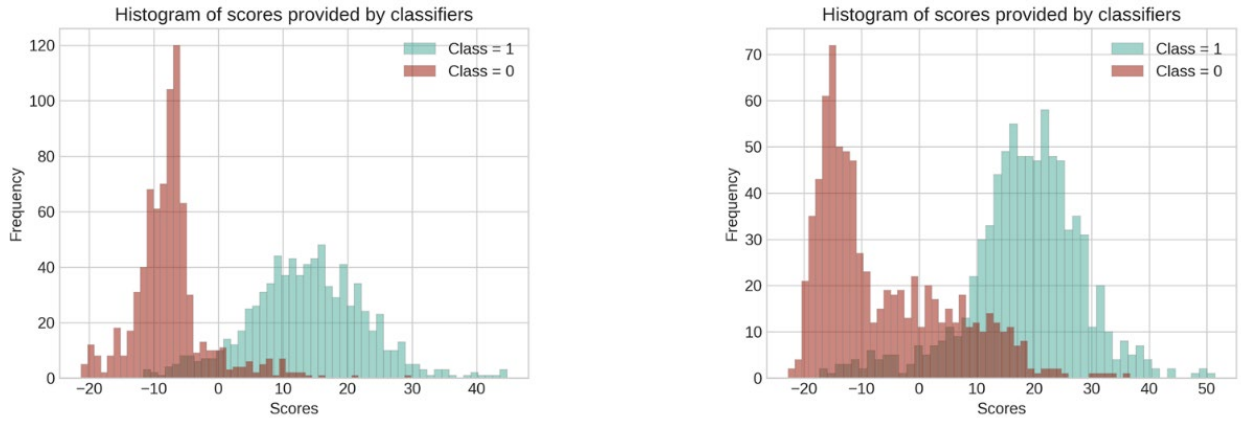

Figure 3.7. Histogram plot for the frequency distribution of scores provided by the classifier (when tested on the 1st fold of moderately bad water quality photos list (D_2)). Left: within-dataset evaluation. Right: cross-dataset evaluation.

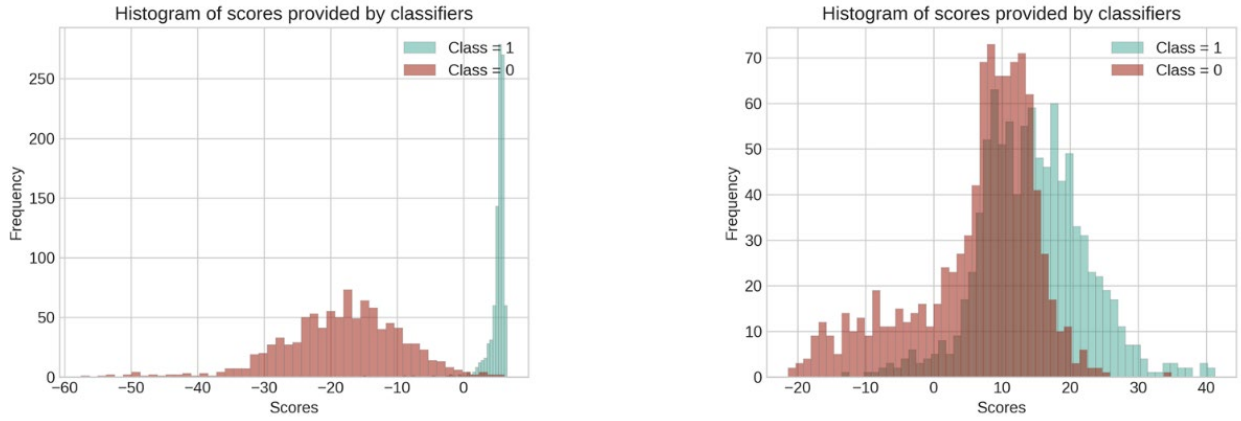

Figure 3.8. Histogram plot for the frequency distribution of scores provided by the classifier (when tested on the 1st fold of HIHO photos list (D_3)). Left: within-dataset evaluation. Right: cross-dataset evaluation.

Chapter 4

Learning Adversarially Unbiased Representation

4.1 Introduction

In the previous chapter, we demonstrated that dataset membership bias impacts cross-dataset generalization and can significantly affect the performance of a classifier on a held-back image data collection. Therefore, we investigated an approach for cross-dataset generalization to overcome the limitations of our baseline model. To do so, we attempted to learn a representation that allows making decisions by concentrating on the variable of interest (e.g., whether an image contains beluga whales or not) while preventing the algorithm from inferring a dataset membership. In recent years there has been a considerable amount of work for learning 'fair' representations, which make it harder to infer a sensitive variable (e.g., race, age) from the latent representation while preserving as much information as possible (Zemel et al. 2013; Madras et al. 2018). Similar ideas can enable solutions for the database bias problem by letting the database membership assume the role of sensitive information and attempting to learn latent representations that 'forget' dataset membership but maintain information about the variable of interest (Ashraf et al. 2018). In this context, we begin by briefly reviewing the work from Madras et al. 2018. They motivate the need for learning fair representations from the perspective of a data owner who wants to sell representations to a predictor vendor while concerns about unfairness in the predictions. For instance, the unfair prediction can be the scenario of an algorithm for selecting a candidate based on the resume but ends up selecting a candidate based on race. To prevent the algorithm from making such a biased choice, they proposed to learn a latent representation that ignores sensitive

variables, such as race, age, gender, etc. while still preserving the rest of the information to the extent possible. The same work can be applied to the dataset bias problem by learning a particular representation or model that can learn to dismiss dataset membership as a sensitive variable but can concentrate on the variables of interest. Inspired by Madras et al.'s work on fair representations, in the following we present a formulation for unlearning dataset membership in an adversarial setting.

4.2 Bias Unlearning through Adversarial Framework

The entire motivation of the bias-unlearning method was to learn a representation from an input image to have an accurate prediction based on that space with respect to the variable of interest but not biased in favour of the information specific to the datasets. Therefore, we built a framework (Figure 4.1), which attempts to learn a data representation $Z \in R^m$ capable of reconstructing the image input $X \in R^n$, classifying the target labels (with or without the beluga) $Y \in \{0, 1\}$, and taking care of the sensitive variable (dataset memberships) $A \in \{0, 1, 2\}$ by an adversary. This framework consists of three units: dataset membership unlearning, variable of interest classification, and autoencoder, which are explained below.

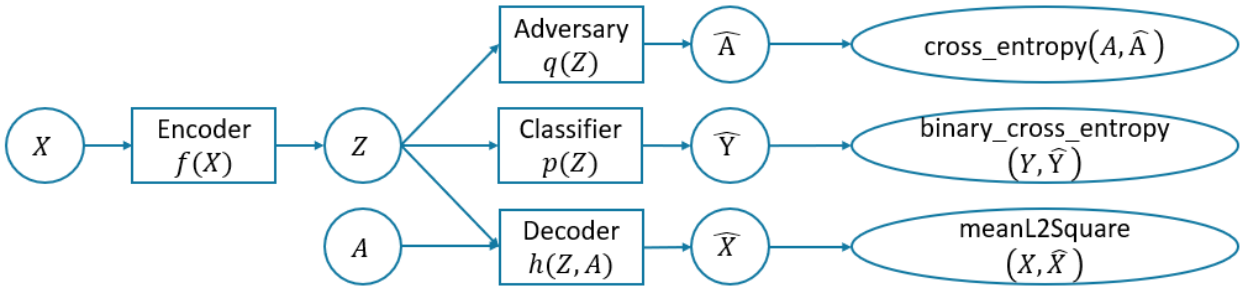

Figure 4.1. A Model for learning adversarially unbiased representations. The variables are images X, latent representations Z, label Y, sensitive attribute A. The encoder f maps X (and possibly A) to Z. The decoder h reconstructs X from Z and A. The classifier p predicts Y from Z. The adversary q predicts A from Z (and possibly Y).

Dataset-membership Unlearning Unit: The first requirement on the latent representation Z was including an adversary network q: $R^m \rightarrow \{0, 1, 2\}$ that attempts to predict the dataset membership A from the representation Z (and possibly Y) but should not be able to do that.

Therefore, the responsibility of this unit is to learn the representation Z that minimizes the performance of the best possible dataset membership classifier such that the dataset membership information is unlearnt and forgotten. For this unit, we utilized cross-entropy loss (Eq. (1)) between the ground truth for our sensitive variable A and the predicted value $\hat{A}$, where N is the batch size and C is the number of classes. Herein, we observed an adversarial relationship that comes from the fact that we are trying to minimize the performance of the best possible dataset membership classifier. Adversarial learning is a popular neural network technique and is inspired by the paper (I. J. Goodfellow, Shlens, and Szegedy 2015). In that paper, they discussed a generative adversarial model that consists of two neural networks, the generator G and the discriminator D. The generator tries to fool the discriminator by generating the synthetic data, while the discriminator aims to better distinguish between real and generated data.

$$\mathcal{L}_{Adv}(\hat{A}, A) = \frac{1}{N}\sum_{n=1}^{N}\ell_n, \quad \ell_n = -\sum_{c=1}^{C} w_c log\frac{\exp(\hat{A}_{n,c})}{\sum_{i=1}^{C}\exp(\hat{A}_{n,i})}\cdot A_{n,c} \tag{1}$$

Variable of Interest Classification Unit: Another requirement on the latent representation Z was including a classifier p: $R^m \longrightarrow \{0, 1\}$ that can predict the variable of interest Y (the presence of beluga whales). Therefore, this unit enables learning a representation Z that maximizes the classification performance of variable of interest Y. In this unit, we utilized a binary cross-entropy loss (Eq. (2)) between the ground truth for our variable of interest Y and the estimated value $\hat{Y}$, where N is the batch size.

$$\mathcal{L}_C(\hat{Y}, Y) = \frac{1}{N}\sum_{n=1}^{N}\ell_n, \quad \ell_n = -w_n[Y_n \cdot log\sigma(\hat{Y}_n) + (1 - Y_n)\cdot \log(1 - \sigma(\hat{Y}_n))] \tag{2}$$

Autoencoder Unit: An autoencoder is a type of neural network which first encodes the image into a latent representation with a lower dimension and then decodes the latent representation back to an image. In our framework, the encoder f: $R^n \longrightarrow R^m$ maps the image input

X (and possibly A) to representation Z and the decoder h: $R^m \times \{0, 1\} \longrightarrow R^n$ reconstructs X from Z and A. Based on the previous two constraints on the representation Z, there would be a possibility that all the information might be lost if every data point maps to the constant number. This way, the adversarial unit's objective would be satisfied because nothing can be inferred from a constant number, either the dataset membership or the variable of interest. In order to prevent this outcome, this unit is added as a regularization constraint on the representation Z in the form of reconstruction error, preventing the representation from losing too much information. The reconstruction loss was measured by the mean squared error (squared L2 norm) between each element of the input X and the target $\hat{X}$, where N is the batch size (Eq. (3)).

$$\mathcal{L}_{Dec\_h}(\hat{X}, X) = \frac{1}{N}\sum_{n=1}^{N} \ell_n, \quad \ell_n = (\hat{X}_n - X_n)^2 \tag{3}$$

Our framework benefited from a combined loss (Eq. (4)) of reconstruction loss, classification loss, and adversarial loss to unlearn the dataset-membership information. The encoder, decoder, and classifier jointly seek to minimize the classification loss $\mathcal{L}_C\left(p\big(f(X,A)\big), Y\right)$ and the reconstruction error $\mathcal{L}_{Dec\_h}\big(h\big(f(X,A)\big), X\big)$ and also minimize the objective of the adversary's q. The adversary's objective is to maximize $L_{Adv}\big(q\big(f(X,\ A)\big),\ A\big)$. For having the desired balance between the reconstruction, classification and adversarial losses, we specified the hyperparameters α, β, ϒ, respectively, for the combined loss.

$$\begin{aligned}\mathcal{L}(f,h,p,q) = \ & +\alpha\mathcal{L}_{Dec\_h}\ \big(h\big(f(X,A)\big), X\big) \\ & +\beta\,\mathcal{L}_C \quad \big(p\big(f(X,A)\big), Y\big) \\ & +\Upsilon\,\mathcal{L}_{Adv} \quad \big(q\big(f(X,A)\big), A\big)\end{aligned} \tag{4}$$

The evaluations were based on the LODO experiment, the same as the previous chapter but only for a different algorithm (learning adversarially unbiased representation). The training and test sets all remain consistent for valid comparison. For the learning process, we alternated gradient descent and ascent steps to optimize the parameters based on the combined loss (Eq. (4)). First, the encoder, decoder and classifier (f, h, p) take a gradient step to minimize $\mathcal{L}$ while the adversary q is fixed (freezing the adversary model). Then, the adversary takes a step to maximize $\mathcal{L}$ with fixed (f, g, p). That means we froze the whole network except the adversary part of the network. For freezing, we set the *requires-grad* flags to *False* for the specific part of the network.

The classifier p and the adversary q were a feed-forward MLP with three hidden layers trained with a learning rate of 0.01, while a learning rate of 0.0001 was used for the autoencoder. The whole network was trained using an Adam optimizer (Kingma and Ba 2015) with 60 epochs and a batch size of 256. Based on our observation in the process of fine-tuning the model's hyperparameters, the Adam optimizer worked better than stochastic gradient descent (SGD). If the data within each batch is not balanced, the cross-entropy adversarial objective potentially leads to the bias towards predicting the majority class correctly. We addressed this issue by ensuring that each batch has an equal number of images per dataset with an equal number of beluga vs non-beluga images. Regarding the criterions, we applied MSELoss in PyTorch for the decoder. For the adversary, we utilized CrossEntropyLoss in PyTorch, combining the Softmax (normalizing the scores for the given classes for multi-class classification) with Cross-Entropy-Loss to calculate the loss of model (LogSoftmax + NLLLoss (negative log-likelihood loss)). For the classifier, we benefited BCEWithLogitsLoss from PyTorch packages that combines a Sigmoid layer and the BSCLoss (binary cross-entropy) for one single class.

4.3 Results and Discussion

We compare the results of learning adversarially unbiased representation with the traditional baseline established in the previous chapter (Tables 4.1, 4.2, and 4.3).

Table 4.1. Comparing the results of learning adversarial representation against our baseline model (trained on D_2 and D_3 and tested on D_1)

Tested on good water quality photos list (D_1)				
Models	Test Set	Class Accuracies	Overall Accuracy	AUC
Traditional Baseline	D_{1b}	[93.49% 78.80%]	86.26%	0.9390
AdvRep	D_{1b}	[96.19% 74.40%]	85.46%	0.9220
Traditional Baseline	D_{1a}	[94.38% 76.95%]	85.53%	0.9314
AdvRep	D_{1a}	[95.87% 73.14%]	84.33%	0.9173

Table 4.2. Comparing the results of learning adversarial representation against our baseline model (trained on D_1 and D_3 and tested on D_2)

Tested on moderately bad water quality photos list (D_2)				
Models	Test Set	Class Accuracies	Overall Accuracy	AUC
Traditional Baseline	D_{2b}	[73.59% 93.61%]	83.37%	0.9322
AdvRep	D_{2b}	[83.18% 87.74%]	85.41%	0.9279
Traditional Baseline	D_{2a}	[72.35% 94.39%]	83.63%	0.9339
AdvRep	D_{2a}	[82.39% 89.66%]	86.11%	0.9232

Table 4.3. Comparing the results of learning adversarial representation against our baseline model (trained on D_1 and D_2 and tested on D_3)

Tested on HIHO photos list (D_3)				
Models	Test Set	Class Accuracies	Overall Accuracy	AUC
Traditional Baseline	D_{3b}	[21.31% 97.28%]	59.62%	0.7496
AdvRep	D_{3b}	[46.23% 98.85%]	72.76%	0.9516
Traditional Baseline	D_{3a}	[20.75% 96.92%]	58.51%	0.7509
AdvRep	D_{3a}	[45.56% 99.68%]	72.39%	0.9510

The cross-dataset performance of the traditional baseline models when evaluated on the third dataset D_3 that contains images with both the horizon and water in the same frame is the

worst compared to when evaluated on the D_1 and D_2. This result is what we expect since the datasets D_1 and D_2 are relatively well-structured datasets (only the water quality is different in these two datasets), while the dataset D_3 is the most challenging dataset. When we do dataset-unlearning on well-structured datasets (D_1 and D_2) which are not noisy, the model is generalized better even for a very different dataset that has a lot of noise, such as D_3. When we bring a noisy dataset D_3 into the training set, as done in the first two cases (trained on D_1- D_3 or D_2- D_3), the inclusion of data points from D_3 makes it increasingly difficult to learn a model from noisy examples. Since D_3 is extremely noisy, even when we are attempting to perform dataset-unlearning, it seems part of the noise persists within the training data representation, and therefore there is even a little bit of drop in test data performance.

D_1 and D_2 datasets are representative of the most practical situations. These photos had a relatively controlled setup, and we might not be able to capture datasets in a very different setting like D_3 with images that are not totally under the water (since the camera was not mounted deep enough in the water) and all other kinds of possible variations. As D_1 and D_2 included photos with normal water quality and murky water but overall camera viewing angles were still standard, performing dataset-unlearning on these relatively structured datasets, resulted in dramatic improvement even for highly unstructured datasets such as D_3. Moreover, observing the same trend in both test folds increases the confidence of the reported results.

Chapter 5

Supervised Contrastive Learning

5.1 Introduction

In the previous chapter, we presented an approach that aims to explicitly handle the dataset bias problem by learning representations that unlearn dataset membership. In this chapter, we will investigate if a contrastive learning paradigm can implicitly address the issues due to dataset bias. Following a supervised contrastive approach, we learned representations by contrasting the set of all samples from the same class as positives against the negatives from the remainder of a training batch (P. Khosla et al. 2020). In fact, the label information is leveraged to bring together the cluster of images belonging to the same class while simultaneously moving apart the cluster of images from a different class. The supervised contrastive loss was set up to contrast the set of all samples from the same class as positives against the negatives from the remainder of the batch, as demonstrated in Figure 5.1.

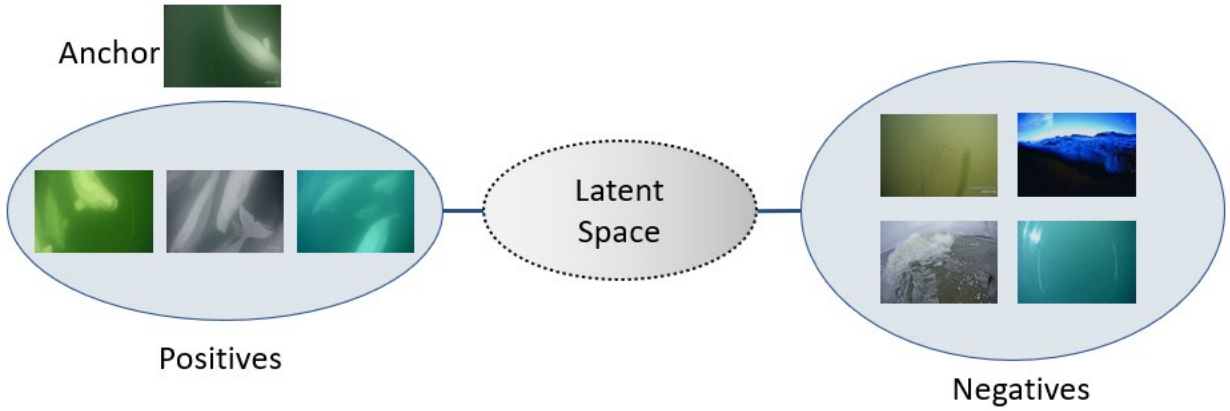

Figure 5.1. In supervised contrastive learning, a random training sample is selected (anchor) from a batch of random examples, and a representation is learned such that the samples of the same class as the anchor are brought closer. In contrast, the rest of the examples are pushed apart. This is repeated over multiple anchors.

5.2 Supervised Contrastive Methodology

We built a framework to learn a latent representation (or embeddings[1]) such that the similarity between images containing beluga is maximized, and the similarity between beluga images and non-beluga images in the latent space is minimized using a contrastive loss (Kopuklu et al. 2021). To the best of our knowledge, a supervised contrastive paradigm has not been adapted for the dataset bias problem. This framework has three main components as shown in Figure 5.2: Architecturally, it consists of two blocks, the base encoder, and a projection head; from the perspective of learning, the key component is a contrastive loss which we will describe shortly.

The base encoder is used to extract latent feature representations of the images. The actual architecture of the encoder is a choice. For our experiments, we used a straightforward VGG-16 architecture(Simonyan and Zisserman 2015) as the encoder to allow a fair comparison with other approaches. Since we did not use the encoder for classification purposes, we excluded the last layer of the VGG-16 architecture (the output layer with Softmax activation), so we were able to transform the input image X into the latent space $f_\theta(x) \in \mathbb{R}^{4096}$. Next, we used a projection network to map the output of the encoder to another space $p \in \mathbb{R}^{128}$. The projection head is a

[1] Embedding is an alternative term used in machine learning literature for latent representations (Mikolov et al. 2013)

multi-layer perceptron network with a single hidden layer of size 1024 with ReLU activation function and outputs a vector of size 128. The encoder and projection head are illustrated in Figure 5.2. Then, we applied $\ell 2$ normalization to the output of the projection network before giving it to the loss function.

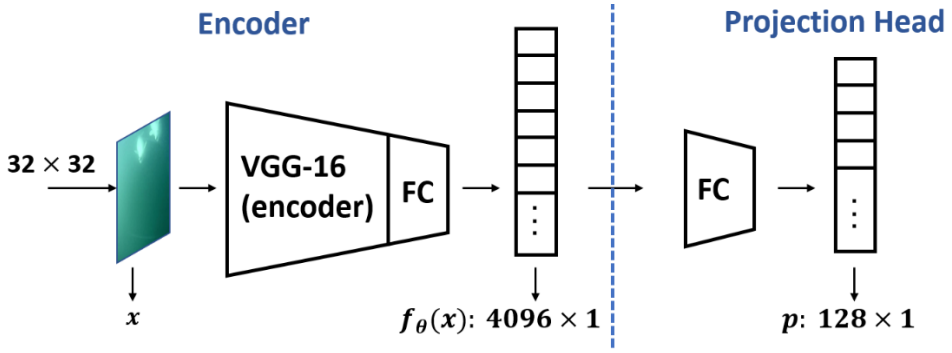

Figure 5.2. Encoder and projection head components in supervised contrastive learning.

The central idea behind the contrastive loss is to enforce the normalized embeddings from the positive class (beluga images) to get closer together as compared to the embeddings from the negative class (non-beluga images). For this reason, positive pairs in the contrastive loss were selected from the beluga images, whereas non-beluga images were used as negative samples. Let K and M be the number of beluga and non-beluga images, respectively, within a batch with index $i \in \{1, \ldots, K+M\}$. Final embedding of the i^{th} beluga and non-beluga images are denoted as p_{pi} and p_{ni}, respectively. In every batch, we have $K(K-1)$ positive and KM negative pairs in total. The contrastive loss is shown below:

$$\mathcal{L} = \frac{1}{K\,(K-1)} \sum_{i=1}^{K} \sum_{j=1}^{K} \mathbb{I}_{i \neq j}\, \mathcal{L}_{ij} \tag{1}$$

where,

$$\mathcal{L}_{ij} = -\,log \frac{exp(p_{pi}^T\, p_{pj}/\tau)}{exp\left(p_{pi}^T p_{pj}/\tau\right) + \sum_{m=1}^{M} exp(p_{pi}^T p_{nm}/\tau)} \tag{2}$$

and $\mathbb{I} \in \{0,1\}$ is an indicator function that returns 1 if $j \neq i$ and 0 otherwise, and τ, a scaler temperature parameter, is chosen between 0 and 1 that can amplify the similarity between samples.

Because p_p and p_n are $\ell 2$ normalized, the inner product of these feature vectors measures the cosine similarity between them. By optimizing Eq. (1) and Eq. (2), the encoder is trained to maximize the similarity between the feature vectors of beluga images p_{pi} and p_{pj} while minimizing the similarity between the feature vector of beluga images p_{pi} and all other feature vectors of non-beluga images p_{nm} in the same batch. The outcome of the contrastive learning process is the latent representations $f_\theta(x)$ that can be achieved by minimizing the contrastive loss (different from our usual classification loss). This representation can be later used for various downstream tasks

To perform the classification on the testing set, we typically freeze the learned encoder after learning a latent representation and then build a classifier on top of the frozen encoder. However, due to having access to the normalized representation of the images, we can simply use the cosine similarity scores for classification that do not require further training or computations (Kopuklu et al. 2021). For classification, we used the trained model to encode every positive training sample (images with beluga) $X_i \in \{1, .., N\}$ into a set of $\ell 2$ normalized feature representations and built a reference template by averaging them. The normalized feature representation could be either the normalized output of the encoder or the projection head. If the normalized feature representation of an example, x_i, is $f_\theta(x_i)$, the similarity score was calculated as follows:

$$sim_i = ref_p^T \frac{f_\theta(x_i)}{\|f_\theta(x_i)\|_2} \tag{3}$$

where,

$$ref_p = \frac{1}{N} \sum_{j=1}^{K} \frac{f_\theta(x_j)}{\|f_\theta(x_j)\|_2} \tag{4}$$

In Eq. (4) above ref_p can be considered as a reference template of class p in the embedding space, while Eq. (3) computes the cosine similarity between the embedding of a test example and the template. Thus, to classify a test image X_i, we encode it again into a $\ell 2$ normalized 4096-dimensional vector (encoder's output) or 128-dimensional vector (projection head's output) and compute the cosine similarity between the encoded image and ref_p by Eq. (4). Lastly, any image whose similarity score was below a threshold, $sim_i < \gamma$, was classified as a non-beluga image.

As we can see, only a simple vector multiplication is performed for the evaluation of the trained model.

All the evaluations were based on the LODO experiment, the same as the previous two chapters, all with the same training and test sets for the purpose of comparison. In this experiment, the stochastic gradient descent (SGD) method was used as the optimization method with a momentum of 0.9. Our model was trained from scratch for 200 epochs and a learning rate of 0.0001 and a temperature $\tau = 0.9$. The batch size was 256, and for every batch, $1/8(batch\ size)$ were beluga images, and $7/8(batch\ size)$ were non-beluga images.

5.3 Results and Discussion

We evaluated the supervised contrastive learning using AUC and accuracy for the aim of cross-dataset generalization. As shown in Tables 5.1, 5.2, and 5.3, we observed that the supervised contrastive approach outperformed the baseline model and adversarial representation learning, demonstrating that our generalization ability improved by using a contrastive loss, even though we have not explicitly set up the model to unlearn the dataset bias information. In Table 5.3, when we trained the model on D_1 and D_2 and tested on the left-out dataset D_3, the performance improved more than in the other two experiments. This might be due to the fact that datasets D_1 and D_2 are relatively well-structured datasets, so the positive samples could be easily contrasted against the negative samples. The improvement for the second experiment (Table 5.2) is more than the first experiment (Table 5.1) because for the first case, the model is trained on D_2 and D_3 but not D_1 which is the most well-structured dataset. The rows corresponding to the contrastive approach are highlighted in both tables. We can conclude that the more a well-structured dataset we have for training, the better we can contrast the positive images versus the negative images. Having well-structured and clean datasets for training set as well as using contrastive representation learning results in a classifier with better cross-dataset generalization. Therefore, only having well-structured training sets is not sufficient based on our experiments, as we observed significant degradation in performance of cross-dataset evaluation when we used our baseline architecture. We have the same trend on both folds for all three experiments. We plotted the histogram of the frequency distribution of the scores for both folds of three testing sets (Figures 5.3, 5.4, and 5.5).

The histograms on both folds show that the score values from positive and negative classes are moved apart while the values within each class are brought together. The distribution of the scores is according to what we expect from supervised contrastive learning.

Table 5.1. Comparing the results of supervised contrastive learning against learning adversarial representation and our baseline model (trained on D_2 and D_3 and tested on D_1)

Tested on good water quality photos list (D_1)				
Models	Test Set	Class Accuracies	Overall Accuracy	AUC
Baseline	D_{1b}	[93.49% 78.80%]	86.26%	0.9390
AdvRep	D_{1b}	[96.19% 74.40%]	85.46%	0.9220
SupCon	D_{1b}	[96.65% 82.66%]	89.76%	0.9669
Baseline	D_{1a}	[94.38% 76.95%]	85.53%	0.9314
AdvRep	D_{1a}	[95.87% 73.14%]	84.33%	0.9173
SupCon	D_{1a}	[96.88% 83.84%]	90.26%	0.9627

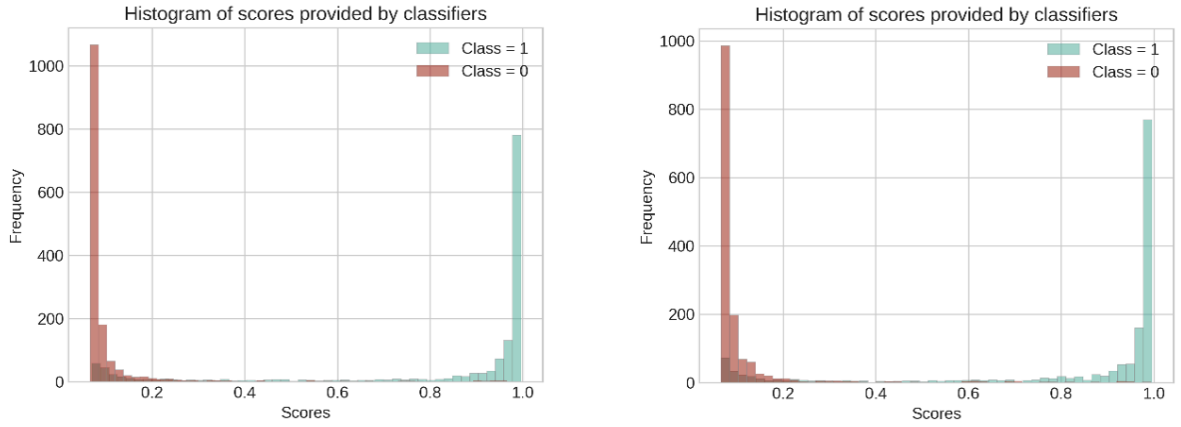

Figure 5.3. Histogram plot for the frequency distribution of scores provided by the classifier (Left: when tested on the 2nd fold of good water quality photos list (D_{1b}). Right: when tested on the 1st fold of good water quality photos list (D_{1a}).)

Table 5.2. Comparing the results of supervised contrastive learning against learning adversarial representation and our baseline model (trained on D_1 and D_3 and tested on D_2)

Tested on moderately bad water quality photos list (D_2)				
Models	Test Set	Class Accuracies	Overall Accuracy	AUC
Baseline	D_{2b}	[73.59% %93.61]	83.37%	0.9322
AdvRep	D_{2b}	[83.18% 87.74%]	85.41%	0.9279
SupCon	D_{2b}	[89.53% 96.61%]	92.99%	0.9832
Baseline	D_{2a}	[72.35% %94.39]	83.63%	0.9339
AdvRep	D_{2a}	[82.39% 89.66%]	86.11%	0.9232
SupCon	D_{2a}	[88.26% 98.00%]	93.24%	0.9823

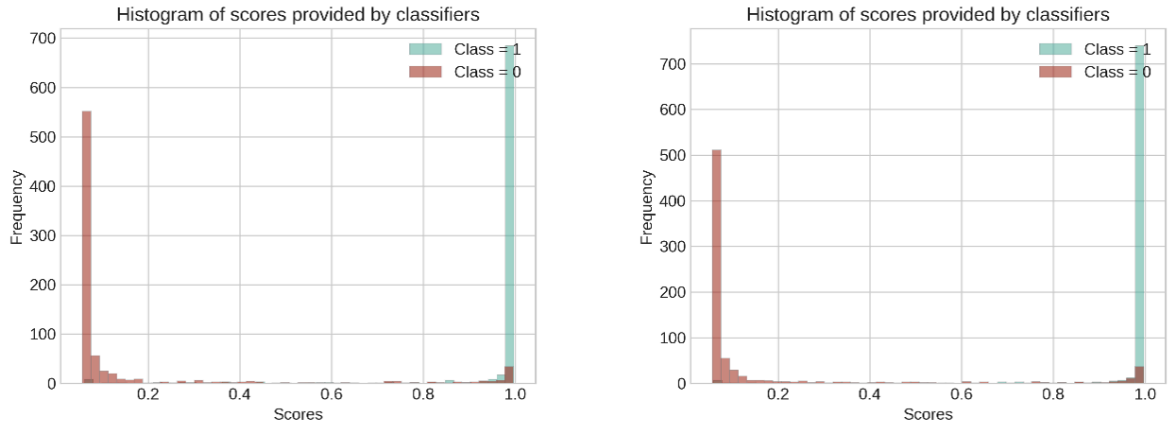

Figure 5.4. Histogram plot for the frequency distribution of scores provided by the classifier (Left: when tested on the 2nd fold of moderately bad water quality photos list (D_{2b}). Right: when tested on the 1st fold of moderately bad water quality photos list (D_{2a}).)

Table 5.3. Comparing the results of supervised contrastive learning against learning adversarial representation and our baseline model (trained on D_1 and D_2 and tested on D_3)

Tested on HIHO photos list (D_3)				
Models	Test Set	Class Accuracies	Overall Accuracy	AUC
Baseline	D_{3b}	[21.31% 97.28%]	59.62%	0.7496
AdvRep	D_{3b}	[46.23% 98.85%]	72.76%	0.9516
SupCon	D_{3b}	[95.22% 96.76%]	96.00%	0.9918
Baseline	D_{3a}	[21.31% 97.28%]	59.62%	0.7496
AdvRep	D_{3a}	[46.23% 98.85%]	72.76%	0.9516
SupCon	D_{3a}	[95.22% 96.76%]	96.00%	0.9918

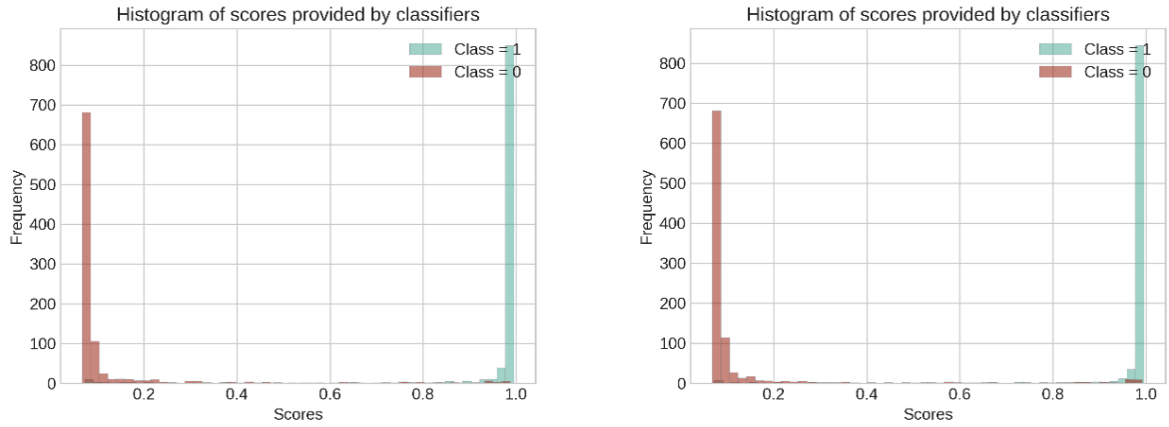

Figure 5.5. Histogram plot for the frequency distribution of scores provided by the classifier (Left: when tested on the 2nd fold of HIHO photos list (D_{3b}). Right: when tested on the 1st fold of HIHO photos list (D_{3a}).)

Chapter 6

Conclusion and Future Works

We demonstrated the challenge of building a generalizable model that we can use across different datasets through our cross-dataset evaluations. Our experiments showed that typical evaluation of the models on the same dataset's splits is unreliable, and we required more evaluation methods to make better judgements regarding a trained model's performance. Data collection for a particular problem might be from different sources; therefore, each might contain its idiosyncrasies, even if the purpose of collection is the same. These differences were shown to lead to the dataset bias issue, negatively affecting the models' generalization. We benefitted from successfully applying a contrastive loss function that showed consistently better performance over cross-dataset evaluation on other datasets as compared to the traditional approach as well as in comparison to an adversarial method that explicitly attempts to unlearn the dataset bias. We observed that the AUC values of cross-dataset evaluation in a contrastive setting reached very close to the values obtained in the within-dataset evaluation of our baseline model.

A few limitations and directions in which this work can progress in future are given below. In our approach, the availability of at least three datasets played a central role in developing cross-dataset evaluations and all the comparisons conducted in our experiments. However, having at least four datasets (i.e., data collected under four different acquisition profiles, if applicable) will help us expand the method by performing internal cross-validation during LODO for better tuning of hyperparameters. To see how four datasets can help, currently with three datasets, during LODO experiments, training is done on two datasets. With four datasets, the training could be done on

three datasets, which means we can have an inner (second level) LODO cross-validation within the training set. This will also allow us to investigate how much dataset membership information has been forgotten in a learned representation of both proposed approaches, i.e., contrastive (Chapter 5) and adversarial (Chapter 4). Building a dataset membership classification on top of the built representation can enable us to assess the extent to which the dataset membership has been unlearned by quantifying the drop in dataset membership classification as done on learned representation in comparison to raw data.

In closing, we should also note that the focus of this thesis has been exploring cross-dataset generalization for binary classification. However, we can extend our work to multi-class classification. This type of image classification can either be a single label problem of categorizing images into precisely one of more than two classes or a multi-label problem wherein each data point could concurrently contain objects from multiple classes. We can investigate cross-dataset generalization on a multi-class, multi-label problem if we have at least three datasets.

REFERENCES

Ahmadianfar, Iman, Omid Bozorg-Haddad, and Xuefeng Chu. 2020. “Gradient-Based Optimizer: A New Metaheuristic Optimization Algorithm.” *Information Sciences* 540: 131–59. https://doi.org/10.1016/j.ins.2020.06.037.

Alex Graves. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in Computational Intelligence. Springer.

Ashraf, Ahmed, Shehroz Khan, Nikhil Bhagwat, Mallar Chakravarty, and Babak Taati. 2018. “Learning to Unlearn: Building Immunity to Dataset Bias in Medical Imaging Studies.” http://arxiv.org/abs/1812.01716.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. New York : Springer.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. “A Simple Framework for Contrastive Learning of Visual Representations.” https://doi.org/https://doi.org/10.48550/arXiv.2002.05709.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. “Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation.” *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–34. https://doi.org/10.3115/v1/d14-1179.

Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” *Machine Learning* 20(3): 273--297.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. 2015. “Explaining and Harnessing Adversarial Examples.” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–11.

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2019. “Momentum Contrast for Unsupervised Visual Representation Learning.”

https://doi.org/https://doi.org/10.48550/arXiv.1911.05722.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem: 770–78. https://doi.org/10.1109/CVPR.2016.90.

Henaff, Olivier J., Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. 2020. “Data-Efficient Image Recognition with Contrastive Predictive Coding,” no. 2018. https://doi.org/https://doi.org/10.48550/arXiv.1905.09272.

Jetley, Saumya, Nicholas A. Lord, Namhoon Lee, and Philip H.S. Torr. 2018. “Learn to Pay Attention.” In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Khosla, Aditya, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. 2012. “Undoing the Damage of Dataset Bias.” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7572 LNCS (PART 1): 158–71. https://doi.org/10.1007/978-3-642-33718-5_12.

Khosla, Prannay, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. “Supervised Contrastive Learning.” *Advances in Neural Information Processing Systems* 2020-Decem (NeurIPS): 1–23.

Kim, Mingyu, Jihye Yun, Yongwon Cho, Keewon Shin, Ryoungwoo Jang, Hyun-jin Bae, and Namkug Kim. 2019. “Deep Learning in Medical Imaging.” *Neurospine* 16 (4): 657–68. https://doi.org/10.14245/ns.1938396.198.

Kingma, Diederik P., and Jimmy Lei Ba. 2015. “Adam: A Method for Stochastic Optimization.” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Kopuklu, Okan, Jiapeng Zheng, Hang Xu, and Gerhard Rigoll. 2021. “Driver Anomaly Detection: A Dataset and Contrastive Learning Approach,” 91–100. https://doi.org/10.1109/wacv48630.2021.00014.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM* 60 (6): 84–90. https://doi.org/10.1145/3065386.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. *Deep Learning*. *Nature*. Vol. 521. https://doi.org/10.1038/nature14539.

Madras, David, Elliot Creager, Toniann Pitassi, and Richards Zemel. 2018. "Learning Adversarially Fair and Transferable Representations."

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Patil, Aseem, and Milind Rane. 2021. "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition." *Smart Innovation, Systems and Technologies* 195: 21–30. https://doi.org/10.1007/978-981-15-7078-0_3.

Ribani, Ricardo, and Mauricio Marengoni. 2019. "A Survey of Transfer Learning for Convolutional Neural Networks." In *Proceedings - 32nd Conference on Graphics, Patterns and Images Tutorials, SIBGRAPI-T 2019*. https://doi.org/10.1109/SIBGRAPI-T.2019.00010.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *Springer International Publishing*, 1–8. https://doi.org/https://doi.org/10.1007/978-3-319-24574-4_28.

Saenko, Kate, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. "Adapting Visual Category Models to New Domains." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6314 LNCS (PART 4): 213–26. https://doi.org/10.1007/978-3-642-15561-1_16.

Simonyan, Karen, and Andrew Zisserman. 2015. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *Computer Vision and Pattern Recognition*. https://doi.org/https://doi.org/10.48550/arXiv.1409.1556.

Sun, Shiliang, Honglei Shi, and Yuanbin Wu. 2015. *A Survey of Multi-Source Domain Adaptation*.

Information Fusion. Vol. 24. https://doi.org/10.1016/j.inffus.2014.12.003.

Tommasi, Tatiana, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. 2017. "A Deeper Look at Dataset Bias." *Advances in Computer Vision and Pattern Recognition*, no. 9783319583464: 37–55. https://doi.org/10.1007/978-3-319-58347-1_2.

Tommasi, Tatiana, Novi Quadrianto, Barbara Caputo, and Christoph H. Lampert. 2013. "Beyond Dataset Bias: Multi-Task Unaligned Shared Knowledge Transfer." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7724 LNCS (PART 1): 1–15. https://doi.org/10.1007/978-3-642-37331-2_1.

Torralba, Antonio, and Alexei A. Efros. 2011. "Unbiased Look at Dataset Bias." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1521–28. https://doi.org/10.1109/CVPR.2011.5995347.

Wu, Zhirong, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. "Unsupervised Feature Learning via Non-Parametric Instance Discrimination." https://doi.org/https://doi.org/10.48550/arXiv.1805.01978.

Yamashita, Ayumu, Noriaki Yahata, Takashi Itahashi, Giuseppe Lisi, Takashi Yamada, Naho Ichikawa, Masahiro Takamura, et al. 2019. *Harmonization of Resting-State Functional MRI Data across Multiple Imaging Sites via the Separation of Site Differences into Sampling Bias and Measurement Bias*. *PLoS Biology*. Vol. 17. https://doi.org/10.1371/journal.pbio.3000042.

Zemel, Richard, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. 2013. "Learning Fair Representations." *30th International Conference on Machine Learning, ICML 2013* 28 (PART 2): 1362–70.

Zooniverse.org. n.d. "No Title." www.zooniverse.org.