TEC | Tecnológico de Costa Rica

Maestría en Computación, énfasis en Ciencias de la Computación

Escuela de Ingeniería en Computación

Guided Data Augmentation by Transfer Function (GUIDATFUN)

Thesis director:

Dr. Saúl Calderón Ramírez

Author:

Barnum F. Castillo Barquero

San José, CRC

August 29, 2024

Escuela de Ingeniería en Computación
Unidad de Posgrado

TEC | Tecnológico de Costa Rica

ACTA DE APROBACION DE TESIS

Guided Data Augmentation by Transfer Function (GUIDATFUN)

Por: CASTILLO BARQUERO BARNUM FRANCO

TRIBUNAL EXAMINADOR

SAUL CALDERON RAMIREZ (FIRMA)
Firmado digitalmente por SAUL CALDERON RAMIREZ (FIRMA)
Fecha: 2024.09.12 14:35:06 -06'00'

Dr. Saúl Calderón Ramírez
Profesor Asesor

MARTIN SOLIS SALAZAR (FIRMA)
Digitally signed by MARTIN SOLIS SALAZAR (FIRMA)
Date: 2024.09.19 03:18:00 -07'00'

Dr. Martín Solís Salazar
Profesor Lector

MANUEL ZUMBADO CORRALES (FIRMA)
PERSONA FISICA, CPF-01-1525-0041.
Fecha declarada: 12/09/2024 06:37:25 PM
Esta es una representación gráfica únicamente, verifique la validez de la firma.

MSc. Manuel Zumbado Corrales
Lector Externo

LILIANA SANCHO CHAVARRIA (FIRMA)
PERSONA FISICA, CPF-03-0257-0983.
Fecha declarada: 19/09/2024 08:32:52 PM

Dra.-Ing. Lilliana Sancho Chavarría
Presidente, Tribunal Evaluador Tesis
Programa Maestría en Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

29 de agosto, 2024

Abstract

Deep Learning models are used in a wide variety of contexts, one of which is the classification of medical images for the diagnosis or detection of deceases. For the models to perform adequately great amounts of data to train them are needed, nonetheless the lack of labeled data in the medical field is noticeable due to the scarcity of medical professionals. To solve this other approaches lean on transfer learning to gather data from different sources but often the distribution between the clusters of data is too different causing accuracy issues for the models. To solve the distribution mismatch this study proposes a scoring base data augmentation policy called GUIDATFUN that measures the relatedness between the source and the target datasets and then a transfer function assigns an augmentation probability to the source images. The approach was tested with four different transfer functions in the context of chest X-ray images binary classification, the results showed that a supervised deep learning model trained with the data generated employing the GUIDATFUN method measured with statistical significance with a higher accuracy in comparison to trained with regular data in the context of domain adaptation for medical images.

Keywords: deep learning, domain adaptation, data augmentation, medical imaging.

Contents

1. Introduction

1.1 Background

Nowadays the interaction between humans and computers is present in all aspects of life, generating a vast amount of information to analyze and make decisions from. Machine Learning (ML) was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks, especially if computers could learn from these data. Deep Learning(DL) is a subset of machine learning, that differs from ML by using a complex structure of algorithms modeled based on the human brain enabling the processing of unstructured data.

Over the last decade, DL has contributed to the development of effective computer-aided diagnosis tools and more recently it has been applied to the detection and classification of diseases by the analysis of medical images of patients suffering from conditions such as cancer [1] and COVID-19 [2].

With DL comes a frequent problem, the lack of labeled data [3]. DL architectures depend on using expensive labeled datasets to train models with millions of parameters to estimate [4]. This problem is not unrelated to the medical field, on the contrary, it is exacerbated because these labels must be generated by highly qualified personnel such as radiologists, pathologists, etc. A couple of approaches to solve this problem are Domain Adaptation and Data Augmentation.

Domain Adaptation is a subset of Transfer Learning that incurs into a simple idea of obtaining more data from another domain, thus the data used to train the DL model are a combination of a dataset called source and a second one called target [5].

The problem with this approach is the differences between the source and target, the data from the target could be a small set of observations from a specific hospital or clinic that cannot be combined with the source dataset due to

the dissimilarity between patient features and imaging protocols [6], and accordingly to [7] these variations should not be ignored because they could hinder the performance of the model.

Other techniques [2] outside of transfer learning have these dissimilarities between labeled and unlabelled datasets, to detect them they use the Mahalanobis distance that measures if the features of an image follow the distribution of another group of images [8], if the differences are too wide they are called out-of-distribution (OOD).

Another quite intuitive approach is to generate more data, a technique that has been utilized for this purpose is data augmentation (DA), which is used to increase the training dataset, as well as, make the DL model more robust to different types of noise. [9]

This work presents an algorithm that qualifies the images from the source dataset based on how closely they resemble the distribution of the target dataset and by using this OOD score the images are then highly or weakly augmented to generate more data, further referred to in this document as GUIded Data Augmentation by Transfer FUNction (GUIDATFUN).

1.2 Problem Definition

The absence of data can hinder DL models [10] but more precisely can cause overfitting. Overfitting happens when the model classifies the training data way too well by memorizing the data patterns, the noise, and random fluctuations, but when confronted with unseen data outside of training the model fails to generalize resulting in poor performance as seen in [11].

Methods to generate more data such as Pseudo-label, data augmentation and domain adaptation rely both on the cluster assumption (data points similar to each other tend to form clusters) and the smoothness assumption (data points close to each other are likely to have the same label) [12].

The problem to be addressed through this study is the differences in distribution between the source and target in the context of domain adaptation, and how the correct use of data with such differences for data augmentation could

lead to a better generalization of a deep learning model [13].

1.3 Objectives

Main Objective:

To develop a novel method using domain adaptation and data augmentation for image classifiers, based on out-of-distribution scores in order to improve the generalization of Deep Learning Models.

Specific objectives:

1. Devise a data augmentation policy method utilizing OOD score based on the Mahalanobis distance.
2. Apply the proposed method and variations to the training data for a specific DL model.
3. Evaluate the proposed method in transfer learning using medical datasets with artificially generated dissimilarities.

2. Literature Study

2.1 Conceptual Framework

Tom M. Mitchell [14] defines a computer which learns as:

Definition 1. ***Machine Learning (ML)****: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*

Translated to image classification, a model learns from images from a dataset, performs a classification and learns from the experience. Now that the task required by the model is determined, it is necessary to define what it means by classification.

Definition 2. ***Classification model****: a model $f(x) = y$ assigns an input described by x to a category identified by numeric code $y \in K$ with K being the set of different categories [4].*

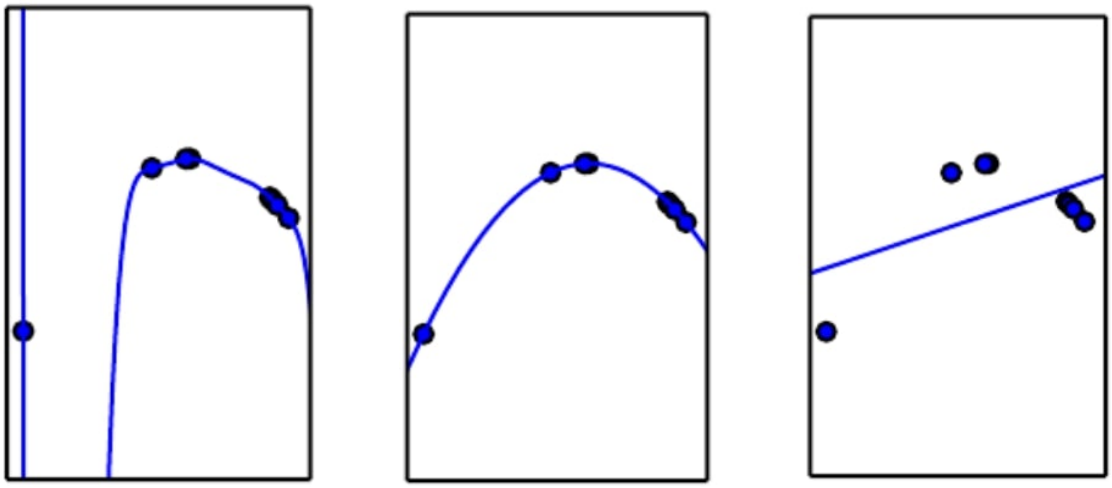
Moreover, the model is measured by its accuracy, that is the proportion of examples for which the model produces the correct output, in this context the correct classification. Thus the accuracy corresponds to a value 0 or 1, 1 if it is correctly classified and 0 if it is not.

In addition, the experience is defined as the dataset D with n observations called $\mathbf{x}_i$, thus $D = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_n\}$, for facility the datataset is defined as:

Definition 3. ***Dataset****: $D = \{X, \mathbf{t}\}$ with $X \in \mathbb{R}^{nxm}$ being a matrix with n observations with each observation having m variables and $\mathbf{t} \in \mathbb{R}^n$ the vector with n labels associated to each observation.*

Finally, to evaluate the model it is also necessary a test dataset with $D_t = \{X'\}$ of unseen examples for the model to predict [4].

Definition 4. ***Test dataset****: $D_t = \{X', \mathbf{t}\}$ with X' being a matrix with unseen examples to the model and $\mathbf{t}$ the labels associated to each observation.*



(a) Overfitting (b) Appropriate capacity (c) Underfitting

Figure 2.1: Regularization [4]

.

A central problem in machine learning is how to make a model perform well on the training data D, but also on new inputs D_t. Many strategies used in machine learning are explicitly designed to reduce the test error, possibly at the expense of increased training error. These strategies are known collectively as regularization [4], a visual representation of this balance is presented in figure 2.1.

Definition 5. ***Regularization:*** *The ability to perform well on previously unobserved inputs [4].*

Definition 6. ***Underfitting:*** *occurs when the model is not able to obtain a sufficiently low error value on the training dataset [4].*

Definition 7. ***Overfitting:*** *occurs when the gap between the training error and test error is too large [4].*

An ML model can be categorized by what kind of experience they are allowed to have during the learning process: supervised, unsupervised and semi-supervised.

Supervised learning model learns from a dataset containing observations $\mathbf{x}$, but each example is also associated with a label or target $t \in K$. Thus, the dataset defined in 3 can now be defined as 8 [4].

Definition 8. ***Labeled dataset***: $D_l = \{X, \mathbf{t}\}$ *with* $X \in \mathbb{R}^{n_l \times m}$, $\mathbf{t} \in \mathbb{R}^{n_l}$, X_l *being the matrix with all* n_l *observations and* $\mathbf{t}$ *the* n_l *labels associated to each observation.*

Unsupervised learning experiences a dataset containing many features, then learns by observing several unlabeled examples from X_u, and attempting to implicitly or explicitly learn the probability distribution $p(\mathbf{x})$, or some interesting properties of that distribution [4].

Definition 9. ***Unlabeled dataset***: $D_u = \{U\}$ *with* $U \in \mathbb{R}^{n_u \times m}$ *being a matrix with* n_u *unlabeled observations.*

A semi-supervised learning model is trained using a set of labeled observations D_l and a set of unlabelled observations D_u with the total number of observations $n = n_l + n_u$. Due to the difficulty of obtaining labeled data, the number of unlabelled observations n_u is considerably higher than the number of labeled observations in most cases.

Neural Networks (NN) are learning algorithms inspired by the biological brain, the concept started with the perceptron which can be viewed as the neuron of the network (figure 2.2). A perceptron is a linear model designed to take a set of m input values $x_0, \ldots, x_n$ and associate them with an output y. This model would learn a set of weights $w_0, ..., w_n$ to calibrate the desired activation value. To extend the linear model to represent nonlinear functions it is also necessary to apply a non-linear function ϕ called activation function [4].

Definition 10. ***Perceptron***: $\phi(f(\mathbf{x}, \mathbf{w})) = \phi(\mathbf{w}^T\mathbf{x})$ *with* $\mathbf{x}, \mathbf{w} \in \mathbb{R}^m$, $\mathbf{x}$ *the features vector and* $\mathbf{w}$ *the weights vector.*

NNs are called networks because they are typically represented by combining many different perceptrons. The model is associated with a directed acyclic graph (figure 2.2) describing how the functions are composed together. For example, $f(x) = y$ could be composed by three layer with respective functions f_1, f_2, and f_3 connected in a chain, to form $f(x) = f_3(f_2(f_1(x)))$. In this case, f_1 is called the input layer, f_2 hidden layer, and f_3 is the output layer.

Deep learning (definition 11) is a further subset of machine learning.

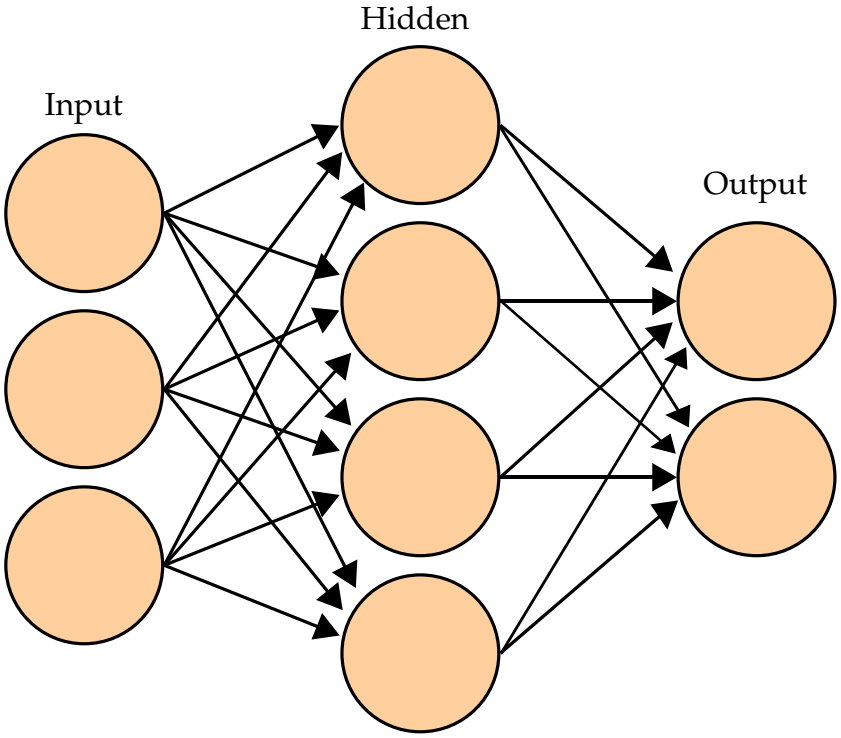


Figure 2.2: Neural Network.

Definition 11. ***Deep learning (DL)****: Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [15].*

If the hidden layer of an NN is composed of two or more layers the network is defined as a Deep Neural Network (DNN) and the overall length of the chain including the input and output layers gives the depth to the model [4].

Combining the definitions 2 and 10 with the previous concept a DNN can be defined as 12 a composite function that depends on two input parameters to give a category y.

Definition 12. ***Deep Neural Network (DNN)****: $f(\mathbf{x}, \theta) = y$ with $\mathbf{x} \in X$ and θ being the set of weights* $\mathbf{w}$ *associated to each neuron.*

Most algorithm improvements involve optimization of some sort. Optimization refers to the task of either minimizing or maximizing (denoted by *) of some loss function $\mathcal{L}$ using gradient descent. Each model defines its own $\mathcal{L}$ functions that best judge the data, and it will change depending on if the model is super-

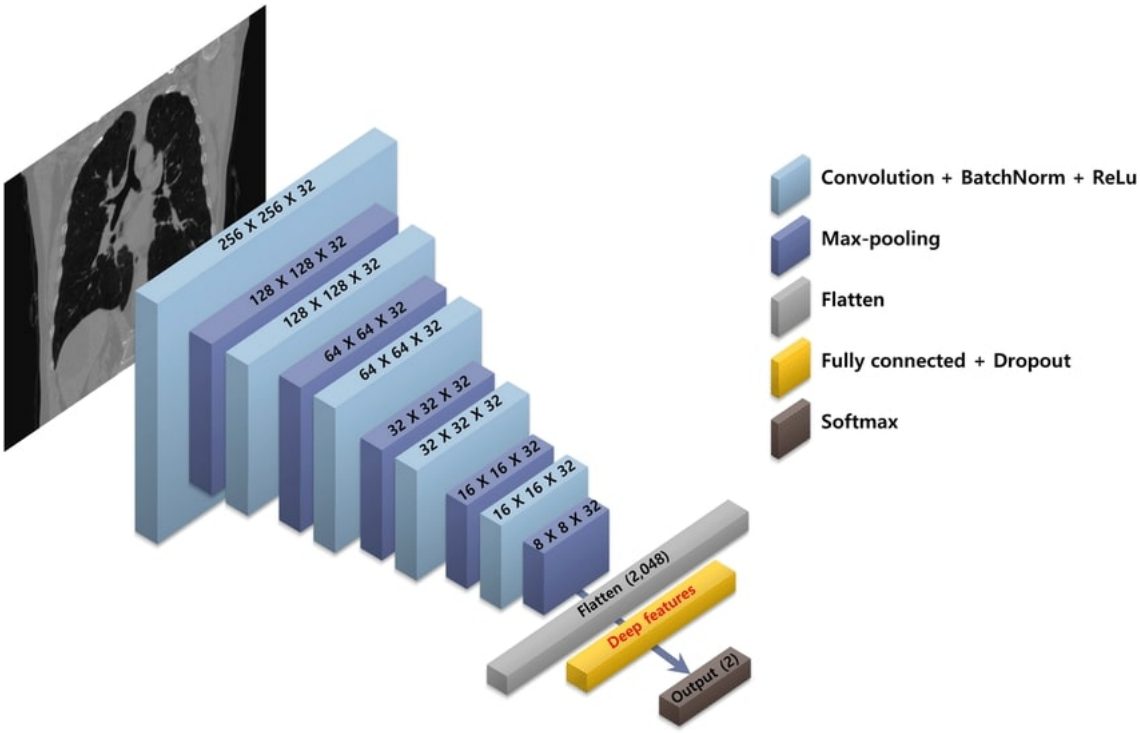


Figure 2.3: CNN architecture example [16].

vised or unsupervised [4]. For an image classifier the definition 13 explains the optimization $\mathcal{L}$ as the value of θ (definition 12) for which $f(\theta; \mathbf{x}, y)$ attains its minimum for an image $\mathbf{x}$ and a label y.

Definition 13. ***Loss function optimization:*** $\mathcal{L} = \theta^* = \arg\min f(\theta; \mathbf{x}, y)$

Convolutional Neural Networks (definition 14) are a class of deep neural networks specifically designed to process and analyze data with a grid-like structure, such as images. They are particularly effective for tasks involving visual data, including image classification, object detection, and image segmentation, as example figure 2.3.

Definition 14. ***Convolutional Neural Networks (CNNs):*** *are deep neural networks that use convolution in place of general matrix multiplication in at least one of their layers [4].*

Definition 15. ***Residual Neural Networks (ResNets):*** *Convolutional Neural Network based that adds some skip-connections or recurrent units between blocks of convolutional and pooling layers [17].*

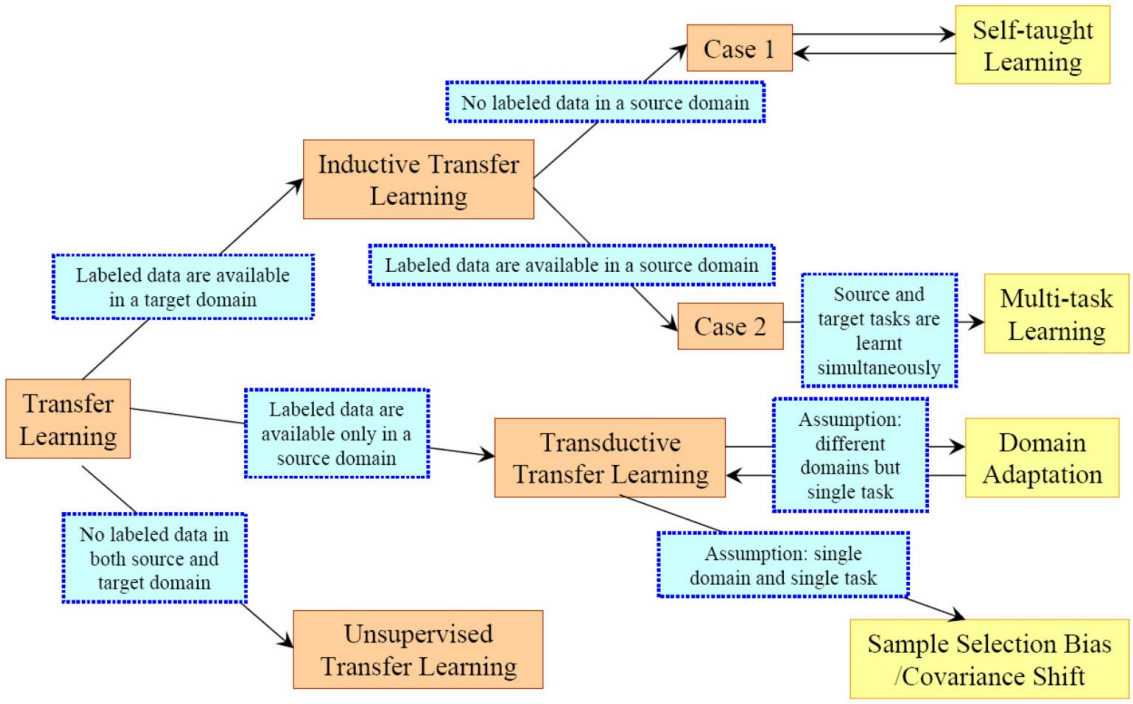


Figure 2.4: Transfer learning taxonomy [19].

In many real-world applications it is expensive to collect the needed data to feed the models, in such cases, knowledge transfer or transfer learning between task domains can be desirable.

Definition 16. ***Transfer Learning (TL):*** *the ability of a system to recognize and apply knowledge and skills learned in previous tasks to new tasks [18].*

In the context of this study, the main focus is around domain adaptation, while TL transfers the knowledge of a model over to another model with a different task, domain adaptation is used to adapt a model to what it has never seen before while maintaining the same task, figure 2.4.

Definition 17. ***Domain Adaptation (DoA):*** *is the ability to apply an algorithm trained in one or more "source domains" to a different (but related) "target domain", the tasks are the same but the domains are different [5].*

By combining definition 8 and 17 the source dataset and target dataset are defined as:

Definition 18. ***Source dataset****: $D_s = \{S\}$ with $S \in \mathbb{R}^{n_s \times m}$ being a matrix with n_s observations of the source domain.*

Definition 19. ***Target dataset****: $D_g = \{G\}$ with $G \in \mathbb{R}^{n_g \times m}$ being a matrix with n_g observations of the target domain.*

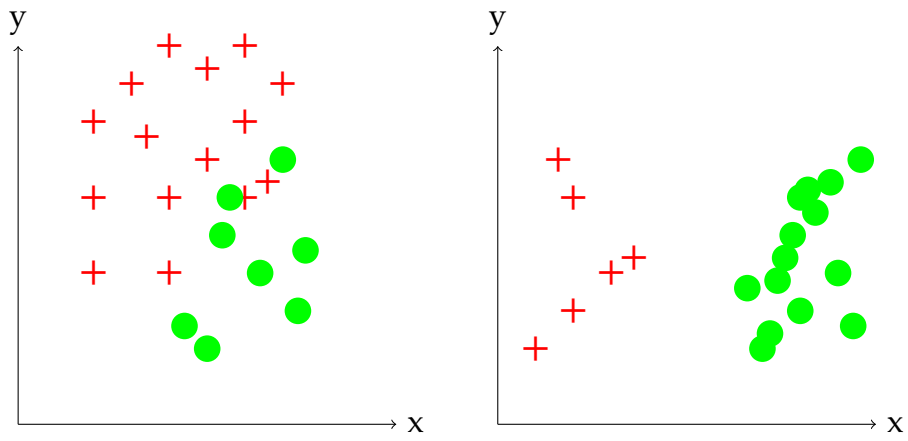
To improve a DL model further more data helps it to generalize better, but as mentioned prior in the real world the amount of data is limited [3]. One way to solve this problem is to create fake data and add it to the training set. This approach is quite useful for an image classifier that needs to take a complicated high dimensional input x and summarize it with a single category identity y. One technique for this purpose is Data Augmentation (definition 20).

Definition 20. ***Data Augmentation (DA):*** *consists of simple transformation $B(x) = x'$ being x the input and x' the modified input, for image classification this includes moving the images a few pixels in each direction, rotating the image, etc [4].*

Definition 21. ***Independent and Identically Distributed (IID) principle****: a collection of random variables is independent and identically distributed if each random variable has the same probability distribution as the others and all are mutually independent [12].*

To train DL models the IID principle (definition 21) has to be taken into consideration if not it could provoke a diminish in the accuracy [2,12]. According to [2,12] the type of violations to this principle can be categorized as:

- Feature distribution skew (covariate shift): A different distribution of the features in the input observations versus the once observed in D_s, causing a distribution mismatch, example figure 2.6.
- Label distribution skew (prior probability shift): The label imbalance between D_s and D_g, example figure 2.5.
- Same label, different features (concept drift): Two images could depict different stages in time but still have the same category, example image 2.7.
- Same features, different label (concept shift): This is associated to a shift in the labels of D_s with respect to D_g for samples with the same features. This is very related to the problem of noisy labeling.



(a) Source with more red crosses labels (b) Target with more green circles labels

Figure 2.5: Probability shift example.

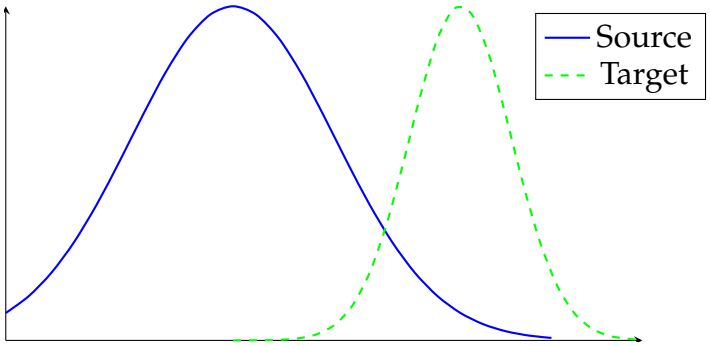


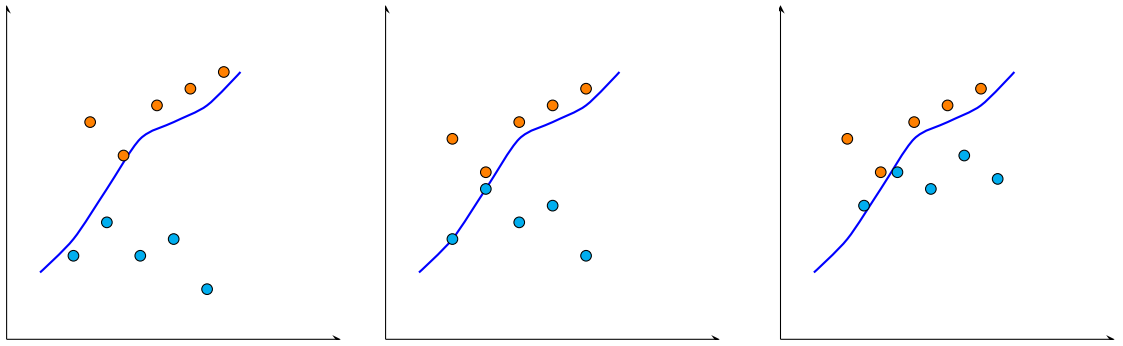
Figure 2.6: Covariate shift example, different distribution for datasets.

- Quantity skew: The dataset D_s contains observations with features that D_g simply does not have.

To counter the effect of a distribution mismatch quantitative distance (definition 22) metrics can be applied to different types of data to determine the closeness between distributions.

Definition 22. ***Distances:*** *measurements to calculate the similarity between data points represented as $M(x_1, x_2)$ [20]. For a distance M and three point $x_1, x_2, x_3 \in X$ the following following properties are satisfied:*

- *Non-negativity:* $M(x_1, x_2) \geq 0$.
- *Identity of indiscernible:* $M(x_1, x_2) = 0$ *if and only if* $x_1 = x_2$.



(a) Original Data at time t (b) Concept drift at time $t+1$ (c) Concept drift at time $t+2$

Figure 2.7: Concept drift example where the prediction on the original date is no longer correct as time increases.

- *Symmetry:* $M(x_1, x_2) = M(x_2, x_1)$.
- *Triangle inequality:* $M(x_1, x_3) \leq M(x_1, x_2) + M(x_2, x_3)$ *the distance between two points is less than or equal to the sum of the distances of a third point.*

In the context of image classification a feature extractor (definition 23) is necessary to transform the raw image data into a set of characteristics or features that can be used for analysis. These features could be edges, textures, shapes, colors, or more abstract representations learned by neural networks.

Definition 23. ***Feature Extractor:*** *is a neural network or a part of a neural network that processes raw image data and transforms it into a set of features, which are high-level representations that capture important patterns, structures, and information in the images [21]. Formally the extraction of the feature is represented as* $A(\mathbf{x}) = \mathbf{h}$ *where* $\mathbf{x}$ *is an image and* $\mathbf{h} \in \mathbb{R}^e$ *the features vector extracted from the image.*

Once the features are extracted from the images, measuring the distance (figure 2.8) between these feature sets allows to quantify the similarity or dissimilarity between images.

2.2 Medical Imaging

DL models are used in a wide range of topics, one of the most prominent of which is for medical applications. They are commonly used to make diagnoses

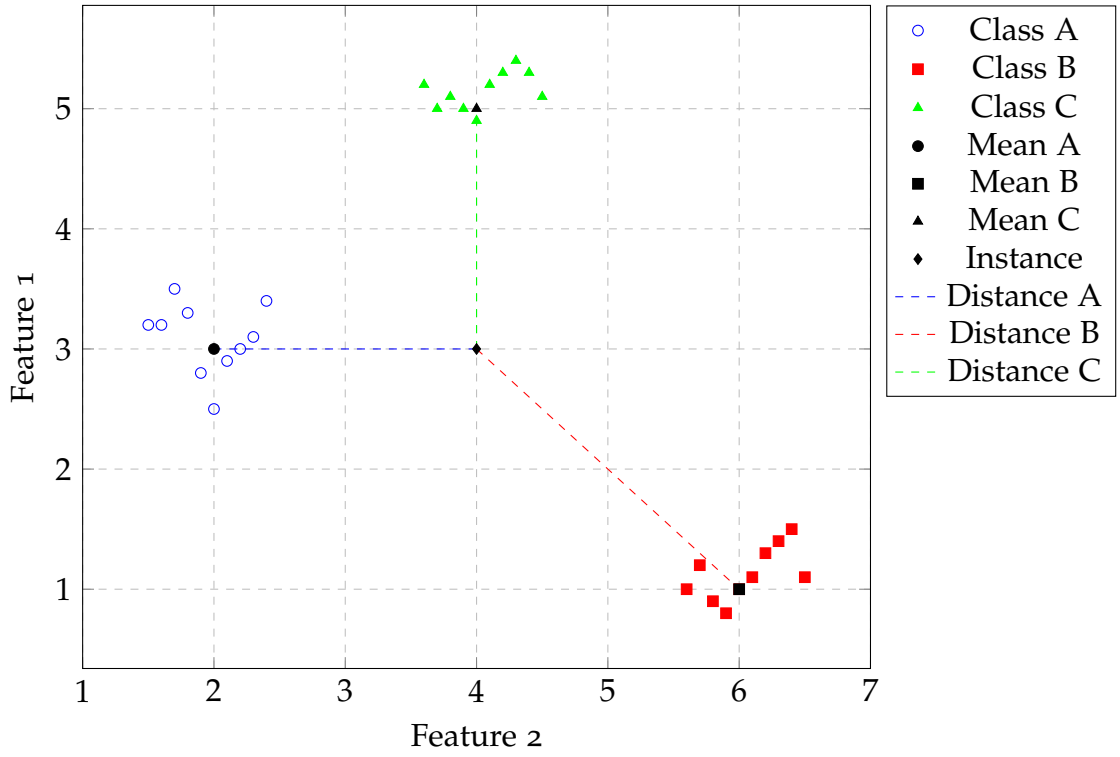


Figure 2.8: Visual representation of an example of distance measurement between three images and how the instance point could be representative of all the features as a whole.

and predictions implementing CNNs (definition 14) models trained with information obtained from patients using different tools [22] such as:

- X-Ray radiography: is a diagnostic technique that uses high-energy electromagnetic radiation to penetrate solids and create images of internal structures. Example figure 2.9 ⓐ.

- X-Ray computed tomography: is a diagnostic technology that combines X-ray equipment with a computer to produce cross-sectional images of the human body. Example figure 2.9 ⓑ.

- Magnetic resonance imaging (MRI): technology that uses magnetic and radio frequency fields to create images of body tissues and monitor body chemistry. It detects changes in proton density and magnetic spin relaxation times to visualize morphological alterations. Example figure 2.9 ⓒ.

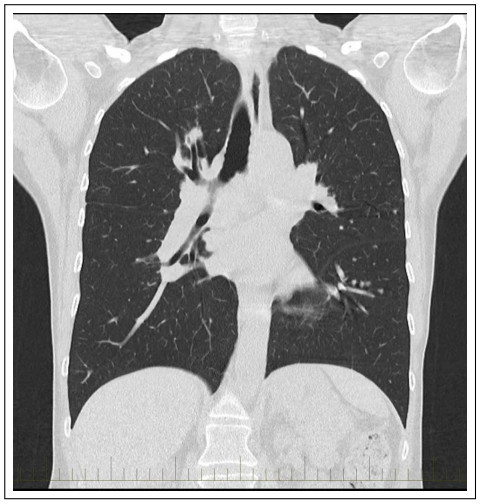
- Radionuclide imaging: technology that uses small amounts of radioactive material to create images of internal body structures. The radioactive isotopes, administered via injection or orally, are absorbed by specific organs or tissues, emitting signals detected by radiation detectors. Example figure 2.9 ⓓ.

- Ultrasonography: technology that uses high-frequency sound waves to create medical images by detecting echoes from body tissues, example figure 2.10 ⓐ.

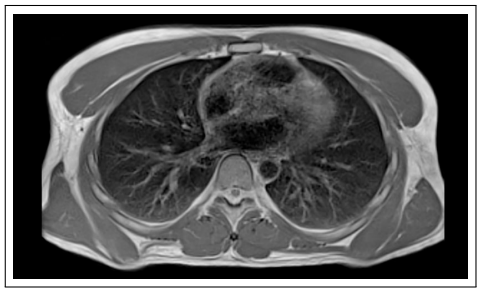
- Elastography: is a non-invasive imaging technique that assesses tissue stiffness (elasticity) to detect abnormalities. It includes ultrasound, magnetic resonance, optical, and tactile imaging. Example figure 2.10 ⓑ.

- Optical imaging: is a noninvasive technology that uses light to visualize cellular and molecular functions within the living body. It probes deep tissues by detecting light interactions with tissue components, providing contrast through exogenous agents or endogenous molecules. Example figure 2.10 ⓒ.

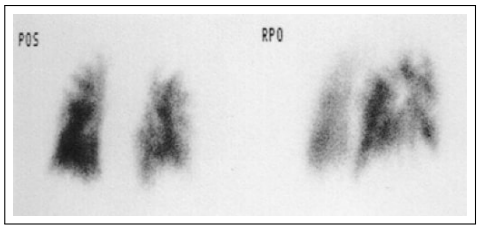
(a) X-Ray Radiography



(b) X-Ray Computed Tomography



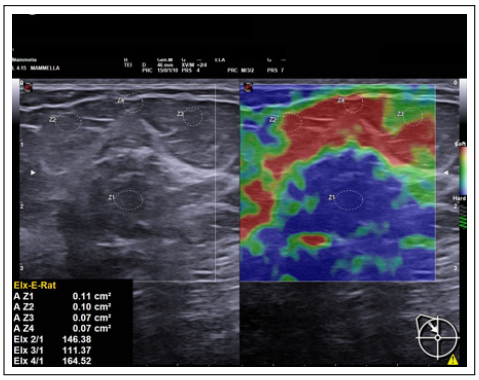
(c) Magnetic Resonance Imaging



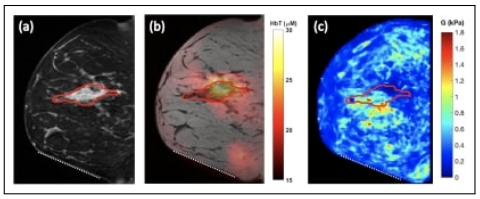
(d) Radionuclide Imaging

Figure 2.9: Medical images of lungs by different sources.

(a) Ultrasonography



(b) Elastography



(c) Optical Imaging

Figure 2.10: Medical images by different sources.

This document focuses on 2D x-ray images due to their availability in a proposal with domain adaptation (definition 17).

The persistent problem of obtaining data to train the models is present in this field due to the fact that different sources are too different, for example figure 2.9 where four images of the lungs are completely incompatible based on the technology used causing one or more of the violations defined in 21 and producing a distribution mismatch as a result. Some of the reasons for distribution mismatch in the context of medical imaging [6] are the following:

- Differences in imaging devices and protocols.
  - Variability in equipment: different hospitals and clinics may use different models and brands of imaging devices, leading to variations in image quality, resolution, and other characteristics.
  - Imaging protocols: protocols for capturing images (e.g., settings on the MRI machine, positioning of the patient) can vary between insti-

tutions, affecting the consistency of the images.

- Patient demographics.
  - Age and gender: the patient population might differ in age, gender, or other demographic factors, which can influence the appearance of medical images.
  - Health conditions: Variability in underlying health conditions and comorbidities can lead to differences in image characteristics.
- Geographic and environmental factors.
  - Geographic differences: patients from different geographic regions might have different prevalent diseases, which can affect the appearance of medical images.
  - Environmental factors: lifestyle and environmental exposures (e.g., smoking, diet, pollution) can also contribute to variations in medical images.
- Data acquisition and annotation.
  - Manual annotation variability: differences in how radiologists or technicians annotate images can introduce variability. This includes differences in labeling styles, criteria for identifying regions of interest, and inter-observer variability.
  - Quality of data acquisition: variations in how data is collected, including inconsistencies in following standardized protocols, can lead to mismatches.
- Temporal changes.
  - Technological advancements: over time, improvements in imaging technology and techniques can lead to differences in images captured at different times.
  - Changes in population health: shifts in the health status of the population, such as the prevalence of certain diseases, can also cause temporal variability in medical images.

- Institutional practices.
  - Differences in clinical practice: variations in clinical practices, such as the frequency of follow-up scans, types of contrast agents used, and pre-imaging preparations, can lead to distribution mismatches.
  - Policy and regulations: differences in local regulations and institutional policies regarding imaging practices can also contribute to variability.

2.3 State of the art

Several deep learning methods tackle low quantities of training data. This section summarizes some important techniques and studies that are relevant to this proposal.

2.3.1 Regularization Techniques

According to [23] regularization techniques can be classified into 13 broad categories:

- Data augmentation: regularization by data augmentation involves using the transformations (definition 20) to create new training examples.
- Noise injection: adds noise to the input data or the model's weights during training. This can help the model become more robust to small variations and prevent overfitting.
- Weight decay: adds a penalty to the loss function proportional to the square of the magnitude of the weights.
- Dropout: randomly drops neurons during the training process.
- Drop connect: during training each weight can be randomly set to zero, which helps in regularizing the model.
- Stochastic depth: involves randomly dropping entire layers during training.

- Early stopping: monitors the model's performance on a validation set and stops the training process when the performance stops improving.
- Label smoothing: softens the target labels by distributing a small portion of the probability mass to all classes.
- Mixup: generates new training samples by taking convex combinations of pairs of examples and their labels.
- Adversarial training: involves training the model on adversarial examples inputs that have been slightly modified to fool the model.
- Architectural regularization: implicitly regularize the training process by improving gradient flow and enabling the training of very deep networks without suffering from the vanishing gradient problem.
- Jacobian regularization: reduces the sensitivity of the model's outputs to its inputs by penalizing the model's predictions with respect to the inputs.
- Virtual adversarial training: uses virtually perturbed unlabeled examples to regularize the model.

On the preliminary review of the existing literature there were limitations found which, inside the context of domain adaptation, could cause issues when training DL models. TODO: FALTAN REFERENCIAS

Data Augmentation

Basic augmentation techniques involve transforming an image to reposition its points or manipulating its intensity values to create an augmented version. These operations are applied to individual images from the existing dataset, which are then added back to increase the dataset size.

Many studies employ small transformations to increase the training data, for example, Shyamalee and Meedeniya [24] trained a CNN model with retinal images for the diagnosis of glaucoma decease using augmented images with rotation, shearing, zooming, flipping, and shifting. Dufumier et al. [24] displayed the effectiveness of the augmentations rotation, random cropping, blurring, and

noise on MRI images to train CNN models for three classification tasks age prediction, sex classification, and schizophrenia diagnosis. [25] applied a mixup augmentation by combining images to generate new 2D CT images for pancreas segmentation.

Data augmentation transformations by themself do not account for verifying the IID principle (definition 21) thus if an out-of-distribution image is augmented and added to the training data for a classification model it will harm its accuracy [26].

A second approach for data augmentation in medical images involves Generative Adversarial Networks (GANs). GANs involve a generator G that creates images conditioned on an input image, and a discriminator D that distinguishes between real and generated images. The objective function combines a conditional adversarial loss and a traditional L1 loss.

$$\mathcal{L} = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

The generator aims to minimize this objective against an adversarial discriminator that tries to maximize it. Additionally, an L1 loss is incorporated to ensure the newly generated image is not too distinct from the real images and λ balances the two objectives.

Under this approach multiple works have been developed, Nishio et al. [27] trained a 3D GAN model to generate CT images of lung nodules, Wu et al. [28] used a GAN model to augment breast mammography images for the training of an image classifier of breast cancer.

While GAN-based methods have demonstrated impressive results, they often struggle with preserving image objects and maintaining translation consistency. This limitation reduces their effectiveness for applications such as generating large-scale training data across different domains [29].

A third approach called Variational Autoencoder (VAE) is a generative model in machine learning that combines neural networks with variational inference. It consists of an encoder, which maps input data to a latent space, and a decoder, which reconstructs the data from the latent space. The [30] utilizes a modified version called Progressive Adversarial Variational Auto-Encoder (PAVAE) to

generate realistic and diverse brain lesion images to expand the training dataset for laser interstitial thermal therapy (LITT).

VAEs learn a useful latent representation and model global structure well but have difficulty capturing small details and complex data distributions [31].

2.3.2 Domain Adaptation

For the purpose of the document the regularization techniques are focused on the domain adaptation training with medical images, according to [32] domain adaptation can be classified following this taxonomy:

- Model type: DoA methods are divided into shallow and deep based on model complexity. Shallow domain adaptation relies on human-engineered features and traditional machine learning, while deep DoA utilizes deep learning models for end-to-end feature learning and training.
- Label availability: domain adaptation methods vary by the availability of labeled data in the target domain. Supervised DoA uses a small amount of labeled data, semi-supervised DoA combines labeled and unlabeled data, and unsupervised DoA relies solely on unlabeled target data.
- Modality difference: DoA methods can address either single-modality, where source and target share the same modality, or cross-modality, where they differ (e.g., MRI to CT). Cross-modality DoA is more challenging due to the different types of data involved.
- Number of sources: methods are classified based on the number of source domains. Single-source DoA assumes one source domain, while multi-source DoA involves multiple domains, increasing complexity due to data heterogeneity.
- Adaptation step: DoA methods can be one-step, where adaptation occurs directly between source and target, or multi-step, involving intermediate domains to bridge significant distribution gaps.

A proposal to address the regularization problem is [33] a supervised and shallow domain adaptation that adopted a weight decay strategy by assigned

to the source domain different weights according to their relevance to the target dataset for MRI scans of patients with Alzheimer's disease. The learning task involves minimizing the empirical risk by solving for the following loss function $\mathcal{L}_\theta$:

$$\mathcal{L}_\theta = \arg\min \sum p(x,y)\, \mathcal{L}(x,y,\mathbf{w})$$

The $p(x,y)$ represents the joint distribution over observations x and labels y functioning as a weight regularizer added to the loss function $\mathcal{L}_\theta$ by multiplying the loss function $\mathcal{L}$.

Other technique is [34] a supervised, shallow and multiple-source DoA used to diagnose Autism spectrum disorder that instead of investing resources in having a more similar source dataset to the target it converts both the source and target into a common feature space. They transform multiple sources by selecting one as the target and using a Low Rank transformation to create Latent Representation Space (figure 2.11), then they train a model on this new data with a common distribution. This transformation is learned by minimizing the difference between source and target domains while preserving the structural information of the data and can be mathematically represented as:

$$\min_{A,B} \|SA - GB\|_F^2 + \lambda(\|A\|_* + \|B\|_*)$$

Where A and B are the transformation matrices applied to source S and target G, $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|_*$ denotes the nuclear norm.

Although DoA has found multiple methods to resolve regularization between source and target there are few studies that focus on augment source images in the medical field. Basic data augmentation methods require validations to comply with the IID principle, and advanced data augmentation methods involve having sufficient data to train models that are not easy to obtain and the sources differ from each other. Outside the medical field, there have been studies that have enough information to combine both areas of DoA and data augmentation by deep learning.

Huang et al. [29] uses a modified GAN model called AugGAN trained with traffic images. By training a GAN on the source domain data, AugGAN gener-

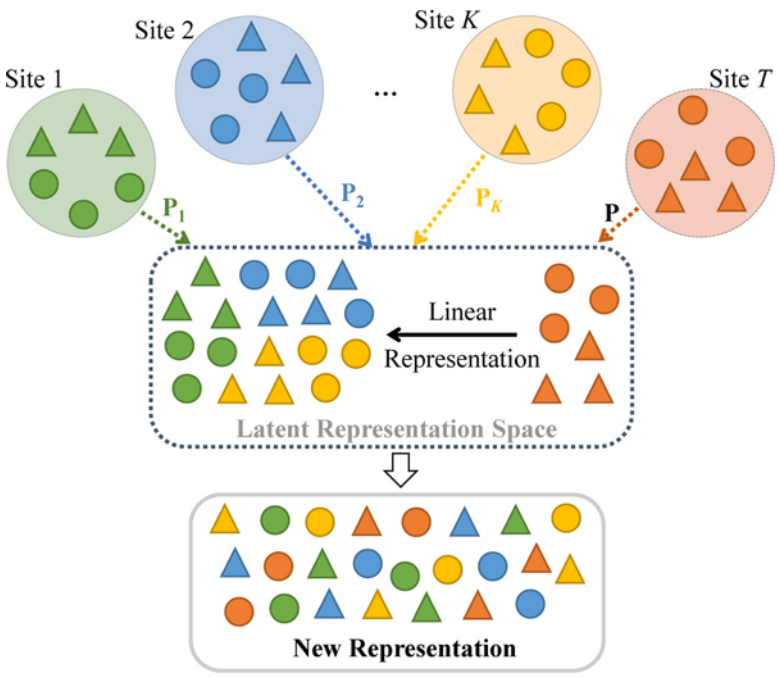


Figure 2.11: Distribution transformations applied to source and target with method [34]

ates new data that has the same label as the source but the style and characteristics of the target domain. The framework uses a cycle-consistent adversarial network (CycleGAN) to ensure that the synthetic images remain true to their original labels while adopting the style of the target domain. This is achieved through a cycle-consistency loss, which ensures that an image translated to the target domain and then back to the source domain remains unchanged. The synthetic data generated by AugGAN is used to augment the training dataset. This augmented dataset helps the model to learn features that are more generalizable across both source and target domains.

3. Scientific Proposal

To solve the problem (section 1.2) other approaches outside of transfer learning were explored, one useful technique is out-of-distribution data filtering used in semi-supervised deep learning [2], it assesses the distribution mismatch between labeled and unlabelled datasets using the Mahalanobis distance (definition 22) and filters the outliers that might harm the DL model accuracy before training.

A new technique to generate more relevant data by taking into consideration an OOD scores based on a distance would ensure that the data from the source dataset used for data augmentation would be drawn from the same probability distribution as the target dataset. With this consideration, any deep learning model for the classification of medical images or any other model with a heavy constraint in training data for that matter could have improved accuracy.

3.1 Research questions

1. Based upon the OOD score can a data augmentation policy improve the robustness of a supervised DL model when facing a distribution mismatch between the source and target dataset?

2. What would be an effective transfer function to know the number of data augmentations necessary for an image-based OOD score?

3. Is there an improvement to a supervised DL model by data augmenting images with a high OOD score?

4. Is there an improvement to a supervised DL model by data augmenting images with low OOD scores?

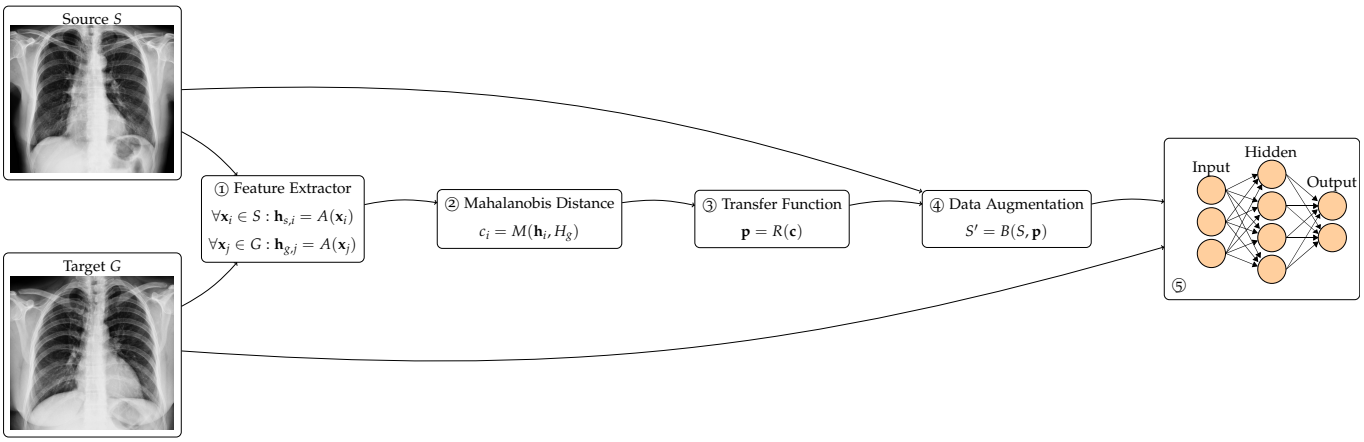


Figure 3.1: A summary of the workflow presented in this document. ① The feature extractor extracts the features for each images in S and G individually represented by $\mathbf{h}_{s,i}$ and $\mathbf{h}_{g,j}$ respectively. From the set formed by the features extracted $\mathbf{h}_{g,j}$ its probability distribution is defined as H_g. ② The Mahalanobis distance calculates the score c_i for each feature $\mathbf{h}_{s,i}$ by comparing it versus all the distribution of H_g. ③ All the scores $\mathbf{c}$ are passed to the transfer function that calculates the probability $\mathbf{p}$ for all the images of S. ④ The Data Augmentation augments the images of S following the probabilities $\mathbf{p}$. ⑤ The new data S' and target are used to train a supervised DL model.

3.2 Proposed method

In terms of regularization this is a data augmentation technique (section 2.3.1) aided with discernment for quality data from OOD data filtering (section 3) in a domain adaptation setting with the characteristics of deep, supervised, single-modality, single source and single step (section 2.3.2).

The proposed method is the GUIded Data Augmentation by Transfer FUNction (GUIDATFUN) policy to improve the robustness of a supervised DL model by selecting and generating the appropriate data. To accomplish this GUIDATFUN scores the source on how closely resembles the distribution of the target dataset, the score is then passed to a transfer function that returns the augmentation probability of the image. The proposal is divided into four sequential steps as exposed in figure 3.1: feature extractor, OOD score based on the Mahalanobis distance, the transfer function, and finally data augmentation. The result of each step is an input for the next.

3.2.1 Feature Extractor

The feature extractor (definition 23) selected is AlexNet a CNN (definition 14) architecture developed by Krizhevsky [35] in 2012 for image classification tasks. The architecture consists of a series of convolutional and max-pooling layers, culminating in three fully connected layers. It features five convolutional layers, making it relatively simple compared to later models. The network employs Rectified Linear Units (ReLUs) as activation functions and dropout regularization is utilized in the fully connected layers to mitigate overfitting (definition 7). AlexNet comprises 60 million parameters and 650000 neurons. The model is presented in figure 3.2.

AlexNet has a low complexity and computational cost as described by [21] and is pretrained with Imagenet [36] this is important so GUIDATFUN does not add a significant computational time, hence the reason for its selection.

By adding to the definition 23 the extraction source S (definition 18) and target (definition 19) images are defined as:

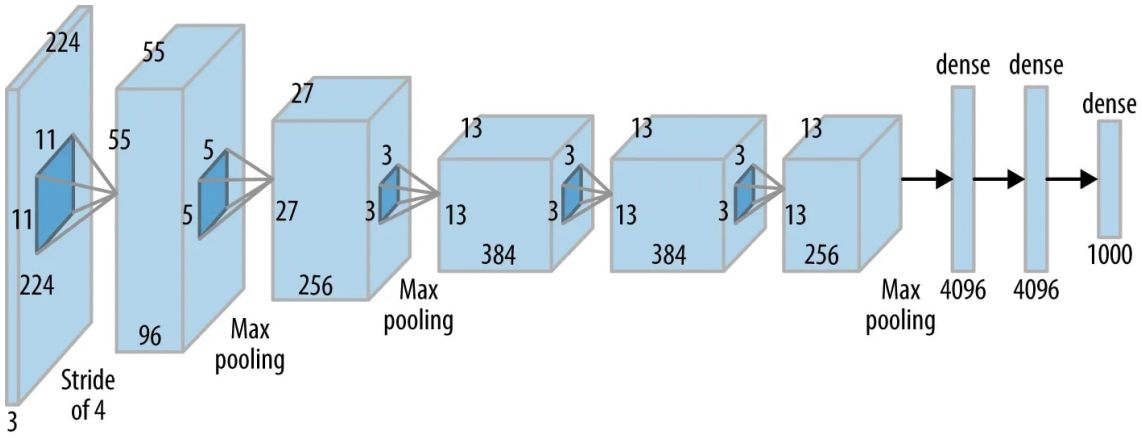


Figure 3.2: AlexNet model architecture [35].

$$\forall \mathbf{x}_i \in S : \mathbf{h}_{s,i} = A(\mathbf{x}_i) \tag{3.1}$$

$$\forall \mathbf{x}_j \in G : \mathbf{h}_{g,j} = A(\mathbf{x}_j) \tag{3.2}$$

From the set formed by the features extracted $\mathbf{h}_{g,j}$ its probability distribution is defined as H_g.

3.2.2 Mahalanobis Distance

The Mahalanobis distance (definition 22) measures the separation between two data points within the space defined by pertinent features [8].

Mahalanobis distance was chosen because it accounts for unequal variances and correlations among features, it effectively evaluates distances by assigning different weights to the features of data points making it robust against outliers. This consideration of correlations between features gives it and advantage over other metrics such as Euclidean distance, Manhattan distance (L1 distance) and Cosine Similarity [8]. In addition Mahalanobis distance is scale-invariant, meaning it is not affected by the scale of the features [8].

Given two data points $\mathbf{x}_1, \mathbf{x}_2 \in R^n$ their Mahalanobis distance can be calculated as follows:

$$\sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{\Sigma}^{-1} (\mathbf{x}_1 - \mathbf{x}_2)} = M(\mathbf{x}_1, \mathbf{x}_2) \tag{3.3}$$

For the calculation of the score $\mathbf{x}_1$ is replaced by the features extracted $\mathbf{h}_{s,i}$ (equation 3.1) and $\mathbf{x}_2$ is replaced by the probability distribution H_g (equation 3.2) so that each image in the source dataset is compared against all the distribution of G.

$$\sqrt{(\mathbf{h}_{s,i} - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}^{-1} (\mathbf{h}_{s,i} - \boldsymbol{\mu}_g)} = M(\mathbf{h}_{s,i}, H_g) \tag{3.4}$$

With $\boldsymbol{\mu}_g \in \mathbb{R}^e$ is the vectors of the means from the n_g extracted features $\mathbf{h}_{g,j}$, Σ is the covariance matrix calculated out of H_g and T is the transpose operation.

$$c_i = M(\mathbf{h}_{s,i}, H_g) \tag{3.5}$$

With $c_i \in \mathbb{R}^+$ (by non-negativity definition 22) is the individual score associated to each features extracted $\mathbf{h}_{s,i}$. To obtain all the scores the comparison has to be made as described by:

$$\forall \mathbf{h}_{s,i} \in H_s^{n_s \times e} : c_i = M(\mathbf{h}_{s,i}, H_g), \text{ where } c_i \in \mathbf{c} \tag{3.6}$$

With $\mathbf{c} \in \mathbb{R}^{n_s}$ is the vector with the out-of-distribution scores attributed to each element in S. When Mahalanobis distance score c_i is closer to zero the distribution of both elements is very similar, on the contrary, the farther the value from zero the more different the distributions are.

3.2.3 Transfer Function

The transfer function is defined as $\mathbf{p} = R(\mathbf{c})$ where the input $\mathbf{c}$ is the vector with the scores for each image in source S obtained from M in the step before and $\mathbf{p} \in \mathbb{R}^{n_s}$ is the vector with the augmentation probability of each image in the source dataset.

This proposal focused on two transfer function implementations 1) a percentage-wise step function subdivided into two named *PercentageWisePositive* and *PercentageWiseNegative* and 2) a decreasing linear function represented as *DecreasingLinear*.

The *PercentageWisePositive* function (charted in figure 3.3) ranks the n_s source images from best to worst and it attributes a 100 augmentation probability to

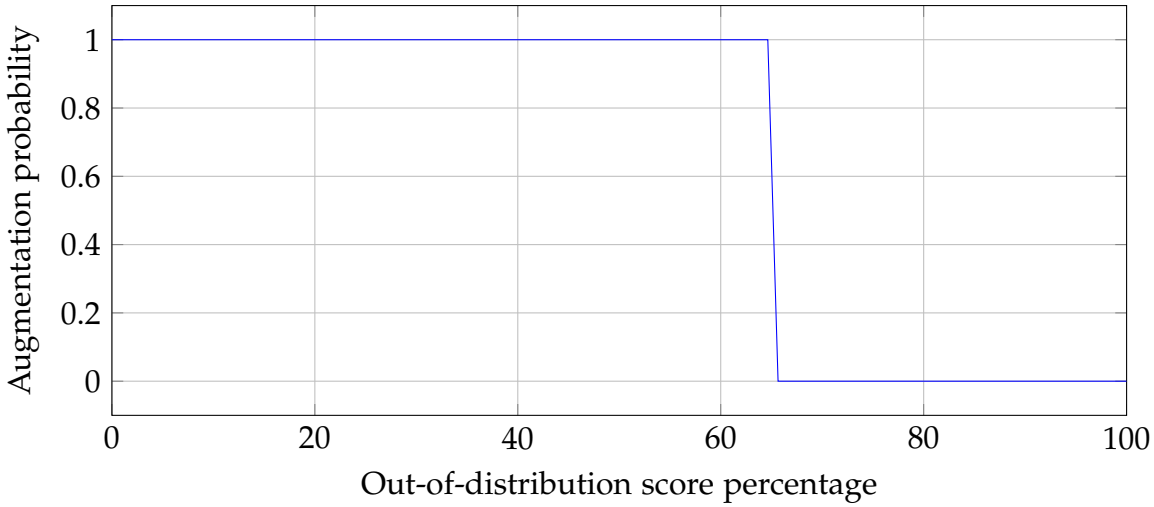


Figure 3.3: *PercentageWisePositive* transfer function.

the top 65% of images, on the contrary, to the other 35% of the worst images it attributes a 0 augmentation probability. *PercentageWiseNegative* (charted in figure 3.4) is the inverse of *PercentageWisePositive* in which the 35% of images with the worst scores are assigned a 100 augmentation probability and the other 65% a 0 augmentation probability.

The *DecreasingLinear* transfer function (charted in figure 3.5) assigns an augmentation probability based on how well the score is, the worst score in c receives a 0% augmentation probability, as the score goes better the probability rises, and a score of 0 receives a 100% augmentation probability. For example, if the S dataset has two images, one is scored 300 and the second one 150 the latter will receive a 50% probability to be augmented.

To assess the effect of the proposal two more transfer functions as baselines were created *NoneAugmentation* (charted in figure 3.6) with augmentation probability of always zero and *ConstantAugmentation* (charted in figure 3.7) with augmentation probability of always one.

3.2.4 Data Augmentation

With $\mathbf{p}$ the images are augmented based on their corresponding probability by modifying the definition 20 to use it as an input, the resulting transformation function is $S' = B(S, \mathbf{p})$ with $S' \in \mathbb{R}^{n_{s'} \times m}$ and $n_s \leq n_{s'}$ to indicate that S' contains both the original images and the augmented ones.

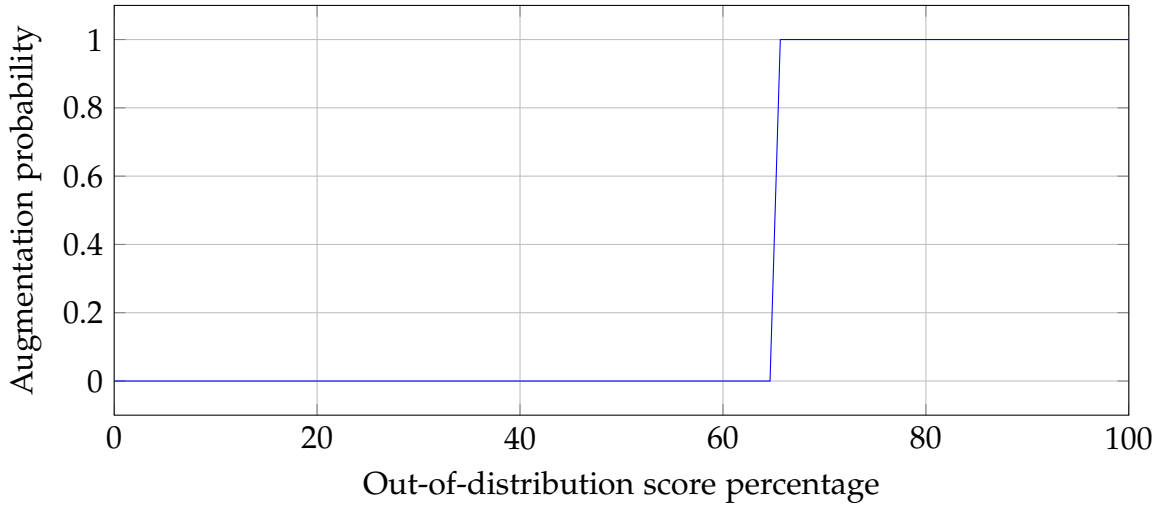


Figure 3.4: *PercentageWiseNegative* transfer function.

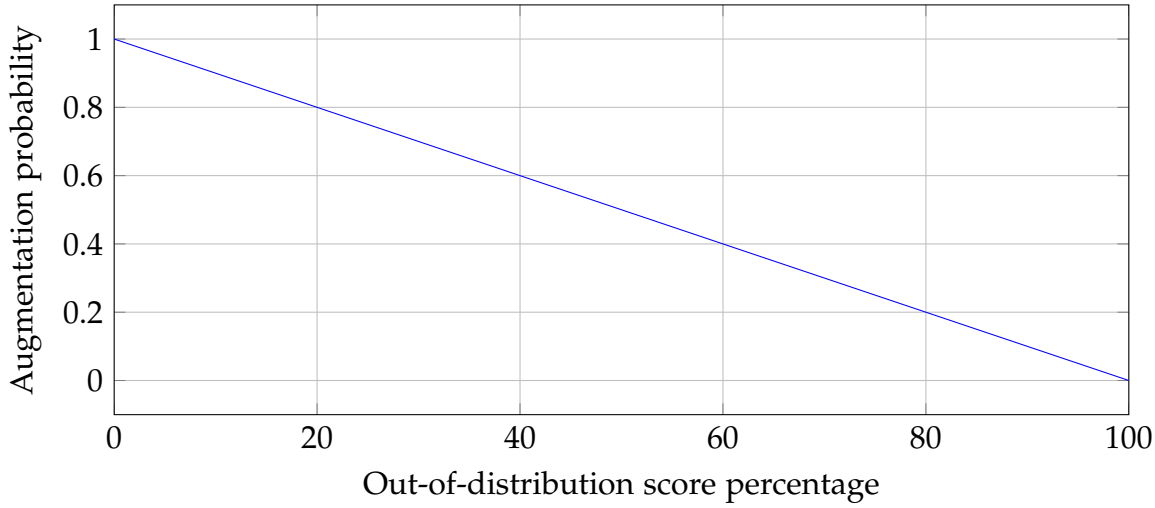


Figure 3.5: *DecreasingLinear* transfer function.

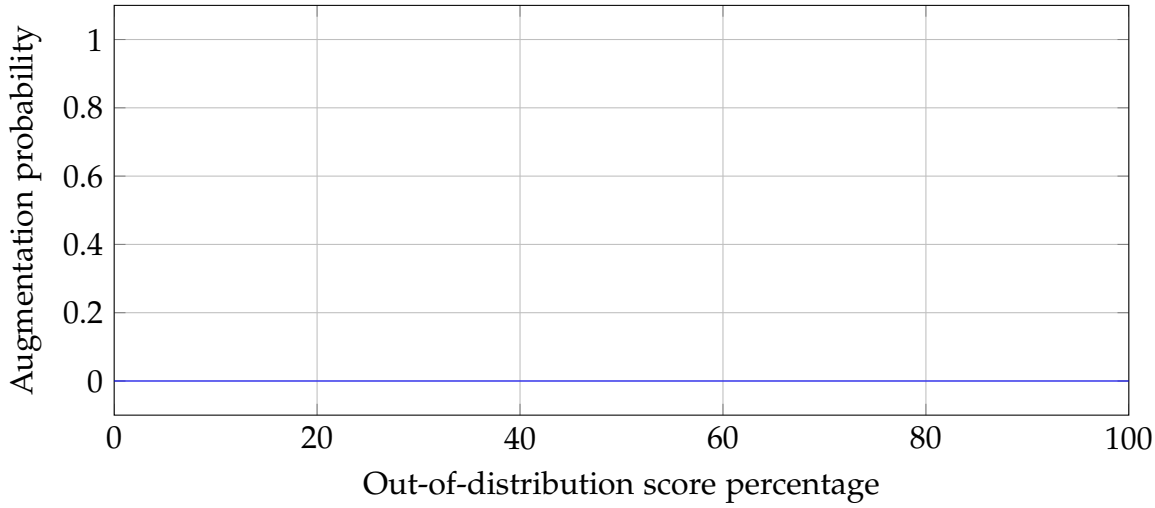


Figure 3.6: *NoneAugmentation* transfer function.

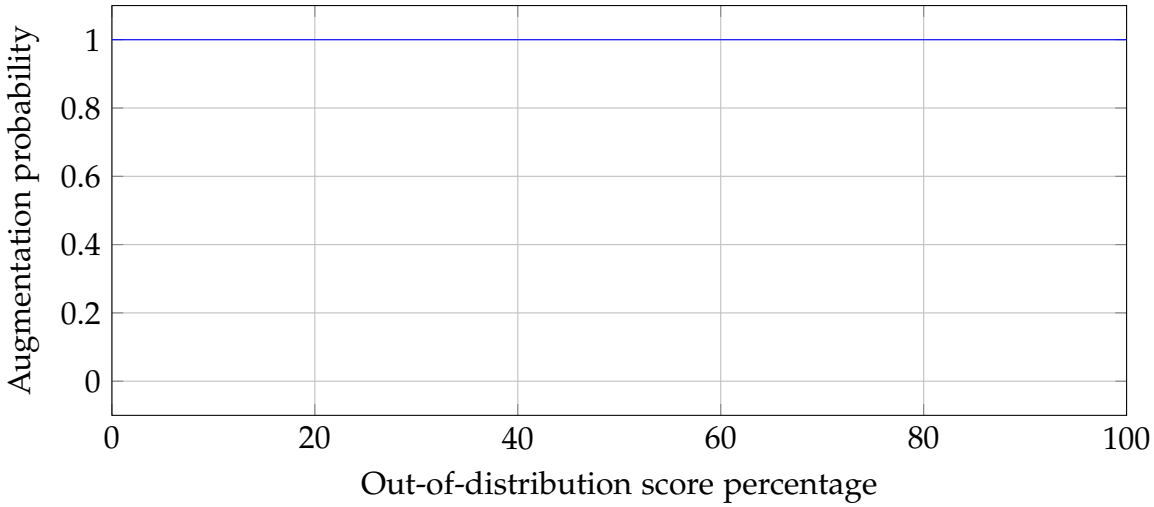


Figure 3.7: *ConstantAugmentation* transfer function.

The Albumentations library [37] is used for the image augmentation. Albumentations implements a long variety of image transform operations optimized for performance, it is a powerful tool for different computer vision tasks, including object classification, segmentation, and detection.

The following transformations were chosen because they do not change the image meaning, which is necessary to avoid confusion for the SDL model but the selection of the augmentations is application dependent thus for use in other contexts a previous analysis is necessary [38].

- **RandomCrop:** resize the image but randomly choose how much will crop the width and height.
- **Resize:** resize the image to a smaller size than the original.
- **Rotate:** rotate the image with a limit of 20 for the angle, more than that would break the correct meaning.
- **Blur:** blurs the image with a limit blur of 5, more than that would be difficult to recognize.
- **OpticalDistortion:** distorts the image of its rectilinear projection.
- **GaussNoise:** applies gaussian noise to the input image, in this case, the variance range for noise (parameter var_limit of GaussNoise function, the higher the stronger the noise) is between 10 and 300.

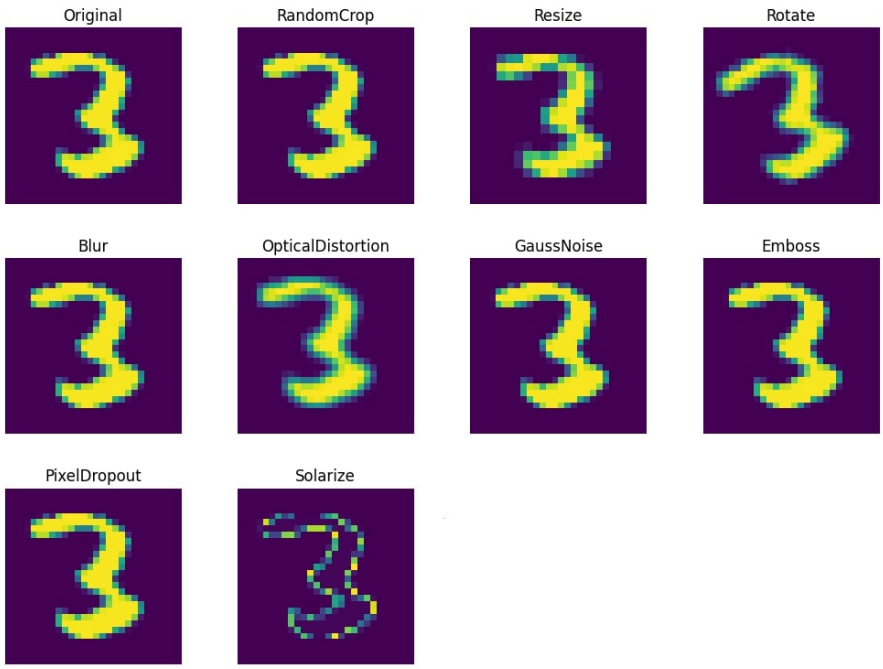


Figure 3.8: Transformations applied to an MNIST image.

- **Emboss:** it makes an emboss in the input image and overlays the result with the original image.
- **PixelDropout:** it sets pixels in 0. In this case the dropout probability is 0.01, which differs from the probability of applying the transformation.
- **Solarize:** inverts all pixel values above a threshold.

When the function to augment images is called it augments the image with one of the previous transformations selected randomly. Performing more than one transformation could result in an incomprehensible image for the model.

Figure 3.8 shows the nine transformations applied to an MNIST [39] image from the original to the subsequent transformations in the order explained prior.

3.3 Hypothesis

A supervised DL model trained with augmented data generated employing the proposed method will measure with statistical significance a higher accuracy in comparison to a supervised DL model trained with regular data in the context of domain adaptation for medical images.

Experimental Factors			**Experimental Subject**
Source (S)	Contamination dataset (F)	Contamination percentage (%)	Transfer function (R(c))
Indiana Covid-19	China Covid-19	25	*NoneAugmentation*
Costa Rica Covid-19	CatsVsDogs	50	*DecreasingLinear*
	Indiana Covid-19 with SnP		*PercentageWisePositive*
	Costa Rica Covid-19 with SnP		*PercentageWiseNegative*
			ConstantAugmentation

Table 3.1: Experiments design.

3.4 Experimental design

A first experimental control to assess the correct working of the OOD score was made inducing ambiguity by comparing two semantically similar source datasets against a target dataset.

For the second part to assess the effect of the augmented data by the proposal on a supervised DL model's accuracy the experiments were designed as a multifactorial test. The proposal represented by the function $O(S + F, G) = S'$ received for each scenario the same target dataset G and source dataset S plus a contamination dataset F in different degrees as described in table 3.1 to replicate a distribution mismatch scenarios.

The training of the supervised DL model was made in a domain adaptation setting where the model is trained first with the generated data S' and secondly with G. Once the model was trained it was used to categorize the images of a subset of G called D_t (definition 4) with a size of 60 images, these images are totally new to the model. The model was evaluated on its accuracy in each scenario.

In table 3.1 the transfer function is the factor to study; source, contamination dataset and contamination percentage are factors to consider. For each transfer function there are 30 batches giving a total of 1800 data entries analyzed ($3 * 2 * 5 * 30 * 2 = 1800$), for each batch the images for the source and contamination datasets were selected randomly as a subset. In table 3.2 source, target, source size and target size are constants throughout the scenarios.

The experiments do not include comparisons with other domain adaptation proposals because these are very dependent on the type of data used to train

Source (S)	Target (G)	Source Size (n_s)	Target Size (n_g)
Indiana	Indiana	100	142
Costa Rica	Costa Rica	100	130

Table 3.2: Experiment constants.

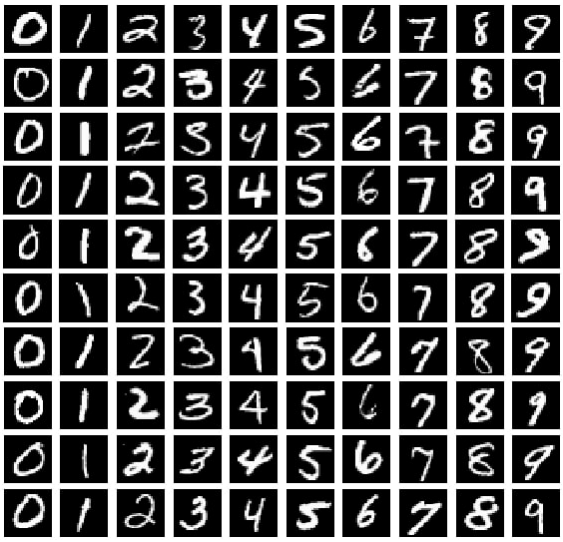


Figure 3.9: MNIST dataset example.

them, so a model trained to categorize images to diagnose e.g. Alzheimer's disease is adjusted to work only in that environment. Likewise, the OOD filtering proposals were made for semi-supervised deep learning models to compare labeled and unlabeled images with each other, not a domain adaptation setting.

3.4.1 Testing Datasets

For the first part a experimental control of the experiments the datasets selected were MNIST [39], handwritten numbers from 0 to 9 as the examples in figure 3.9, and SVHN [40], photos of house-numbered addresses as the examples in figure 3.10. These were selected due to their simplicity where there are not too many defining features and the similarity between them with the intention of confusing the score implementation. Sources have 500 images each and target has 500 images.

By the definition of the hypothesis in section 3.3 the proposal is meant for medical images, specifically for the experiments the context selected is x-ray

Figure 3.10: SVHN dataset example.

Datasets	Indiana Covid-19	Costa Rica Covid-19	China Covid-19	CatsVsDogs	Indiana Covid-19 with SnP	Costa Rica Covid-19 with SnP
Resolution	1400×1400	1907×1791	1300×600	224×224	1400×1400	1907×1791
Description	Indiana Network for Patient Care dataset combined with Valencian Region Medical Image COVID-19 dataset.	Costa Rican private clinic images combined with Cohen and Valencian Region Medical Image COVID-19 dataset.	Chinese dataset with pediatric images dataset combined with Cohen and Valencian Region Medical Image COVID-19 dataset.	Images of cats and dogs.	Indiana Covid-19 with gausian noise.	Costa Rica Covid-19 with gausian noise.

Figure 3.11: Experiment datasets.

image datasets related to Covid-19, the information of the datasets selected is summarized in table 3.11. Two source datasets were selected:

- Indiana Covid-19 dataset: dataset created by [41] when images from Indiana Network for Patient Care without any pathologies and Cohen and Valencian Region Medical Image COVID-19 datasets are combined. The images have a resolution of 1400×1400 pixels. Examples of this dataset are in figure 3.12.

- Costa Rican Covid-19 dataset: dataset created by [41] when images from a Costa Rican private clinic, Clinica Imagenes Medicas Dr. Chavarria Estrada, and Cohen and Valencian Region Medical Image COVID-19 datasets are combined. This dataset includes chest X-rays from 153 patients, aged between 7 and 86 years. Among these patients, 63% are female and 37% are male. Each image has a resolution of 1907×1791 pixels. Examples of this dataset are in figure 3.13.

To replicate the distribution mismatches described in section 2.2 the source

datasets are contaminated with the contamination dataset F based on the following testing categories:

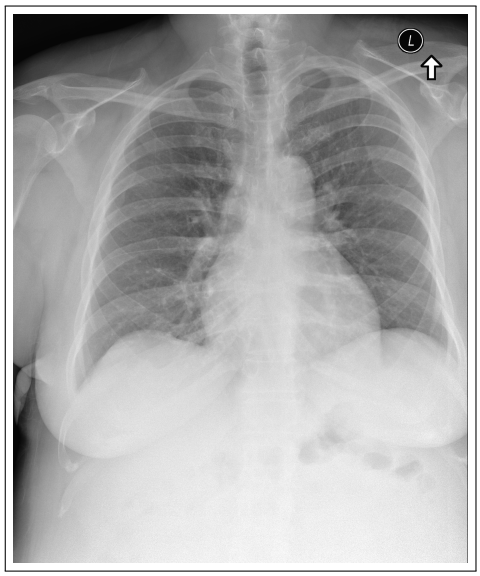
- Distorted: this consists of the same source datasets Indiana Covid-19 and Costa Rican Covid-19 with a heavy Gaussian Noise transformation with a variance noise (parameter var_limit of GaussNoise function, the higher the stronger the noise) range between 1000 and 5000 applied. It is expected to replicate a distribution mismatch using differences in imaging protocols. Examples in figure 3.14.

- Similar: an unassociated medical center China COVID-19 dataset, dataset created by [41] when images from a pediatric chinese dataset and Cohen and Valencian Region Medical Image COVID-19 datasets are combined. The patient sample consists of chinese children and the images have a resolution of 1300×600 pixels. It is expected to replicate a distribution mismatch using differences in patient demographics and imaging devices. Examples in figure 3.15.

- Unrelated: a semantically distinct dataset of pets called CatsVsDogs [42] is used and the images have a resolution of 224×224 pixels. The labels of the cats are 0 the same as if the patient has not Covid-19 and the labels of the dogs are 1 the same as if the patient has Covid-19. It is expected to produce a heavy distribution mismatch by replicating a poor quality of data acquisition. Examples in figure 3.16.

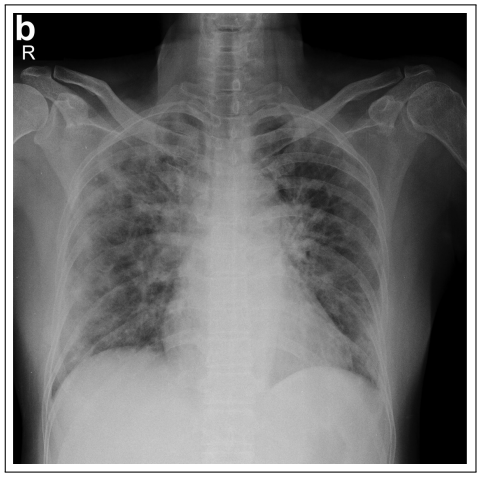
3.4.2 Model and Hyperparameters

The model selected to be trained is Resnet50 [17] a well established variation of a Residual Neural Network (definition 15) with 50 layers deep commonly used for image classification. The Resnet50 implementation of Pytorch was reused and its architecture is displayed in figure 3.18. The input layer of the ResNet50 model is arranged to receive images from the dataset with a resolution of 224×224 and the output layer was replaced so it could classify the images into two categories.

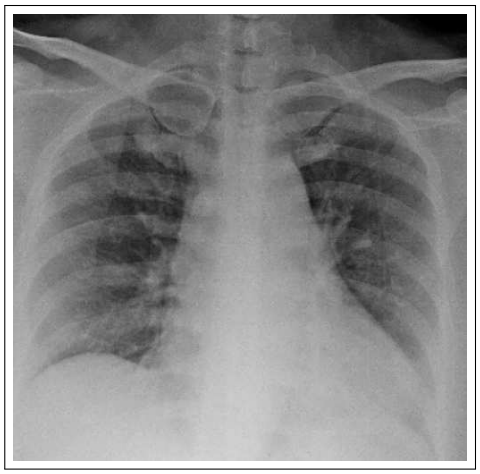


(a) Without COVID-19

(b) Without COVID-19

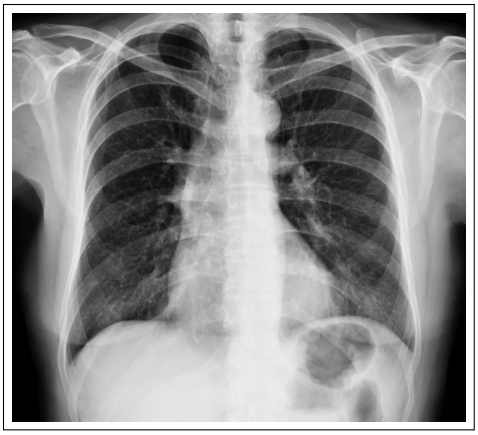


(c) With COVID-19



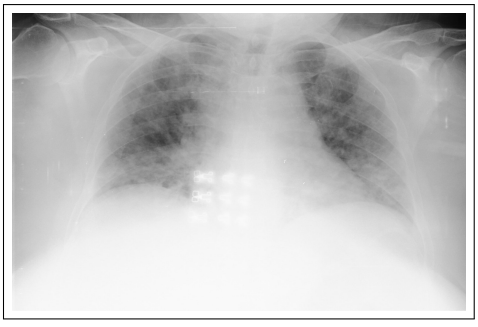
(d) With COVID-19

Figure 3.12: Indiana COVID-19 dataset examples.



(a) Without COVID-19

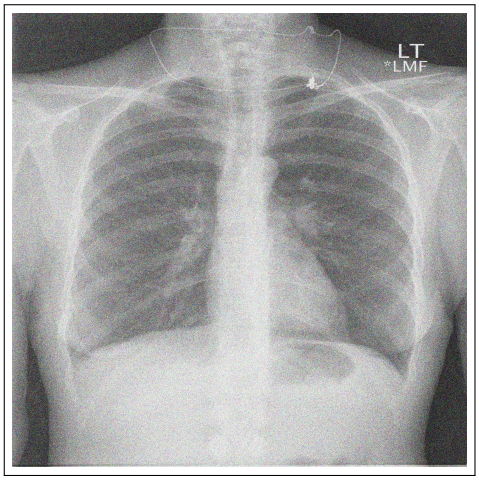
(b) Without COVID-19



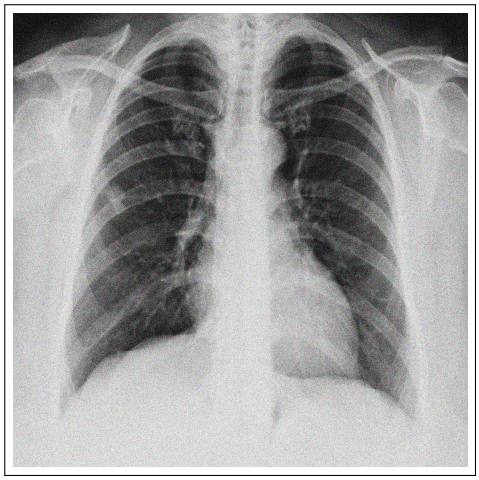
(c) With COVID-19

(d) With COVID-19

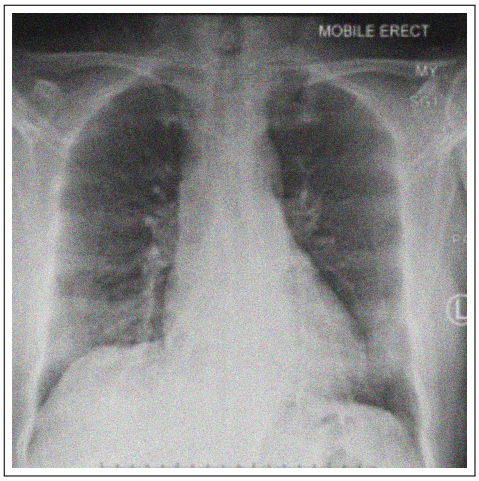
Figure 3.13: Costa Rica COVID-19 dataset examples.



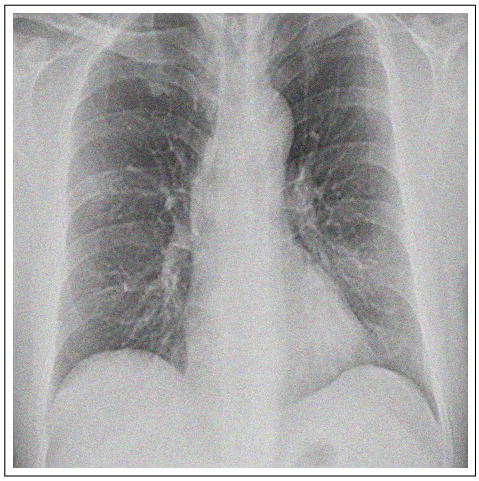
(a) Indiana without COVID-19



(b) Costa Rica without COVID-19

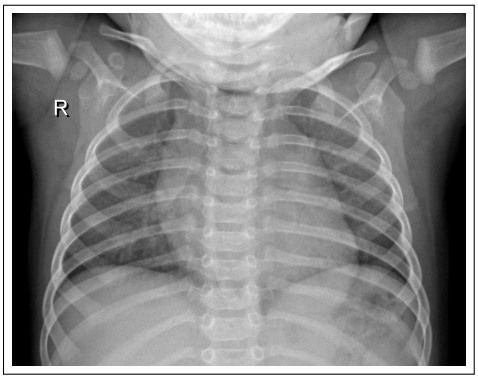


(c) Indiana with COVID-19

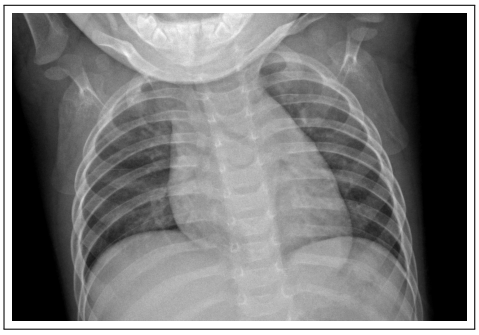


(d) Costa Rica with COVID-19

Figure 3.14: SaltAndPepper noise dataset examples.



(a) Without COVID-19



(b) Without COVID-19

(c) With COVID-19

(d) With COVID-19

Figure 3.15: China COVID-19 dataset examples.

Figure 3.16: CatsVsDogs dataset example.

Learning Rate	Momentum	Batch Size	Number of Epochs	Weight Decay
0.01	0	32	10	0

Figure 3.17: Resnet50 hyperparameters used for the experiments.

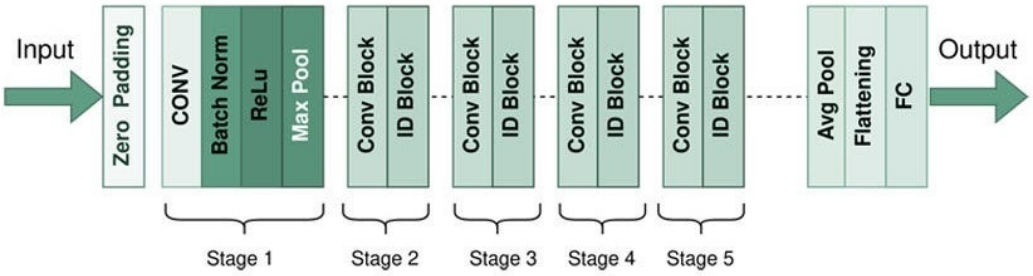


Figure 3.18: Resnet50 model architecture [17].

Source	Mean	Standard Deviation
MNIST	1376.378	1103.336
SVHN	1846.691	677.892

Table 3.3: Means and standard deviations for the OOD score of two sources MNIST and SVHN with 500 images each compared separately against a target MNIST.

The model's weights were created randomly. For the training with S' the hyperparameters (table 3.17) were set with a learning rate of 0.01, momentum of 0, batch size of 32, number of epochs 10 and a weight decay of 0.

3.5 Results

The first part of the experiment results are presented in table 3.3 followed by figure 3.19 where the means and confidence intervals of the same data are depicted.

The totality of the second part of the experiment results are presented in table 3.4, each row represents an experiment scenario, its average accuracy with its standard deviation and the transfer function with the highest accuracy is highlighted in bold. The transfer function performances of table 3.4 are exhibited in figure 3.21, in addition the information is further split by contamination dataset in figure 3.21, by contamination percentage in figure 3.22 and scenarios in figures 3.24, 3.25 and 3.26.

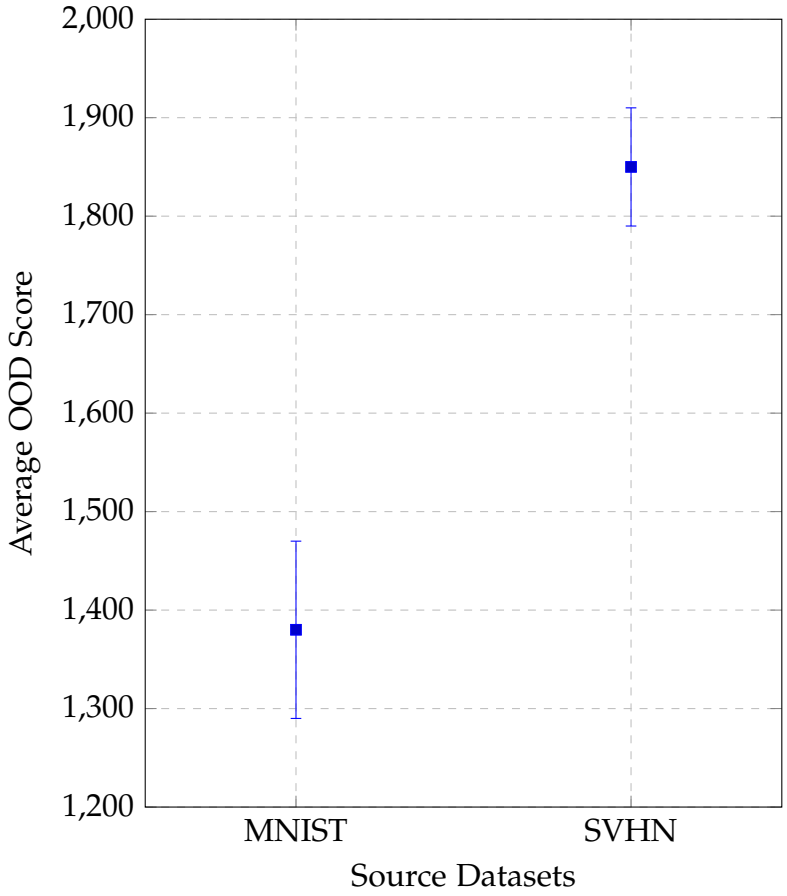


Figure 3.19: Averages OOD score and 95% confidence intervals from two sources MNIST and SVHN with 500 images each compared separately against a target MNIST with 500 images, control experimental data in table 3.3.

	Scenarios				Transfer Functions			
	Source (S)	Contamination dataset (F)	Target (G)	Contamination percentage (%)	Decreasing Linear	Percentage Wise Positive	Percentage Wise Negative	Constant Augmentation
Unrelated	Indiana Covid-19	CatsVsDogs	Indiana Covid-19	25	**60.9 ± 16.5**	52.2 ± 11.1	56.1 ± 12.5	55.4 ± 13.9
	Indiana Covid-19	CatsVsDogs	Indiana Covid-19	50	53.6 ± 10.2	**53.7 ± 10.9**	52.8 ± 16.9	52 ± 7.7
	Costa Rica Covid-19	CatsVsDogs	Costa Rica Covid-19	25	63.3 ± 19.6	53.2 ± 13.2	**63.5 ± 16.7**	60.8 ± 14.5
	Costa Rica Covid-19	CatsVsDogs	Costa Rica Covid-19	50	58 ± 15.1	55.3 ± 12.1	56.4 ± 13.4	**61.6 ± 17.3**
Similar	Indiana Covid-19	China Covid-19	Indiana Covid-19	25	71.2 ± 14.5	68.5 ± 16.4	64.2 ± 14.3	**75.4 ± 12.5**
	Indiana Covid-19	China Covid-19	Indiana Covid-19	50	**73.2 ± 13**	68.2 ± 13.3	66.9 ± 12.1	68.2 ± 13.4
	Costa Rica Covid-19	China Covid-19	Costa Rica Covid-19	25	82.6 ± 15.1	77.6 ± 16.8	82.8 ± 14.6	**86.1 ± 13.1**
	Costa Rica Covid-19	China Covid-19	Costa Rica Covid-19	50	**76 ± 16.6**	74 ± 14.9	72.6 ± 19.1	72.6 ± 18.7
Distorted	Indiana Covid-19	Indiana Covid-19 with SnP	Indiana Covid-19	25	**61.3 ± 17.1**	55.3 ± 10.4	59 ± 14.1	59.4 ± 11.9
	Indiana Covid-19	Indiana Covid-19 with SnP	Indiana Covid-19	50	**65.9 ± 15**	54.5 ± 9.57	58.6 ± 13.7	60.4 ± 14.2
	Costa Rica Covid-19	Costa Rica Covid-19 with SnP	Costa Rica Covid-19	25	**84 ± 14.2**	82 ± 16.8	79 ± 17.7	78.4 ± 19
	Costa Rica Covid-19	Costa Rica Covid-19 with SnP	Costa Rica Covid-19	50	**88.1 ± 11.6**	80.6 ± 16.3	74.6 ± 19.2	82.4 ± 17.5

Table 3.4: Accuracy means and standard deviations obtained from the Resnet50 model when trained with the parameters defined by each experimental scenario (read from left to right) to classify the testing dataset. For each scenario 5 transfer functions were run separately. Each box belonging to the transfer functions represents 30 batches, for each batch the images for the source and contamination datasets were selected randomly. The transfer function with the highest accuracy average for each scenario is written in bold. Experiment scenarios defined in table 3.1. The categories Unrelated, Similar and Distorted are defined in subsection 3.4.1.

.

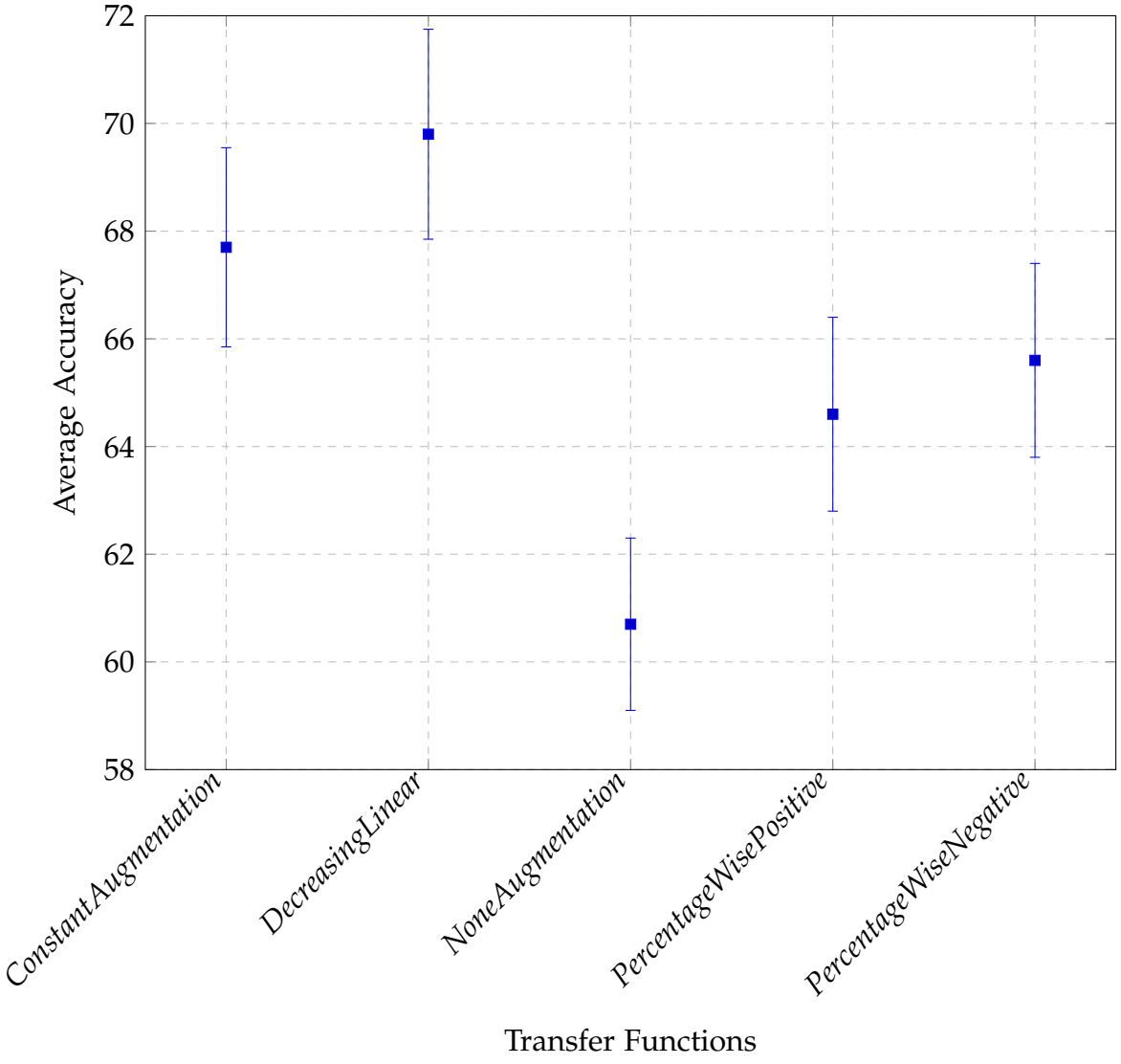


Figure 3.20: Central points represent the average accuracy for each transfer function. Error bars represent the 95% confidence intervals of the means. All scenarios in table 3.4 are gathered and grouped by transfer function.

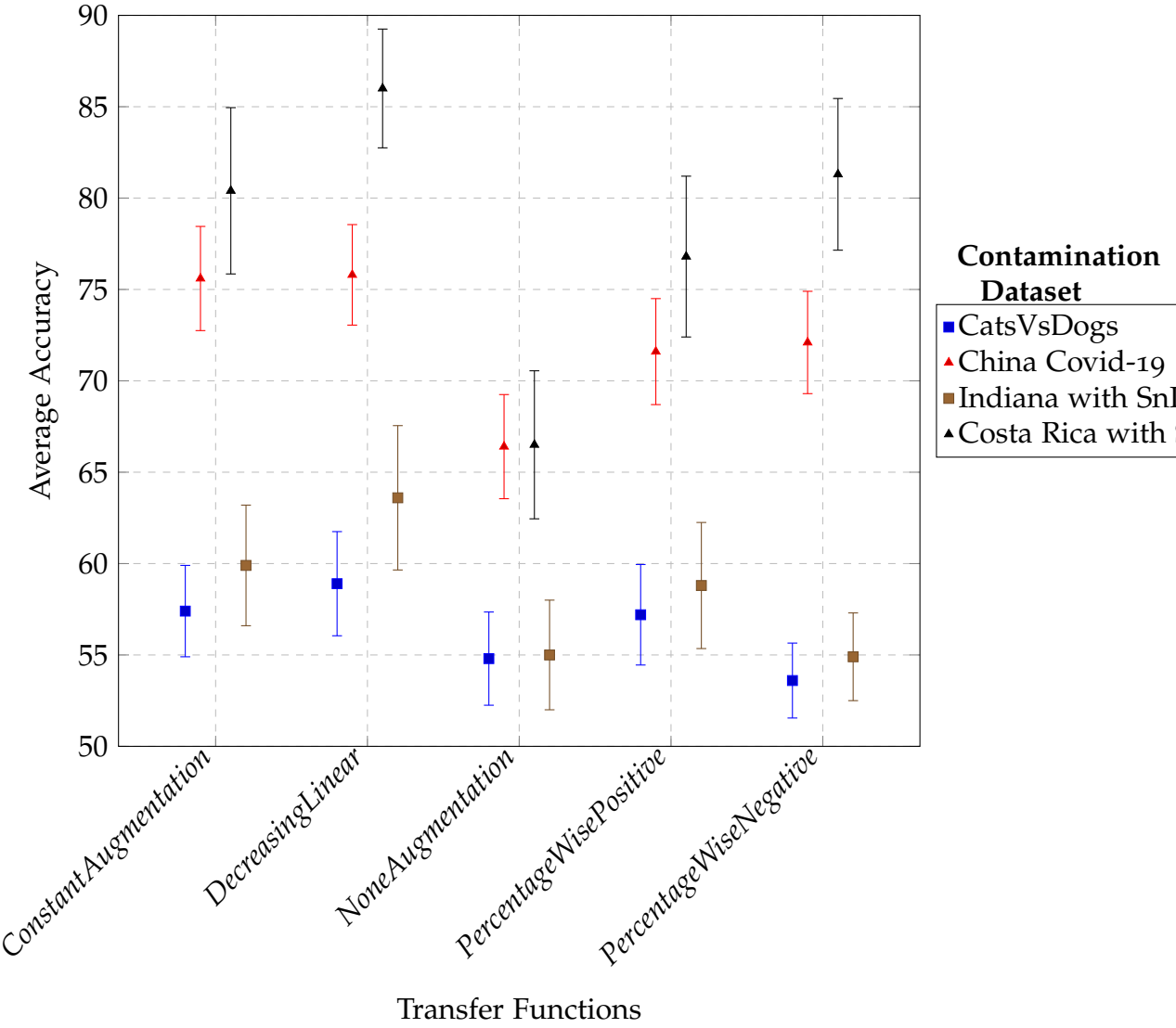


Figure 3.21: Central points represent the average accuracy for each transfer function. Error bars represent the 95% confidence intervals of the means. Data in table 3.4 is grouped by transfer function and contamination dataset.

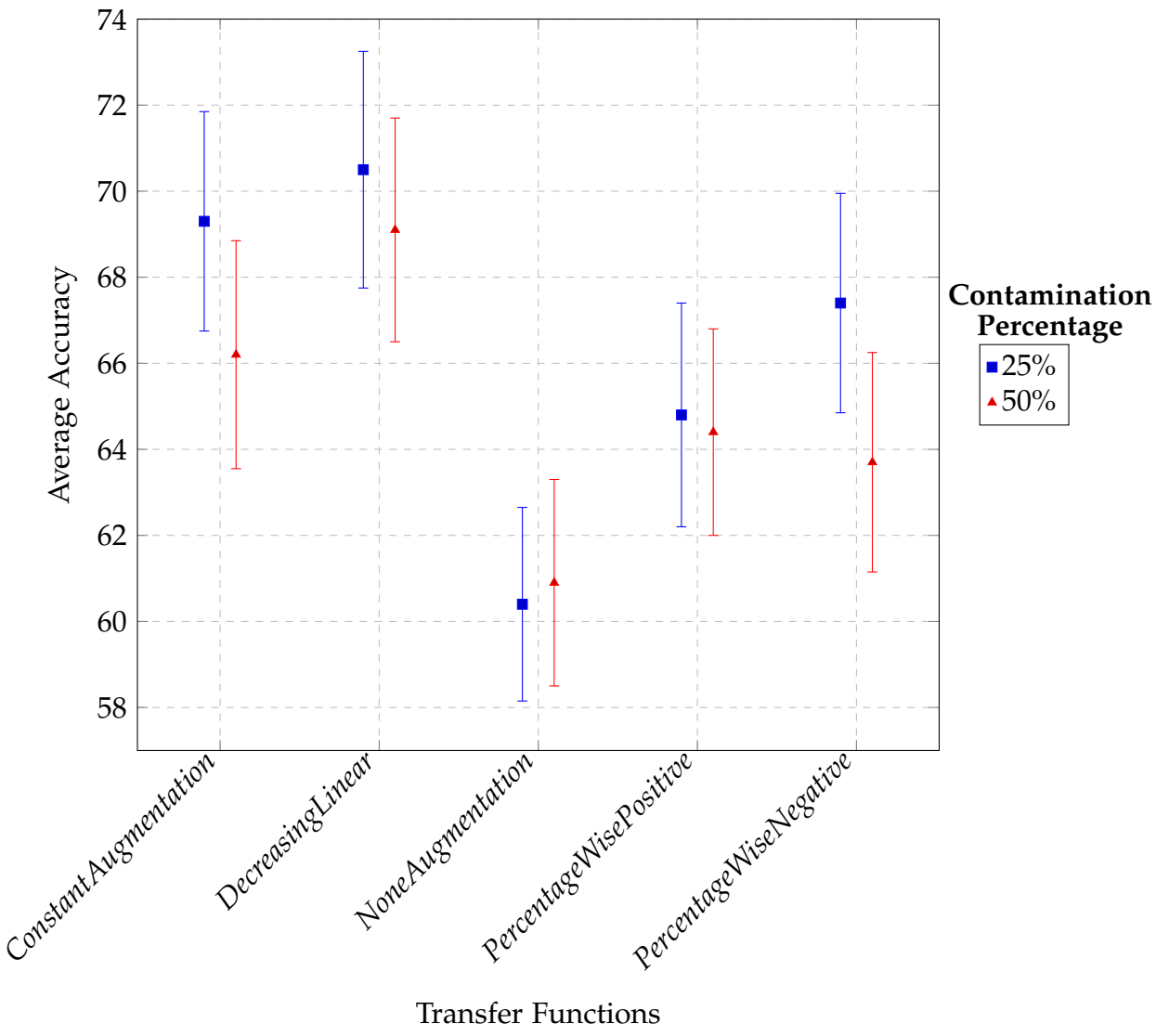


Figure 3.22: Central points represent the average accuracy for each transfer function. Error bars represent the 95% confidence intervals of the means. Data in table 3.4 is grouped by transfer function and contamination percentage.

Comparison			Z	P.unadj	P.adj
ConstantAugmentation	-	DecreasingLinear	-1.67737	9.35×10^{-2}	1.04×10^{-1}
ConstantAugmentation	-	NoneAugmentation	5.711395	1.12×10^{-8}	5.60×10^{-8}
DecreasingLinear	-	NoneAugmentation	7.38876	1.48×10^{-13}	1.48×10^{-12}
ConstantAugmentation	-	PercentageWiseNegative	1.744312	8.11×10^{-2}	1.01×10^{-1}
DecreasingLinear	-	PercentageWiseNegative	3.421677	6.22×10^{-4}	1.24×10^{-3}
NoneAugmentation	-	PercentageWiseNegative	-3.96708	7.28×10^{-5}	1.82×10^{-4}
ConstantAugmentation	-	PercentageWisePositive	2.759861	5.78×10^{-3}	8.26×10^{-3}
DecreasingLinear	-	PercentageWisePositive	4.437226	9.11×10^{-6}	3.04×10^{-5}
NoneAugmentation	-	PercentageWisePositive	-2.95153	3.16×10^{-3}	5.27×10^{-3}
PercentageWiseNegative	-	PercentageWisePositive	1.01555	3.10×10^{-1}	3.10×10^{-1}

Table 3.5: Dunn test results applied all scenarios in table 3.4 gathered and grouped by transfer function. Comparison column lists the pairs of transfer functions being compared. The Z column shows the Z-values that represent how many standard deviations the observed difference is from the null hypothesis of no difference between groups. P.unadj column shows the unadjusted p-values for each comparison. The P.adj column provides the adjusted p-values for multiple comparisons, these values have been adjusted using the Benjamini-Hochberg method to control the false discovery rate, reducing the likelihood of type I errors.

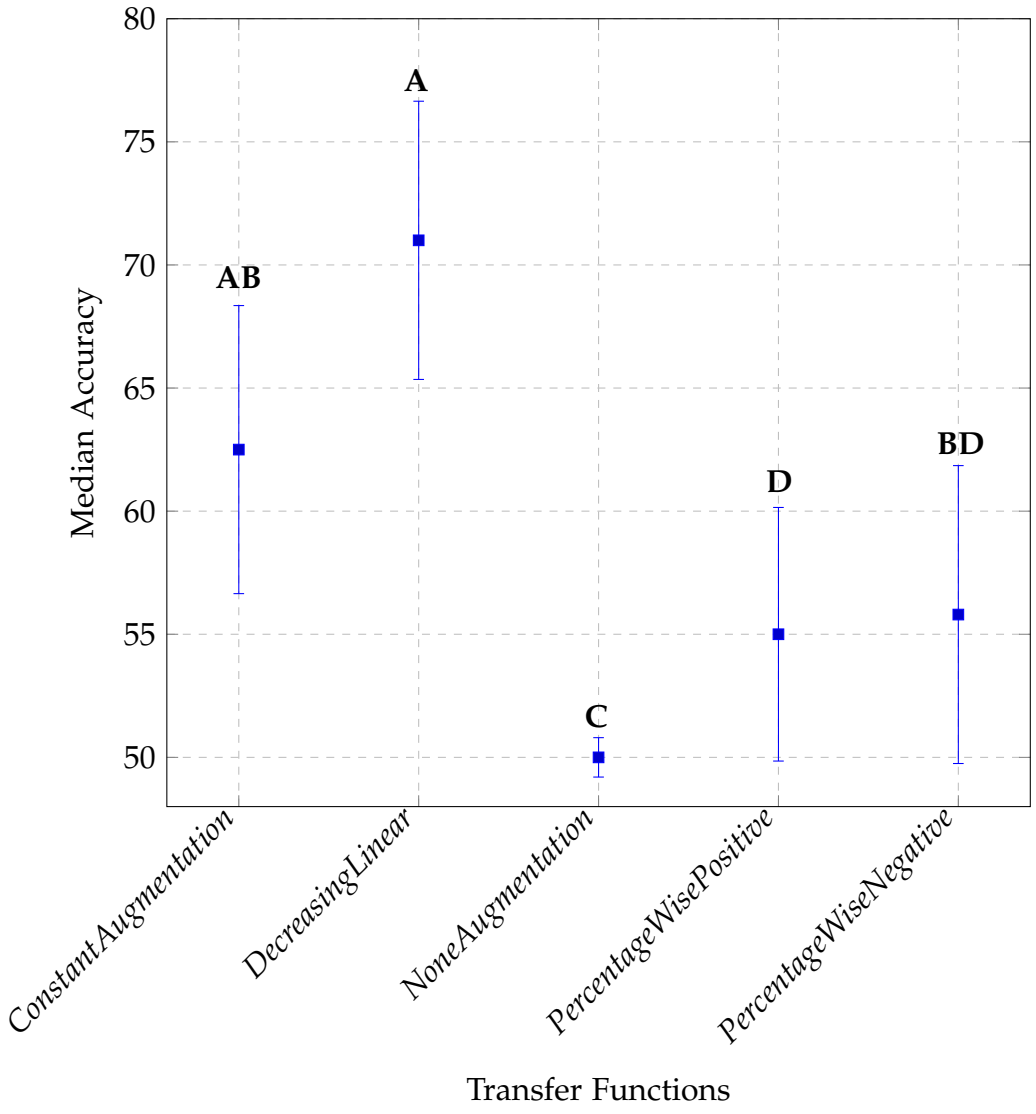


Figure 3.23: Central points represent the median accuracy for each transfer function. Error bars represent the 95% confidence intervals of the medians. Text labels indicate groups that are significantly different from each other based on Dunn's test results in table 3.5. Groups that share a letter are not significantly different.

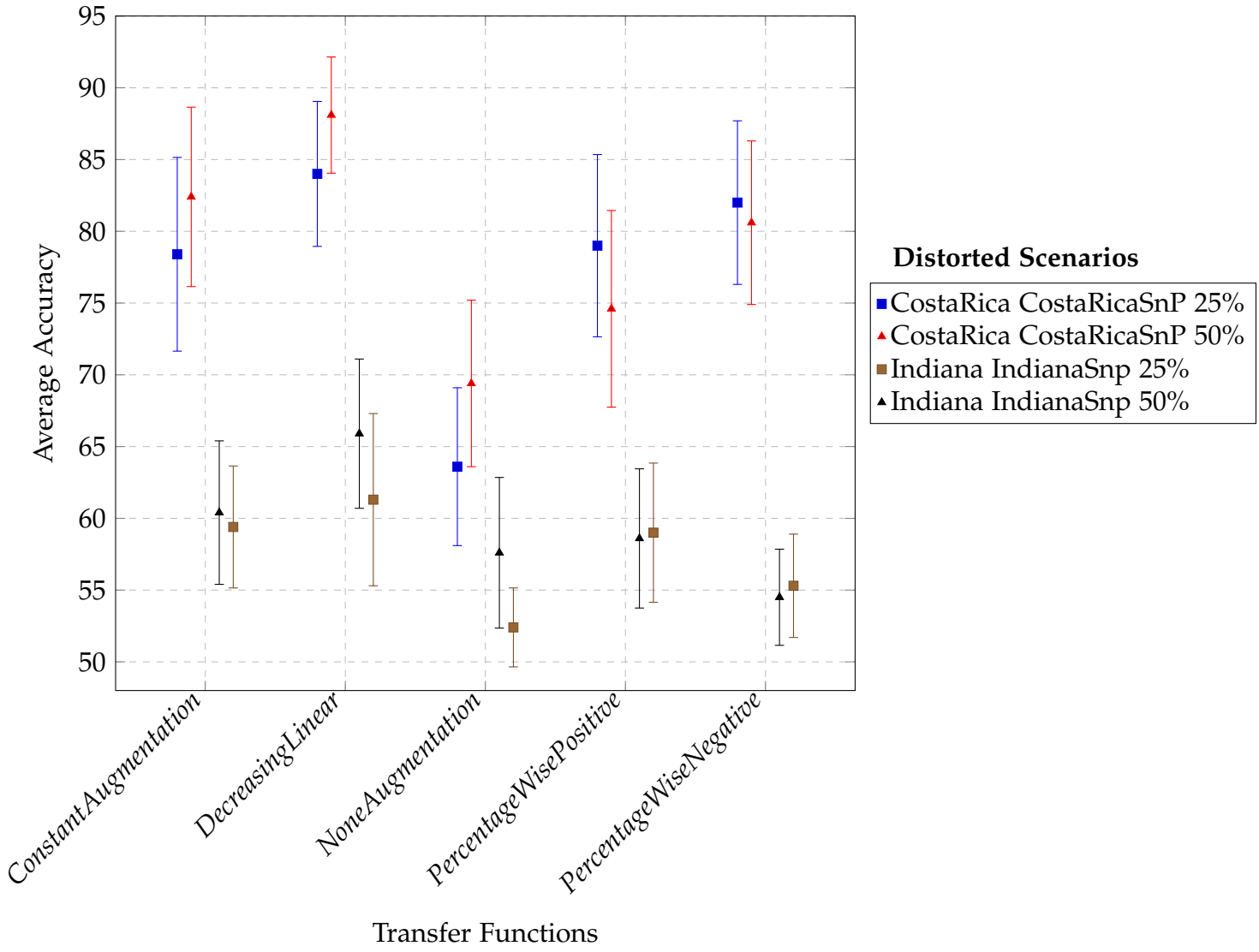


Figure 3.24: Central points represent the average accuracy for each transfer function. Error bars represent the 95% confidence intervals of the means. The data displayed are only the rows associated to **Distorted** in table 3.4, each line in the legend indicates in this order the source, the contamination dataset and contamination percentage.

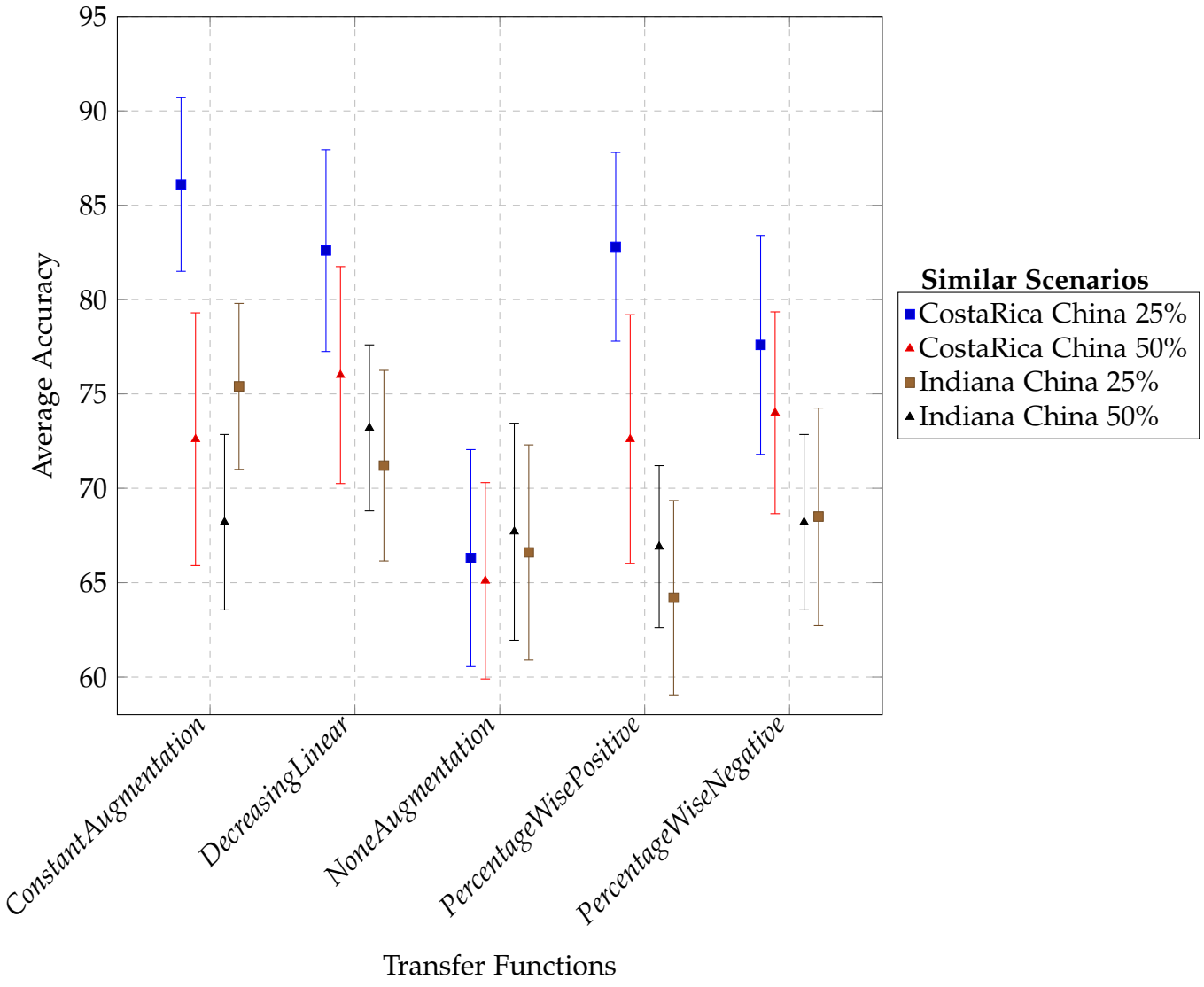


Figure 3.25: Central points represent the average accuracy for each transfer function. Error bars represent the 95% confidence intervals of the means. The data displayed are only the rows associated to **Similar** in table 3.4, each line in the legend indicates in this order the source, the contamination dataset and contamination percentage.

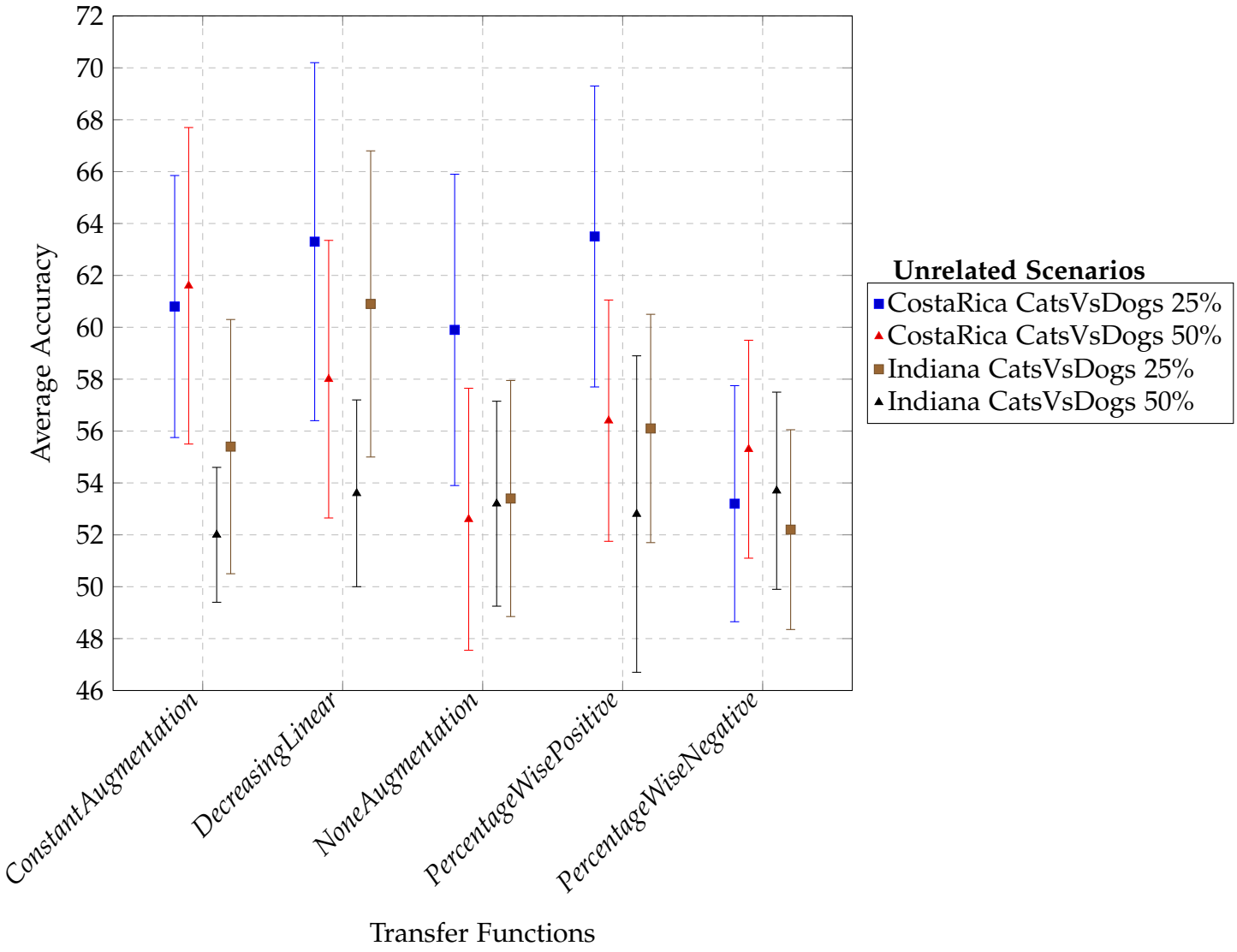


Figure 3.26: Central points represent the average accuracy for each transfer function. Error bars represent the 95% confidence intervals of the means. The data displayed are only the rows associated to **Unrelated** in table 3.4, each line in the legend indicates in this order the source, the contamination dataset and the contamination percentage.

3.6 Results Analysis

The results for the OOD scores measurements displayed in table 3.3 and in figure 3.19 indicate that the OOD scores given to the MNIST source are statistically different from the SVHN OOD scores given by the lack of overlap between the confidence intervals, the MNIST OOD scores are on average 470.313 points lower concluding that the out of distribution score was not confused by the test.

The data of the second part of the experiment did not pass the Levene test check for homoscedasticity with a value of 1.301×10^{-8}, thus the non-parametric statistical test Kruskal–Wallis was used followed by Dunn's Kruskal-Wallis Multiple Comparisons test in combination with the analysis of means and standard deviations.

The severity of the contamination heavily impacted the accuracy of the model as expected presenting a mean accuracy close to 50% in some scenarios such as with the contamination dataset CatsVsDogs with contamination percentages of 25 and 50 in table 3.4. The overall accuracy of the transfer functions in figure 3.20 indicates that *DecreasingLinear* is statistically different from *NoneAugmentation*, *PercentageWisePositive* and *PercentaWiseNegative* and comparable to *ConstantAugmentation*, at the same time *ConstantAugmentation*, *PercentageWisePositive* and *PercentaWiseNegative* are comparable between them and statistically different from *NoneAugmentation*, in both analysis it is confirmed that a transfer function is statistically different from other by the lack of overlap between confidence intervals. The Kruskal-Wallis test corroborates that there is a significant difference between transfer functions with a p-value of 6.702×10^{-13}.

The results of table 3.4 were further split into contamination datasets in figure 3.21. For the contamination dataset CatsVsDogs the *PercentaWiseNegative* did not have much impact in comparison versus *NoneAugmentation*, the other three functions had similar performances slightly above *NoneAugmentation*. For the contamination dataset Indiana Covid-19 with SnP *DecreasingLinear* proved to be above the rest of the functions, where *ConstantAugmentation* and *PercentageWisePositive* had similar accuracies and *PercentaWiseNegative* performed almost identically to *NoneAugmentation*. For contamination dataset China Covid-19 *De-*

creasingLinear and *ConstantAugmentation* are equivalent and both are above the rest. For the contamination dataset Costa Rica Covid-19 with SnP had the highest accuracies, *DecreasingLinear* was considerably better than the rest, at the same time a unique result occurs with *PercentageWiseNegative* performed as well as *ConstantAugmentation*.

The results of table 3.4 were grouped by contamination percentage in figure 3.22, the functions had similar performances against themselves between the settings of 25% and 50%, the *DecreasingLinear* when presented with a big difference in the distribution as the case for 50% had a noticeable impact being superior from the rest even *ConstantAugmentation*, *PercentaWiseNegative* too had a better performance but for the 25%.

The results of the Dunn test in table 3.5, in which all the transfer functions and baselines were compared against each other, are displayed in figure 3.23 with the addition of a Compact Letter Display comparison, *DecreasingLinear* and *ConstantAugmentation* had the best results and belong to the same group A, *PercentageWiseNegative* was slightly better to have no significant difference with *ConstantAugmentation* belonging to group B but not good enough to separate from group D with *PercentageWisePositive*, lastly *NoneAugmentation* had little to no impact on the accuracy being alone in group C.

To know the effective transfer function for each testing category (described previously in subsection 3.4.1) the transfer functions are further analyzed by each scenario composed by the factors source, target, contamination dataset and contamination percentage as presented in table 3.4.

- Similar: data displayed in figure 3.25.
  - Contamination dataset China Covid-19 with contamination percentage 25%: when the similarity between source and target is small the recommended transfer function is *ConstantAugmentation* followed by *DecreasingLinear* indicating that when the distribution mismatch is low the correct strategy is to augment without regard, this is true for both Sources Costa Rica Covid-19 and Indiana Covid-19.
  - Contamination dataset China Covid-19 with contamination percentage 50%: as the dissimilarity increased the *DecreasingLinear* took the

first position and *ConstantAugmentation* the second one.

- Distorted: data displayed in figure 3.24, this presented a scenario in which the variety in the augmentation was positive for the model having *DecreasingLinear* as the best option for all scenarios in this category. For the second-best transfer function they are as follows:
  - Contamination dataset Costa Rica Covid-19 with SnP with contamination percentage 25%: *PercentageWiseNegative*.
  - Contamination dataset Costa Rica Covid-19 with SnP with contamination percentage 50%: *ConstantAugmentation*.
  - Contamination dataset Indiana Covid-19 with SnP with contamination percentage 25%: *ConstantAugmentation*
  - Contamination dataset Indiana Covid-19 with SnP with contamination percentage 50%: *ConstantAugmentation*
- Unrelated: data displayed in figure 3.26, the performance for the model was unfavorable as expected with the intense contamination from CatsVsDogs, only in two scenarios a transfer function was slightly better than the others:
  - For source Costa Rica Covid-19 with contamination dataset CatsVsDogs and contamination percentages 50%: *ConstantAugmentation*.
  - For source Indiana Covid-19 with contamination dataset CatsVsDogs and contamination percentages 25%: *DecreasingLinear*.

To answer the research questions in section 3.1 the results of the experiments led to the following conclusions:

DecreasingLinear displayed to be better than *NoneAugmentation* in all scenarios and as good as *ConstantAugmentation* with the added benefit of generating between 20% to 30% fewer images than *ConstantAugmentation*, resulting in lower training times for the model. When the gap in distribution closes the *ConstantAugmentation* transfer function starts to produce better results, on the other side when the gap in distribution is too wide the model's accuracy is too impacted

and there was no difference between the transfer functions. Since *DecreasingLinear* favors the augmentation of images with a high score, within the context of high distribution mismatch, it can be concluded it improves the supervised DL model.

By the hypothesis defined in section 3.3 and the results it can be concluded that the ResNet50 supervised DL model trained with the data generated employing GUIDATFUN method measured with statistical significance with a higher accuracy in comparison to trained with regular data in the context of domain adaptation for medical images. When contemplating the results as a whole this was true for *DecreasingLinear*, *PercentageWisePositive* and *PercentageWiseNegative* transfer functions.

3.7 Discussion

For the OOD data filtering technique [2] when the distribution mismatch is too great the image is discarded entirely, one limitation to this approach is that if there are few outliers in the unlabeled data the model will see little to no improvement as described by [43], the GUIDATFUN has a great positive impact on these scenarios. The unrelated testing category did not have improvement from GUIDATFUN thus a combined approach that can discard harmful images and weakly or strongly augment relevant images seems to be the way to go to increase the flexibility, avoid overfitting and prone the model of mistakes in the training data.

Domain adaptation for Alzheimer's (section 2.3.2) disease diagnostics results [33] showed (i) that training on only the target training set yields better results than the naive combination (union) of source and target training sets and (ii) that domain adaptation with instance weighting yields the best classification results. This is supported by the fact that *NoneAugmentation* transfer function results were close to 50% accuracy and that *DecreasingLinear* is the recommendation on the majority of the scenarios but it is a new approach to avoid overfitting that does not inquire on the modification of the model thus its flexibility.

For domain adaptation for Autism spectrum disorder (section 2.3.2) the im-

plication of transforming features to align domains might reduce the discriminative power of the features, especially if the transformation overly smooths or generalizes the features, another ill effect is finding the optimal subspace can be challenging and computationally expensive [34], for both is not the case for GUIDATFUN which keeps the original images as well as the augmented and the extraction of the feature space is made with an already pretrained model.

Even though *PercentageWiseNegative* and *PercentageWisePositive* had very comparable results *PercentageWiseNegative* augments 35% of the images versus the 65% *PercentageWisePositive*, further tests were *PercentageWiseNegative* augment the 65% of the lower OOD scored images could favor its performance.

3.8 Future Work

There is still research to be done regarding the applicability of a data augmentation policy for images with a low OOD score. Take the following context: source and target datasets are x-ray images of lung cancer, the target has 90% of advanced stages of the illness and the source has an 80% of premature cases (concept shift), in this context it can be counterproductive to assign a low augmentation probability to the images with a low score if the objective of the model is to diagnose patients in all stages of the disease.

The testing size for GUIDATFUN was inspired by medical images and in this context the quantity of training data is often limited thus the small size selected for source and target in the experiments, remains to be seen if this approach scales or diminishes with higher amounts of images.

The extraction of the features is made with AlexNet for its cheap computational cost but a more recent model such as DenseNet could be used to improve the accuracy of the feature extractor and thus the out-of-distribution score at the cost of computational resources.

One of the primary challenges in medical image research is the scarcity of labeled data. To address this, recent studies have increasingly adopted unsupervised domain adaptation methods, which allow models to be fine-tuned without using labeled target data [32]. Future research could use GUIDATFUN to ad-

dress the particular challenges for the context of unsupervised DA to address the distribution mismatch between unlabeled source and unlabeled target data to have a more balanced representation as a training dataset for an unsupervised model.

Most DA methods are single-source, however real-world applications often involve multiple source domains, such as data from various medical facilities. Future work could investigate the application of GUIDATFUN on a multi-source domain adaptation context where multiple sources could be augmented based on how closely resemble the distribution of a single target dataset.

Given the transfer function designs of GUIDATFUN the quantity of necessary augmented images are reduced in comparison with a classical approach of data augmentation, for example *PercentaWiseNegative* only augmented 35% of the images but had similar accuracies versus *ConstantAugmentation*. New research question would be, how the selected images with a guided data augmentation policy affects the training times for deep learning model?

References

[1] A. Rampun, R. Zwiggelaar, A. Hamidinekoo, E. Denton, and K. Honnor, "Deep Learning in Mammography and Breast Histology, an Overview and Future Trends Article in Medical Image Analysis · March," Medical Image Analysis, vol. 47, pp. 45–67, 2018.

[2] S. Calderon-Ramirez, S. Yang, D. Elizondo, and A. Moemeni, "Dealing with Distribution Mismatch in Semi-supervised Deep Learning for Covid-19 Detection Using Chest X-ray Images: A Novel Approach Using Feature Densities," 2020 25th International Conference on Pattern Recognition (ICPR), 2021.

[3] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Y. Ucla, "Weakly-and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.

[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.

[5] S. Sun, H. Shi, and Y. Wu, "A survey of multi-source domain adaptation," Information Fusion, vol. 24, pp. 84–92, 2015.

[6] G. Mårtensson, D. Ferreira, T. Granberg, L. Cavallin, K. Oppedal, A. Padovani, I. Rektorova, L. Bonanni, M. Pardini, M. G. Kramberger, J.-P. Taylor, J. Hort, J. Snædal, J. Kulisevsky, F. Blanc, A. Antonini, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, A. Simmons, D. Aarsland, and E. Westman, "The reliability of a deep learning model in clinical out-of-distribution mri data: A multicohort study," Medical Image Analysis, vol. 66, p. 101714, 2020.

[7] J. Tan, A. Au, Q. Meng, and B. Kainz, "Semi-supervised Learning of Fetal Anatomy from Ultrasound," Lecture Notes in Computer Science (including

subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11795 LNCS, pp. 157–164, oct 2019.

[8] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," Pattern recognition, vol. 41, no. 12, pp. 3600–3612, 2008.

[9] O. C. Baltatu, M. Nishio, H. R. Marateb, M. Elgendi, M. U. Nasir, Q. Tang, D. Smith, J.-P. Grenier, C. Batte, B. Spieler, W. D. Leslie, C. Menon, R. R. Fletcher, N. Howard, R. Ward, W. Parker, and S. Nicolaou, "The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective," Front. Med, vol. 8, p. 629134, 2021.

[10] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling, "Do not have enough data? deep learning to the rescue!," in Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp. 7383–7390, 2020.

[11] L. Rice, E. Wong, and J. Z. Kolter, "Overfitting in adversarially robust deep learning," in Proceedings of the 37th International Conference on Machine Learning (H. D. III and A. Singh, eds.), pp. 8093–8104, PMLR, 2020.

[12] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. D'Oliveira, H. Eichner, S. El Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecní, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, vol. 14, pp. 1–210, dec 2019.

[13] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, J. Woo, et al., "Deep unsupervised domain adaptation: A review of recent advances and perspectives," APSIPA Transactions on Signal and Information Processing, vol. 11, no. 1, 2022.

[14] T. Mitchell, Machine Learning. New York, USA: McGraw-Hill, 1997.

[15] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature 2015 521:7553, vol. 521, pp. 436–444, 5 2015.

[16] J. Yun, Y. Cho, S. M. Lee, J. Hwang, J. Lee, Y.-M. Oh, S.-D. Lee, L.-C. Loh, C.-K. Ong, J. B. Seo, and N. Kim, "Deep radiomics-based survival prediction in patients with chronic obstructive pulmonary disease," Scientific Reports, vol. 11, p. 15144, 07 2021.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[20] A. Gleason, Fundamentals of Abstract Analysis, pp. 223–228. A K Peters/CRC Press, 10 2018.

[21] O. Elharrouss, Y. Akbari, N. Almadeed, and S. Al-Maadeed, "Backbones-review: Feature extractor networks for deep learning and deep reinforcement learning approaches in computer vision," Computer Science Review, vol. 53, p. 100645, Aug. 2024.

[22] H. Kasban, M. El-Bendary, and D. Salama, "A comparative study of medical imaging techniques," International Journal of Information Science and Intelligent System, vol. 4, no. 2, pp. 37–58, 2015.

[23] R. Moradi, R. Berangi, and B. Minaei, "A survey of regularization strategies for deep models," Artificial Intelligence Review, vol. 53, no. 6, pp. 3947–3986, 2020.

[24] T. Shyamalee and D. Meedeniya, "Cnn based fundus images classification for glaucoma identification," in 2022 2nd International Conference on Advanced Research in Computing (ICARC), pp. 200–205, IEEE, 2022.

[25] M. Nishio, S. Noguchi, and K. Fujimoto, "Automatic pancreas segmentation using coarse-scaled 2d model of deep learning: usefulness of data augmentation and deep u-net," Applied Sciences, vol. 10, no. 10, p. 3360, 2020.

[26] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song, "Anomalous example detection in deep learning: A survey," IEEE Access, vol. 8, pp. 132330–132347, 2020.

[27] M. Nishio, C. Muramatsu, S. Noguchi, H. Nakai, K. Fujimoto, R. Sakamoto, and H. Fujita, "Attribute-guided image generation of three-dimensional computed tomography images of lung nodules using a generative adversarial network," Computers in Biology and Medicine, vol. 126, p. 104032, 2020.

[28] E. Wu, K. Wu, and W. Lotter, "Synthesizing lesions using contextual gans improves breast cancer classification on mammograms," arXiv preprint arXiv:2006.00086, 2020.

[29] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, "Auggan: Cross domain adaptation with gan-based data augmentation," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 718–731, 2018.

[30] J. Huo, V. Vakharia, C. Wu, A. Sharan, A. Ko, S. Ourselin, and R. Sparks, "Brain lesion synthesis via progressive adversarial variational auto-encoder," in International Workshop on Simulation and Synthesis in Medical Imaging, pp. 101–111, Springer, 2022.

[31] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, "Pixelvae: A latent variable model for natural images," arXiv preprint arXiv:1611.05013, 2016.

[32] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," IEEE Transactions on Biomedical Engineering, vol. 69, no. 3, pp. 1173–1185, 2021.

[33] C. Wachinger and M. Reuter, "Domain adaptation for alzheimer's disease diagnostics," NeuroImage, vol. 139, pp. 470–479, 10 2016.

[34] M. Wang, D. Zhang, J. Huang, P.-T. Yap, D. Shen, and M. Liu, "Identifying autism spectrum disorder with multi-site fmri via low-rank domain adaptation," IEEE Transactions on Medical Imaging, vol. 39, no. 3, pp. 644–655, 2020.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems (F. Pereira and C.J. Burges and L. Bottou and K.Q. Weinberger, ed.), Curran Associates, Inc., 2012.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, Ieee, 2009.

[37] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," Information, vol. 11, no. 2, 2020.

[38] O. Stephen, M. Sain, U. J. Maduh, and D.-U. Jeong, "An efficient deep learning approach to pneumonia classification in healthcare," Journal of healthcare engineering, vol. 2019, no. 1, p. 4180949, 2019.

[39] L. Deng, "The mnist database of handwritten digit images for machine learning research," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141–142, 2012.

[40] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al., "Reading digits in natural images with unsupervised feature learning," in NIPS workshop on deep learning and unsupervised feature learning, vol. 2011, p. 7, Granada, Spain, 2011.

[41] S. Calderon-Ramirez, S. Yang, A. Moemeni, D. Elizondo, S. Colreavy-Donnelly, L. F. Chavarría-Estrada, and M. A. Molina-Cabello, "Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images," Applied Soft Computing, vol. 111, p. 107692, 2021.

[42] Kaggle, "Dogs vs. cats dataset," 2013. Accessed: 2024-05-26.

[43] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in Computer Vision – ECCV 2020 (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 438–454, Springer International Publishing, 2020.