UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Features transfer learning between domains for image and video recognition tasks

Fernando Pereira dos Santos

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Fernando Pereira dos Santos

Features transfer learning between domains for image and video recognition tasks

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
March 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
        Pereira dos Santos, Fernando
P436f      Features transfer learning between domains for
        image and video recognition tasks / Fernando
        Pereira dos Santos; orientador Moacir Antonelli
        Ponti. -- São Carlos, 2020.
           99 p.


           Tese (Doutorado - Programa de Pós-Graduação em
        Ciências de Computação e Matemática Computacional) --
         Instituto de Ciências Matemáticas e de Computação,
        Universidade de São Paulo, 2020.


           1. features transfer learning. 2.  deep
        learning. 3. generalization measures. 4.  cross-
        domain. 5. manifold alignment. I. Antonelli Ponti,
        Moacir, orient. II. Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Fernando Pereira dos Santos

Aprendizado de características e sua transferência entre domínios em tarefas de reconhecimento em imagens e vídeos

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Supervisor: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
Março de 2020

To my love and wife, Ana.

ACKNOWLEDGEMENTS

First, I would like to thank God for allowing me to find the wonderful world of science. Many say that science and religion are opposites, but I believe they are coexisting and everything comes from God.

I also would like to thank the most special person, a brilliant and beautiful woman, who chose to face the walk of life with me, Ana. Thank you for helping me, advising me, and encouraging me from the beginning.

I am very grateful to my supervisor, Moacir Antonelli Ponti, for the advice, guidance, friendship, and for having believed that we could make this project something concrete. I have learned a lot about research, leadership, and teaching with you.

I am very grateful especially to my parents Claudecir and Mileide, to my sister Monize, to my in-laws Paulo and Silmara, to Aunt Naide, and my sisters-in-law Dânia and Bruna.

I am very grateful to friends who have collaborated directly with this research: Leonardo Sampaio Ferraz Ribeiro, Gabriel de Barros Paranhos da Costa, and Cemre Zor. Also, I am very grateful for the opportunities offered by Professors Maria Cristina Ferreira de Oliveira, Rosane Minghim, and Josef Kittler.

I am very grateful to Diego Renan Bruno, Gabriel Tomiatti, Leandro Mugnaini, Diego Silva Cintra, Fábio Felix, Germain Zanabria, Filomen Incahuanaco, Nicolas Roque, Evandro Ortigossa, Aline Becher, Tu Bui, Jaqueline Alvarenga, and Leonardo Martinussi for the friendship.

And, I am very grateful to CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for financial support throughout this research.

Thank all very much for the support.

“Com a sabedoria se edifica a casa, e com o entendimento ela se firma;
pelo conhecimento se enchem as câmaras de todas as substâncias preciosas e deleitáveis.
O homem sábio é forte, e o homem de conhecimento aumenta a força;
(Bíblia Sagrada, Provérbios 24:3-5)”

RESUMO

SANTOS, F. P. **Aprendizado de características e sua transferência entre domínios em tarefas de reconhecimento em imagens e vídeos**. 2020. 99 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

A transferência de aprendizado de características objetiva reaproveitar o conhecimento adquirido previamente em um conjunto de dados de origem para aplicá-lo em outro domínio ou tarefa alvo. Um requerimento para a transferência de conhecimento é a qualidade dos espaços de características obtidos, em que métodos de aprendizado profundo são altamente aplicados por proverem descritores discriminativos e generalizáveis, em particular para imagens e vídeos, que são o foco desse trabalho. Neste contexto, as principais questões incluem: o que transferir — alinhando as distribuições dos dados de origem e alvo, e ajustando os parâmetros para aumentar a capacidade de generalização dos modelos; como transferir — investigando métodos que trabalham tanto sobre os espaços de características quanto sobre os modelos aprendidos; e quando transferir — estudando quais dados são mais adequados para transferência, considerando discrepâncias entre os dados origem e alvo, como diferentes meios de aquisição, presença de objetos confusos e iluminação, entre outros. Esse trabalho defende o foco na transferência dos espaços de características aprendidos por redes neurais convolucionais, em particular na investigação do potencial descritivo das camadas iniciais e internas das redes convolucionais profundas e a aproximação dos espaços de características antes do alinhamento das distribuições de dados para disponibilizar melhores soluções, e no uso de dados rotulados e não rotulados para aprendizado de características. Além dos métodos de transferência de aprendizado, como *fine-tuning* e *manifold alignment* com uso de medidas clássicas de avaliação de performance de reconhecimento, uma métrica de generalização entre domínios foi também proposta para avaliar a transferência de aprendizado. Esta tese contribui com: uma análise de múltiplos descritores contidos em redes profundas supervisionadas; uma nova arquitetura com função de perda para redes profundas semi-supervisionadas (*Weighted Label Loss*), em que todos os dados disponíveis, rotulados ou não, são incorporados para prover aprendizado; e uma nova medida de generalização (*Cross-domain Feature Space Generalization Measure*) que pode ser aplicada para qualquer modelo e sistema de avaliação.

Palavras-chave: transferência de aprendizado de características; aprendizado profundo; medidas de generalização; cruzamento de domínios; alinhamento de variedades.

ABSTRACT

SANTOS, F. P. **Features transfer learning between domains for image and video recognition tasks**. 2020. 99 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Feature transfer learning aims to reuse knowledge previously acquired in some source dataset to apply it in another target data and/or task. A requirement for the transfer of knowledge is the quality of feature spaces obtained, in which deep learning methods are widely applied since those provide discriminative and general descriptors. In this context, the main questions include: what to transfer — align the data distribution from source and target, and adjusting the parameters to increase the model's generalization capability; how to transfer — investigating methods that work on the features spaces or also on the learned models; and when to transfer — studying which datasets are mode adequate for transferring, considering discrepancies between source and target data, such as they different acquisition settings, clutter and illumination variation, among others. This thesis advocates that the focus should be in transferring feature spaces, learned by convolutional neural networks, in particular investigating the descriptive potential of inner and initial layers of such deep convolutional networks, and the approximation of feature spaces before aligning the data distribution in order to allow for better solutions, as well as the use of both labeled and unlabeled for feature learning. Besides the transfer learning methods, such as fine-tuning and manifold alignment, with use of classical evaluation metrics for recognition performance, a generalization metric between domains is also proposed to evaluate transfer learning. This thesis contributes with: an analysis of multiple descriptors contained in supervised deep networks; a new architecture with a loss function for semi-supervised deep networks (Weighted Label Loss), in which all available data, labeled or unlabeled, are incorporated to provide learning; and a new generalization metric (Cross-domain Feature Space Generalization Measure) that can be applied to any model and evaluation system.

Keywords: features transfer learning; deep learning; generalization measures; cross-domain; manifold alignment.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

G_{comp}	Complete Cross-domain Feature Space Generalization
G_{part}	Partial Cross-domain Feature Space Generalization
Adam	Adaptive Moment Estimation
AE	AutoEncoder
AUC	Area Under the Curve
CDFG	Cross-Domain Feature Space Generalization Measure
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Networks
EER	Equal Error Rate
MMDE	Maximum Mean Discrepancy Embedding
MSE	Mean Square Error
PCA	Principal Component Analysis
RBF	Radial Basis Function
ReLU	Rectified Linear Function
RKHS	Reproducing Kernel Hilbert Space
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SLT	Statistical Learning Theory
SN	SmallNet
SRN	SmallResNet
SVM	Support Vector Machine
TCA	Transfer Component Analysis
TL	Transfer Learning
WLL	Weighted Label Loss

CONTENTS

CHAPTER 1

INTRODUCTION

In recent years, machine learning has collaborated in computer vision area with mechanisms that offer high performances for pattern recognition tasks. For these cases, a learning task involves absorbing instances, each example that constitutes a database, and adjusting their descriptors to provide a feature space, a set of attributes that describes each instance, which should be descriptive enough for all provided data. Such learning approaches rely mainly in availability of data for which training sets, used to infer the models, and target sets, e.g. the test set or any unseen data, are assumed to be represented by the same feature space or share the same data distribution. However, this assumption is not always true in real-world applications. In this sense, the study of how such models generalize is one of the main issues. When the target set differs from the training set, one may need to consider to fully reconstruct the original model from scratch, retraining it with new data. This approach can be expensive and sometimes impossible (LU *et al.*, 2015), in particular when considering the high human cost to collect and annotate large databases (SHAO; ZHU; LI, 2015). In this scenario, the possibility of reusing similar and large datasets would guarantee the reduction in effort to recollect new data (PAN; YANG *et al.*, 2010). For this purpose, there is an immense incentive to leverage previous learning in order to obtain Transfer Learning (TL), which can be defined as follows:

> “Transfer learning and domain adaptation refer to the situation where what has been learned in one setting can be exploited to improve generalization in another setting” (GOODFELLOW; BENGIO; COURVILLE, 2016).

Consequently, TL takes advantage of concepts already learned, for example as a classifier or detector, and apply those to facilitate the search of parameters for new classifiers or detectors (LU *et al.*, 2015). The premise is that some source domain or task[1] can provide useful knowledge for some target domain or task (YOSINSKI *et al.*, 2014). If the source and the target

[1] See definition of domain and task in Chapter 2 (State-of-the-art Context).

datasets are sufficiently similar, it is expected that the model has acceptable performance between those datasets (TZENG *et al.*, 2015). Since this is not always the case, the main challenge in TL is to correlate the source training data distribution to the target test data distribution (HU; LU; TAN, 2015). Based on this definition, TL should be analyzed in three perspectives: what to transfer by investigating the similarity between domains in which common peculiarities must be highlighted and discrepancies must be minimized, for example using supervised and/or unsupervised models, learned features, and parameters; how to transfer the knowledge, such as exploring machine learning techniques and pattern recognition; and when to transfer the knowledge detecting scenarios where transfer is useful and avoiding negative transfer (PAN; YANG *et al.*, 2010), that occurs when the acquired knowledge worsens the model performance (TORREY; SHAVLIK, 2010). In this thesis, we addressed some of these questions by studying scenarios of Inductive TL and Transductive TL. **Inductive TL** occurs when there are differences in the target task when compared to the one already learned so that it needs labeled examples from the target domain in order to adapt the knowledge. On the other hand, **Transductive TL** does not require target domain information during the transfer task, only the availability of source domain labels (RIBANI; MARENGONI, 2019).

The meaning of TL changes accordingly with the assigned task: in classification tasks one desires that a classifier, trained with some existing data, is sufficiently representative to allow distinguishing coexisting labels in both domains, or even to adapt itself in order to predict new labels which do not exist in the source (see Figure 1). Considering the task of anomaly detection, TL should be used to enhance the similarity between normal instances and abnormal instances, so the main objective is to learn a general concept of normality, as illustrated in Figure 2. In such tasks, although the aim and meaning of the methods is different, there is a common underlying task: to make sure the feature space of source and target data are compatible. This can be seen as a **transfer of the learned features**, in which both representations (source and target) are transformed to emphasize the similarity to the detriment of discrepancies.

A remarkable application of TL of feature spaces is in computer vision, in particular image classification and video recognition tasks. In both scenarios, TL has the potential to be applied to various problems, such as for example traffic control (WANG *et al.*, 2017), facial attribute classification (ZHUANG *et al.*, 2018), video classification (WU *et al.*, 2015), and anomaly detection in surveillance videos (XU *et al.*, 2017). As it is the current standard in computer vision (PONTI *et al.*, 2017), all the aforementioned studies applied Deep Learning (DL) as a tool (see more details in section 2.1), exploring TL via architecture retraining or feature extraction. Because of the hierarchical structure of Deep Neural Networks (DNN), such methods are able to represent both low-level (shapes, borders, and colors) and high-level (texture and semantics) visual features (YOSINSKI *et al.*, 2014; PONTI *et al.*, 2017). In Convolutional Neural Networks (CNNs), different processing layers can be incorporated, where convolutional, dense, and pooling are the most relevant ones. These layers, depending on the purpose of the task, are set so that the output of one layer is the input of the next one. When the parameters

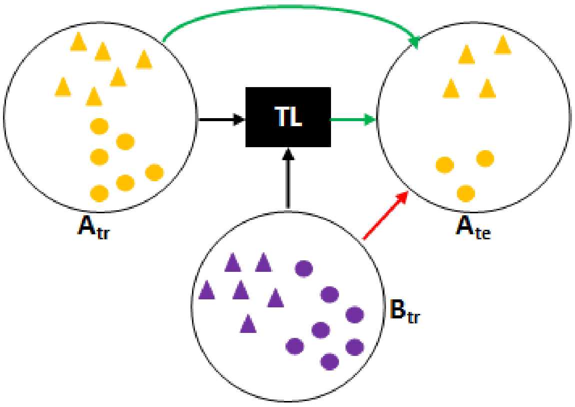

Figure 1 – Expected scenario in feature TL for classification tasks. Green arrows represent correct classification and red arrow identifies incorrect pattern recognition from two distinct datasets (A and B). However, when applying some TL method between them, it is expected that classification will be improved. Samples on (tr) represent training sets and samples on (te) indicate the test set. When applying TL, B_{tr} is influenced by information contained in A_{tr}, approaching the data distributions. Although the two training domains have the same classes (triangles and circles), they may have discrepancies due to irregularities in acquisition and other particularities.

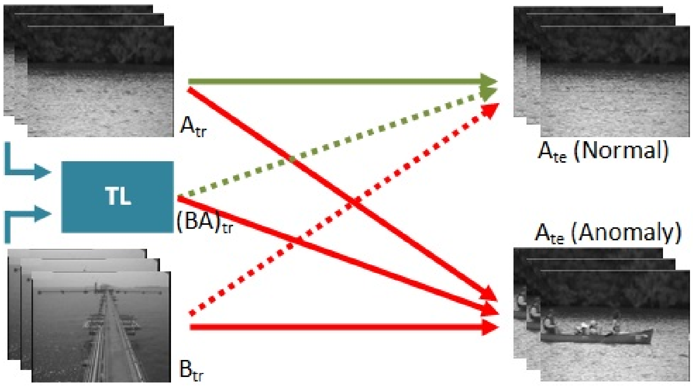

Figure 2 – Expected scenario in feature TL for anomaly detection tasks. Green arrows represent normal events detection and red arrows identify anomalous occurrences from two distinct datasets (A and B). However, when applying some TL method between them, it is expected that normal events detection will be improved (dashed flows). Frames on the left represent training sets (tr) and right frames indicate the test set (te). When applying TL, B is influenced by information contained in A, approaching the data distributions. From (SANTOS; RIBEIRO; PONTI, 2019).

of such models are learned, it is possible to map some input data, e.g. image or video, into a subspace that represents this instance. We call this a feature embedding. In the case of DNNs, multiple feature embeddings are available at the different network layers.

The optimization of the model is guided via a loss function that measures the efficiency of the current parameter settings. Examples of such functions are the cross-entropy and Mean Square Error (MSE). These loss functions can be applied distinctly considering different paradigms of supervision. In supervised learning, loss functions are applied to verify if the predictive model has learned to distinguish different labels for all instances used in its training. The most likely label in the last layer is then compared to the label provided in the input for all examples. Consequently, supervised learning requires that all data be labeled (PONTI *et al.*, 2017). In contrast, unsupervised learning does not require any label. In this structure, the network processes the image provided in the input and verifies its ability to reconstruct the data, comparing the output with its respective input (PONTI *et al.*, 2017). Semi-supervised learning are models that consider partially labeled data, where we can leverage all existing instances and make use of correct labeling measurement and data reconstruction. Consequently, hybrid networks can be used to absorb both paradigms of learning with different purposes (SANTOS *et al.*, 2020).

In this context, two main approaches are commonly employed, both use CNNs that are trained in a source domain, often referred to as pre-trained networks: feature extraction; and fine-tuning (KORNBLITH; SHLENS; LE, 2019). The first consists in performing feature extraction using the pre-trained network by obtaining the outputs of some layer of the network, forming a feature space for the target task, as we show later (SANTOS; PONTI, 2018; SANTOS; RIBEIRO; PONTI, 2019). The second approach is to train the network by initializing its parameters with those from a pre-trained network (instead of random initialization), then re-designing some layers according to the target task (the last layer, responsible for prediction is often redefined as a new layer depending on the classes of the target problem), or even training it from scratch, as we demonstrated in (SANTOS *et al.*, 2020). During fine-tuning, specific properties from the target domain are incorporated into the network, promoting adaptability from one domain to another and increasing the learning (KORNBLITH; SHLENS; LE, 2019). This has the advantage of allowing the use of state-of-the-art CNNs that can be tailored to the target task. In such cases, it is possible to obtain a combination of different feature embedding, e.g. coming from different layers, in order to improve feature TL, as we show in (SANTOS; PONTI, 2019a).

One of the main advantages presented by feature learning methods in relation to hand-crafted extraction is the feature space generalization. In terms of supervised learning, generalization is a divergence measure of how a classifier or detector performance with unseen data (test) is consistent with its performance on seen data (training) (MELLO; PONTI, 2018). In several scenarios, CNN shows good generalization capacity for unseen data within the same visual domain (SHI *et al.*, 2018). In a comparison among few models, Sengupta and Friston (2018) demonstrate empirically that studies must be concerned not only with performance metrics, but also with the generalization capacity of one solution, proposing an investigation of highly detailed stability. Evidently, the degree of similarity between data distribution determines the degree of semantics maintained. However, a generalization measure for the purpose of consistently evaluate the level of knowledge transferred between two data distributions is still to be investigated. In this regard, we are concerned with “when to transfer”.

This study investigates how to manage previously acquired knowledge and how to evaluate its generalization in image and video recognition tasks. Two distinct approaches are explored, although they may be used in the same framework: network fine-tuning (see section 2.2 for more details); and manifold alignment (described in section 2.3). When using supervised pre-trained networks, it is common to study latter layers (BAHETI; GAJRE; TALBAR, 2018; SADIGH;

SEN, 2018) as potential feature extractors. Due to the scarcity of in-depth studies with initial and inner layers, this research aimed at a more concise analysis of the descriptive capacity contained in these layers, applying feature extraction and fine-tuning (SANTOS; PONTI, 2018) and exploring feature spaces fused from multiple layers for adaptation with manifold alignment methods (SANTOS; PONTI, 2019a). In addition, in order to leverage unlabeled data, we also investigated a novel semi-supervised network, showing that such model can provide discriminative embedding for several domains in different proportions of labeled examples (SANTOS *et al.*, 2020). Also, we propose a new generalization metric. Although many studies involving images or videos employed feature TL techniques, they are not evaluated in their generalization capacity. Because of this, our generalization measure may be used to distinguish which datasets and models provide the highest learning transfer rate for a selected domain (SANTOS; RIBEIRO; PONTI, 2019). Such proposed method is used for evaluating transfer of learning in video anomaly detection (SANTOS; RIBEIRO; PONTI, 2019) and semi-supervised image classification (SANTOS *et al.*, 2020). Consequently, we studied what to transfer using pre-trained model parameters and feature spaces learned in deep networks. Additionally, we investigated how to use the knowledge acquired in different paradigms of learning by identifying when to apply the transfer through generalization metrics.

1.1 Motivation and Objectives

Many studies propose specific solutions to a given dataset, in which the model offers suitable feature spaces and high accuracies (or other measures related to the task). Meantime, if another dataset (same domain) was applied to this model, the performance may be lower, not being fully adaptable to this new dataset. In this case, changes in the model are required to improve the performance. Consequently, this approach is adequate only for restrictive scenarios or particular cases. For generic scenarios, the contradiction between models and similar domains must be investigated more deeply to provide one single solution for the same task. It would be unproductive to have specific models for each dataset of same domain, making it almost impossible to implement systems for the purpose of solving real-world problems. For those cases, the generalization of solutions is paramount, in which the number of models proposed should be smaller, although more robust. One of the alternatives is to apply feature embedding adaptation by TL techniques, in which feature spaces become more correlated among involved domains, increasing the feature generalization.

As listed earlier, to implement generalizable models we need to be aware of what to transfer and how to transfer (PAN; YANG *et al.*, 2010). Additionally, one of reasons for a model not to be completely adaptable to several domains is the absence of generalization measures evaluation. In order to guarantee all these requirements, our models were evaluated with classical and generalizable measures in different scenarios, trying to provide the best feature spaces and making the model more embracing for distinct datasets. Hence, this research aimed at investigating feature TL in terms of deep network representations, considering visual attributes, to find methods which generate feature embedding that generalize for unseen data. In particular, the following topics for transferring representation learning were pursuit:

- generate discriminative, compact, and generalized feature spaces for different domains;
- investigate network fine-tuning as well as manifold alignment methods and their impact on obtaining better representations for target datasets;
- integrate supervised and unsupervised architectures to improve feature space generalization;
- propose and evaluate the architectures via generalization metrics and divergences.

These topics were thoroughly explored in investigations involving initial and inner layers from CNNs, semi-supervised learning, and evaluation of generalization capacity with anomaly detection.

1.2 Hypotheses

Based on the wide variety of possible applications for feature TL and the existing gaps, this research aims to investigate several points related to this theme. Our premise is that, although it requires a lot of processing and memory space to provide high performance solutions (MELLO; FERREIRA; PONTI, 2017), the CNN is a consolidated deep network architecture, providing different feature spaces transformations via non-linear functions. The general hypothesis relates multiple levels of representations (layers) in deep networks for different paradigms of learning (supervised, semi-supervised, and unsupervised) in which classic and generalization metrics are applied to assess the learning ability of predictive models. Consequently, the following general hypothesis is:

“Deep networks for feature transfer learning tasks should be properly analyzed at different hierarchical levels of representations and paradigms of learning, considering both classical and generalization measures.”

This hypothesis is divided into three statements specific to each investigated point. First, there is sufficient empirical evidence to say that the layers in the beginning of the network provide low-level features while last layers contain semantic context (YOSINSKI *et al.*, 2014). However, it is unclear which of the different layers provide the best discriminative capability, and whether those are complementary and if so, how those can be combined. Therefore, the first specific hypothesis is that:

(i) Different inner layers of supervised deep networks should be considered, and potentially combined, when obtaining feature spaces in order to improve image and video recognition in transfer learning scenarios.

Despite the high performance of TL methods and DL architectures, generally, studies have focused only on classical metrics of classification or anomaly detection to measure the performances. Due to this convention, the models have refrained to prove their generalization capability with respect to other datasets. Consequently, the applicability of these models has remained specific for some datasets. However, concern about domain generalization has current evidence (SENGUPTA; FRISTON, 2018). In this context, the basic foundation of TL is to be able to leverage only aspects that provide increased of performance, avoiding negative transfer (LU *et al.*, 2015). In these cases, the need to investigate which dataset can provide better previous knowledge, or even indicate which model is most suitable for a task, is essential. Based on these requirements, the second specific hypothesis states that:

(ii) The descriptive capacity of a model to transfer the acquired learning should be measured by metrics of each task and by different levels of divergence.

Due to the cost and difficulty of annotating data from different domains (SHAO; ZHU; LI, 2015), some databases are found partially labeled. In these scenarios, we may consider a supervised model that leverages only data with labels or incorporate all examples into an unsupervised prediction. However, semi-supervised models can be developed to absorb all data in its natural form (REN *et al.*, 2019). Considering that both unlabeled and labeled instances contain important information for the learning of a predictive model, and consequently impacting its ability to describe the feature representation, our third specific hypothesis states that:

(iii) In partially labeled data transfer learning scenarios, labeled and unlabeled examples should be used jointly to increase the performance of the feature space.

Through extensive experiments, these statements are highlighted over the next chapters, using several domains, networks, tasks, and paradigms of learning.

1.3 Outline

The following chapters of this document include: State-of-the-art Context; Features Transfer Learning using Multiple CNN Layers; Feature Transfer Learning in One-class Scenarios with a Generalization Analysis; Feature Transfer Learning in Semi-supervised Settings; and Conclusions. Aiming at a broader view of the literature, **State-of-the-art Context** (Chapter 2) presents the concepts that involve features TL, such as deep learning, network fine-tuning, and manifold alignment, as well as a more in-depth context on the research subject. The models and techniques developed during this research to confirm the hypotheses are described in **Features Transfer Learning using Multiple CNN Layers** (Chapter 3), **Feature Transfer Learning in One-class Scenarios with a Generalization Analysis** (Chapter 4), and **Feature Transfer Learning in Semi-supervised Settings** (Chapter 5). Finally, **Conclusions** (Chapter 6) report the contributions, publications, future directions, and final considerations.

CHAPTER

2

STATE-OF-THE-ART CONTEXT

Domain discrepancy is the biggest obstacle to existing predictive models (LONG *et al.*, 2015). A domain $D = \{X, P(X)\}$ consists of a feature space $X = \{x_1, ..., x_n\}$ and a probabilistic distribution $P(X)$. A task $T = \{Y, f(.)\}$ consists of a label space $Y = \{y_1, .., y_m\}$ and a prediction function $f(.)$, which models $P(y|x)$ for all $y \in Y$ and $x \in X$ (XIE *et al.*, 2016). Therefore, given a source domain D_s and a learning task T_s, a target domain D_t and a learning task T_t, TL aims to improve the function learning of the target prediction in D_t using the knowledge in D_s and T_s, where $D_s \neq D_t$ or $T_s \neq T_t$ (PAN; YANG *et al.*, 2010). Specifically, feature TL is a particular case where the dissimilarity between data distributions from the source feature space in relation to the target feature space is minimized, regardless of the feature extraction method used (handcrafted or DL) (SHAO; ZHU; LI, 2015; PAN; YANG *et al.*, 2010). Consequently, data distribution must be generated by smooth functions, e.g without appearance of large discrepancies. Furthermore, the learning process must have high generalization capacity, in which the input-output mapping of the model should be equivalent for unseen data (test sets) during the training phase (HAYKIN, 2001; SANTOS; RIBEIRO; PONTI, 2019).

Defining appropriate features to categorize a set of images or videos is a costly task due to analyze which attributes are relevant. Additionally, in view of the variation of poses, illumination, or shadows, the process of recognizing an object may not be generalized (SANTOS; RIBEIRO; PONTI, 2019). As a result of these drawbacks, the development of efficient algorithms to reduce classification or detection costs from different domains are encouraged. Because of that, feature mapping preceded by feature learning has shown to be a pertinent subject with good results. In addition to greater accuracy (BENGIO; COURVILLE; VINCENT, 2013), feature learning has the generalization property for different applications. All these happen because of their ability to adapt to new tasks and domains, requiring less human intervention in the process (GOODFELLOW; BENGIO; COURVILLE, 2016). Into feature embedding adaptation context, many DL techniques present themselves with high performance and flexibility, ensuring a wide range of applications.

2.1 Deep Learning

According to Deng, Yu *et al.* (2014), DL can be defined as a machine learning technique that exploits many layers of non-linear information processing. This technique aims the extraction and transformation of supervised or unsupervised features for pattern recognition and other tasks. Therefore, deep neural networks are considered intelligent feature extraction modules that offer great flexibility and high levels of cross-domain TL (LU *et al.*, 2015). In these architectures, the number of layers is defined by the need to represent feature levels, where each successive layer uses the output data from previous one as input. Hence, higher-level features are derived from lower-level features in a hierarchical representation form defined by simpler relationships, allowing the computer to learn through experiments (GOODFELLOW; BENGIO; COURVILLE, 2016). A fundamental requirement to guarantee that deep networks are able to learn concepts of one domain is the amount of data (RAVISHANKAR *et al.*, 2016), which can be very expensive (DUAN *et al.*, 2012). Consequently, one domain rarely have representation enough to learn its own concepts, and the higher the number of parameters in hidden layers, the greater the amount of examples required (MELLO; FERREIRA; PONTI, 2017).

Among many layers contained in a deep neural network, three types stand out with great relevance: convolutional; pooling; and dense. Composed by a filters set of fixed size, convolutional layers generate a new space representation to the next layer applying an affine transformation of linear combination in all pixels concentrated inside of filter neighborhood (SANTOS; PONTI, 2018). Each filter produces a new feature map, increasing quickly the amount of parameters accumulated during successive layers. Consequently, pooling layers operate dimensionality reduction, mainly considering the maximum value of the defined patch (GUO *et al.*, 2016). Dense layers aim to vectorize feature maps, converting data to classes probabilities contained in the training dataset for supervised networks (PONTI *et al.*, 2017). Often, the activation function Rectified Linear Function (ReLU) is applied in hidden layers to cancel all negative values and maintaining all positive values linearly ($max[0,x]$) (NAIR; HINTON, 2010). Additionally, normalization is employed as the last step in each hidden layer to preserve consistency (common issue in image processing, whose RGB values must range in $[0,1]$).

Mainly, the deep neural networks differentiate among themselves due to the loss function in their last layer. Convolutional Neural Network (CNN) relates its performance in the comparison between the input label and the output label for each sample. Therefore, in this learning context, CNNs are supervised networks (PONTI *et al.*, 2017). Through those hierarchical structure, CNNs have shown to be effective descriptors for low-level (shapes and edges) and high-level (textures and semantics) features, presumably due to high abstraction capacity codified in their many layers (RAZAVIAN *et al.*, 2014). Furthermore, low error rates using raw RGB images enable bypass pre-processing steps from standard pipelines in several applications (MISHKIN; SERGIEVSKIY; MATAS, 2017; SANTOS; PONTI, 2018). Therefore, as the image proceeds through the first layers, the built-in feature map adds both shape, border, and color information.

Since all CNN models incorporate the same concept, the aspect of these layers refers almost exclusively to Gabor filters and color blobs (YOSINSKI *et al.*, 2014). This important property allows networks pre-trained in large datasets from different domains to be used as feature descriptors for small target domain. Accordingly, the hierarchical function $f_l(.)$ related to some layer l considers a parameters set W_l for an input image x_1. Hence, an input x will produce an output $f(x)$, as described:

$$f(x) = f_L(...f_2(f_1(x_1, W_1), W_2), ...W_L) \tag{2.1}$$

As an example, considering an hypothetical CNN model as illustrated on Figure 3, an input image provides three maps (RGB) into the first layer. In each convolutional layer, the depth of each filter is equivalent to the output from the previous layer. For each filter, an activation function f is placed to generate a single resulting map. Consequently, the amount of maps is equal to the amount of filters. In pooling layers, each region (represented by a same color) will result in a single corresponding value as output. This occurrence is for each map coming from previous convolutional layer. For the first dense layer, also called as Fully Connected Layer, there is only an input vectorization. However, the last one provides probabilities of each class. In this illustration, five categories were considered and, for an image containing a kangaroo, the prediction indicates the class A as more representative.

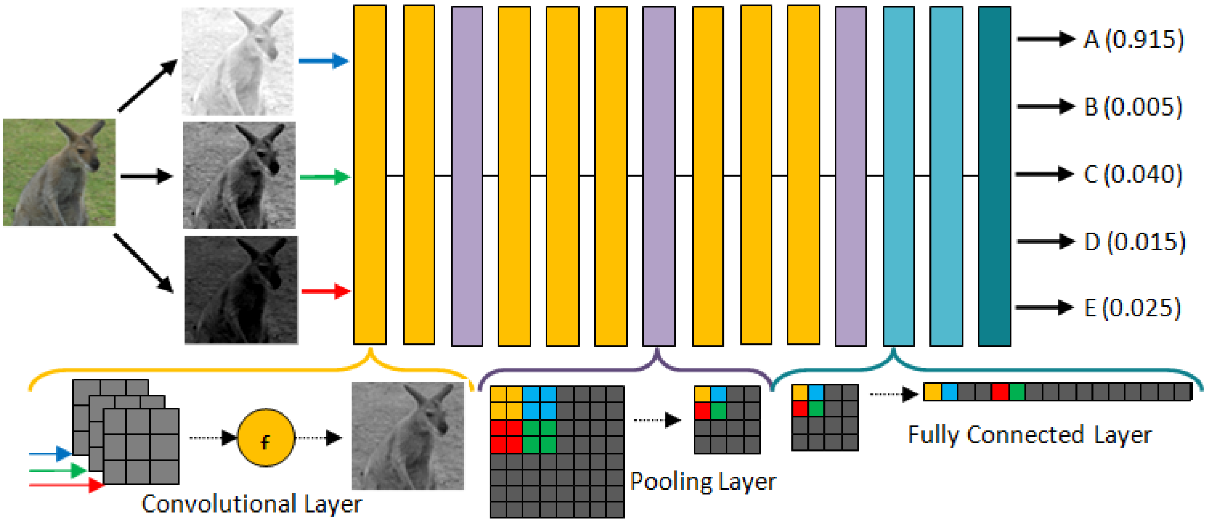

Figure 3 – Example of a CNN architecture containing convolutional (orange), pooling (purple), and dense (green) layers. On the left, an input image is provided to the first layer by its three RGB channels. To the right, resulting probabilities of each class. Below, details of each internal procedure of the three types of layers.

CNNs also have a high degree of invariance in translation, scaling, and other forms of distortions. After processing a layer, the exact location of one feature is not relevant, as long as its relative position to the rest was preserved. In addition, each layer generates distinct feature maps, but respective weights are shared between them, providing displacement invariance and parameters reduction (HAYKIN, 2001). However, small variations, as color quantization or noisy perturbations, can lead to a significant decrease in testing error of feature space generality (NAZARE *et al.*, 2017; SANTOS; PONTI, 2018). Derived from the training accuracy, some models are able to memorise the dataset, resulting in overtraining (PONTI *et al.*, 2017). Also, CNNs suffer from overparametrization and high parameters correlation (BAGHERINEZHAD; RASTEGARI; FARHADI, 2017)[1]. Several CNN architectures have been developed in recent years, with VGG-19 (SIMONYAN; ZISSERMAN, 2014), ResNet50 (HE *et al.*, 2016), and MobileNet (HOWARD *et al.*, 2017) being the most consolidated in this field.

VGG-19 was developed applying 19 weight layers, in which most of filters in convolutional layers are 3×3 size. The almost exclusive use of this shape is based on the concept that two consecutive 3×3 filters have an effective receptive field equivalent to one 5×5 filter, and three 3×3 filters can be used as one 7×7 filter. Additionally, this structure reflects in fewer parameters, even increasing the amount of layers to supply the filter size reduction. Filters 1×1 are applied only to perform linear projection of a position across all feature maps from one layer. After an intercalated sequence of convolutional layers with pooling (maximum value), the top of VGG is composed of three consecutive dense layers, being the last one for probabilities of trained classes (SIMONYAN; ZISSERMAN, 2014).

ResNet50 applies the residual blocks concept to train a CNN with a greater number of layers. Using sequential regular convolution layers, residual blocks aim to preserve features from the input vector before its transformation, adding to the output after some layers of delimited block. Another interesting property of ResNet50 is the absence of dense layers: a pooling is put after the last convolution layer to compute predictions in output layer (HE *et al.*, 2016).

MobileNet is a compact CNN which uses the concept of depthwise separable convolutions. A standard convolutional layer joins inputs and filters into an output set in a single step. However, depthwise convolution maintains the data separated, one layer for filtering (depthwise convolution) and another one for combining them (pointwise convolution). A sequence of N regular convolutional layers of dimensions $D \times D \times M$ are replaced by M depthwise layers of $D \times D \times 1$ and N pointwise layers of $1 \times 1 \times M$, where D is the height and width of the input and M is the amount of maps. Hence, pointwise convolution performs linear combination among filters applied to input (single filter per channel). This factorization allows a model size reduction and less computation cost (HOWARD *et al.*, 2017).

In contrast to supervised learning, unsupervised networks, such as AutoEncoder (AE), aim to learn an approximate identity function between the data contained in the input and output layers, which parameters in hidden layers are representations of intrinsic properties from these images (GUO *et al.*, 2016). Accordingly, an image x provided as input will result in a new image $\hat{x}$, as similar as possible to x. The idea is not to generate an identical image, but learning parameters to represent the same class. The internal structure is composed by *encoder* and *decoder* modules, where each one can have several layers (convolutional, transpose convolu-

[1] More details in section 2.2 (Network Fine-tuning).

tional, pooling, unpooling, and normalization among others) with its own function (BENGIO; COURVILLE; VINCENT, 2013). An image passed as input x is processed by the *encoder* and generates a constrained representation of the data, called *code*. Sequentially, this representation is reconstructed by the *decoder*, $\hat{x}$. Because the *code* is limited to data representation, AEs avoid performing a perfect copy of input image (PONTI *et al.*, 2017). To this purpose, some obstacles are put to hamper the construction, such as dimensionality reduction, as shown in Figure 4, noise injection, or regularization term.

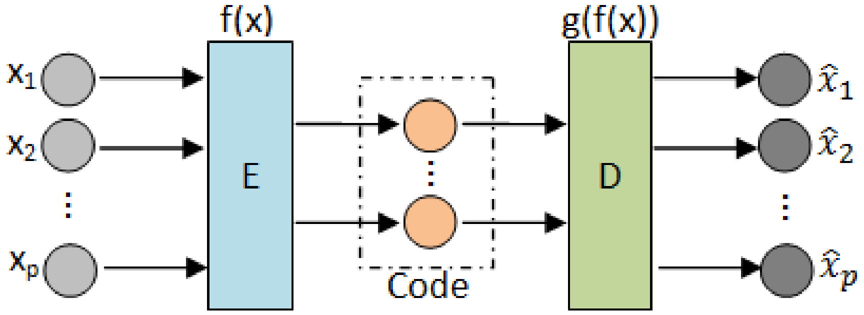

Figure 4 – Generic architecture of AEs, being a feed-forward unsupervised network. An image x composed of p pixels is passed to the encoder E. In this module, a function $f(x)$ operates the image coding in a new representation. In the decoder D, a function g reconstructs the “code” into a new image $\hat{x}$. The code size varies according to the architecture specification.

In this architecture, encoder maps the input x applying a non-linear activation function Φ in a weight matrix W and a bias vector b: $f(x) = \Phi(Wx + b_e)$. Using a similar function, decoder reconstructs the code by just changing the input: $g(f(x)) = \Phi(Wf(x) + b_d)$ (PATHIRAGE *et al.*, 2018). Inside the encoder, the latent representation h of a layer k is described by $h^k = \sigma(x * W^k + b^k)$, where the bias b is shared across the whole feature map, σ is an activation function, and $*$ denotes the convolution between the input x and weights matrix W. Weights are shared across all locations, preserving local aspects provided by images (MASCI *et al.*, 2011). Consequently, the reconstruction $\hat{x}$ is performed by linear combinations of several image patches generated in the code:

$$\hat{x} = \Phi\left(\sum_{k \in H} h^k * \tilde{W}^k + b\right) \tag{2.2}$$

Both structures (CNNs and AEs) can be trained from scratch, fine-tuned after a previous trained, or provide feature spaces to be improved using manifold alignment methods. The network fine-tuning approach is based on the generalization capability that DL models offer to other domains. Considering a pre-trained deep network with a large dataset, these models can be modified by incorporating the knowledge acquired and suppressing a possible lack of representativeness from new domains. The second approach, manifold alignment aims to provide a new feature space by aligning data distributions in a new latent space.

2.2 Network Fine-tuning

One of techniques addressed on this research to achieve feature embedding adaptation was network fine-tuning. It consists of reusing weights from pre-trained deep networks with large datasets, e.g ImageNet (RUSSAKOVSKY *et al.*, 2015), and refining the solution by retraining layers with the dataset of current task domain (PONTI *et al.*, 2017). Despite the great potential, several difficulties are directly related to its practical application: image resolution; overtraining, overfitting, and overparametrization; and amount of examples required.

First, architectures have a fixed resolution due to a predefined input. Even by varying the input size, the architecture is not fully adaptable to some domains, which may have different shapes. Therefore, a pre-processing step must be applied to reduce or increase resolutions, impacting on loss of information or noisy addition. Second, a major concern is defining accurately when to stop the training. Overtraining provides to the network a memorization of data distribution, resulting in poor performance with test examples and avoids generalization to other similar domains (HAYKIN, 2001), called of overfitting. Due to current complexity networks, they provide an enormous amount of parameters (overparametrization) in which computational costs (hardware and process time) are excessive (BAGHERINEZHAD; RASTEGARI; FARHADI, 2017). Despite the high number of descriptors, many of them have the same activation value in all samples, indicating absence of representativeness. Therefore, eliminating attributes without variance implies in reducing costs of processing in the predictive model. In this context, more compact models, e.g. MobileNet (HOWARD *et al.*, 2017), may offer adaptable solutions for scenarios which require lower processing. Third, to train a deep network is necessary plenty of examples from all classes belonging to the domain. The training set size is defined by three aspects: sample representativeness; depth of architecture network; and complexity of the task. However, the architecture is fixed and the task difficulty is not measurable (HAYKIN, 2001). Hence, to ensure representativity of the domain is required a large set of instances (SRIVASTAVA *et al.*, 2014). In this scenario, if a dataset has few examples for network training, data augmentation is an option that has excellent benefits for increasing representativeness, including variations of original data with noise injection, image composition, and rotations.

All these aspects should be considered to fine-tune efficiently a model. Generally, CNNs are pre-trained with ImageNet dataset (RUSSAKOVSKY *et al.*, 2015) which is composed of 1000 classes (see Figure 5-a). Accordingly, the last layer provides a vector with 1000 probabilities. However, other datasets can be used as source domain. Hence, to operate fine-tuning is necessary to adapt the prediction layer to have outputs equivalent to the number of classes n contained in the task domain (Figure 5-b). In this context, the model top undergoes changes, not only in the last layer, but in some previous ones. The number and which layers to change depend on the model structure and the purpose of the task (Figure 5-c). Therefore, to train an architecture is required to choose the method: frozen layers; or propagation. Accordingly, the first layers provide low-level features, regardless of the domain used. Hence, the frozen approach maintains weights of the already trained layers unchanged. In this situation, only a few layers are refined without changing the initial structure of the network. The propagation is similar to frozen layers, except that weights of the already training layers can be influenced by the new domain. In addition to the training mode, it is necessary to define weights initialization of each layer in the new architecture: original ones; or randomly. Initialization with original weights is only possible on unmodified layers. The random approach is theoretically applied across all layers, been more common in new layers (YOSINSKI *et al.*, 2014).

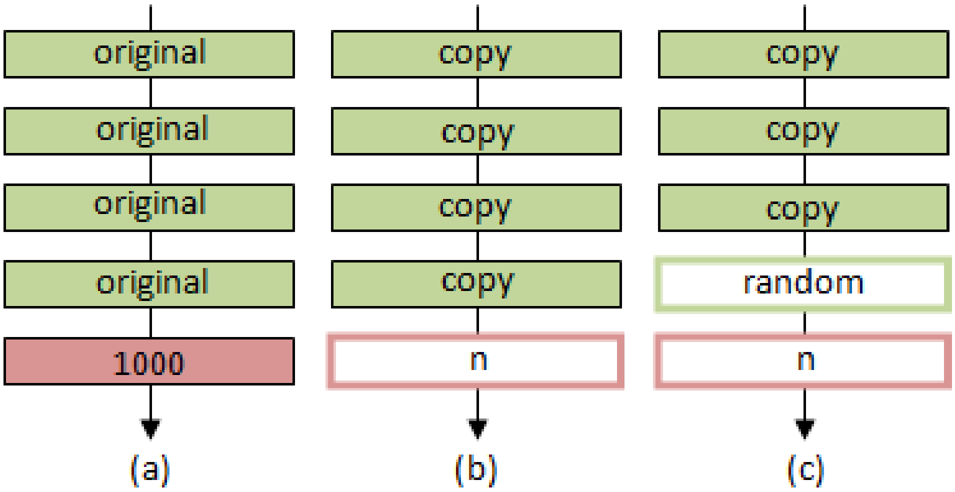

Figure 5 – Fine-tuning: (a) an example of network model to be used in a fine-tuning task, where the red block is the prediction layer and green blocks are regular layers; (b) only the prediction layer is modified, copying all regular ones; and (c) in addition to the prediction layer, the second last layer was also modified or copied without respective weights. All layers copied integrally can be frozen or are prone to be influenced by the new training.

In the example illustrated in Figure 5, it was presented fine-tuning applied only to supervised network (CNN). However, fine-tuning can also be applied to unsupervised and semi-supervised networks. Considering AEs, additional layers can be added next to the code generation to further refine the solution or even remove some layers in an attempt to generalize the model to more domains. In the semi-supervised learning, the maintained layers from CNN can be transformed into an encoder and by adding extra layers, without prior training, to form the decoder in a hybrid model (SANTOS *et al.*, 2020).

To measure the progress of supervised network training, considering instances provided so far, a loss function in the prediction layer is applied. This function will express the penalty for predicting a label $\hat{y}$ in which should be y. Cross-entropy loss $l^{(ce)}$ is often applied, minimizing the estimation between class probabilities e^{f_y} and output probabilities e^{f_k} of a sample j (PONTI *et al.*, 2017):

$$l_j^{(ce)} = -\log\left(\frac{e^{f_{y_j}}}{\sum_k e^{f_k}}\right) \tag{2.3}$$

With two probability vectors containing values in $[0,1]$, the cross-entropy loss function can be seen as a divergence measure between two distributions. However, the full loss function L (with finite amount of samples N) is the average of all inputs x_j, given the current set of all parameters W and label vectors y_j (PONTI *et al.*, 2017):

$$L(W) = \frac{1}{N}\sum_{j=1}^{N} l(y_j, f(x_j;W)) + \lambda \cdot \sum_{k}\sum_{l} W_{k,l}^2 \tag{2.4}$$

The sum regularized the loss function to undesired solutions (e.g. many similar W) and does not hamper the discovery of good parameters in training. However, the training of an AE is measured by the reconstruction error, defined by $\varepsilon = x - \hat{x}$. In case of the activation function was linear, the reconstruction error will have only a single local and global minimum (BALDI; HORNIK, 1989), being described by Principal Component Analysis (PCA) (JOLLIFFE, 1986).

To adjust parameters and minimize the loss function an optimization algorithm is applied, among them Stochastic Gradient Descent (SGD). By random selection samples of size B, SGD operates approximations to calculate new parameters, which η is the learning rate. Commonly, η is initialized by high values and it goes exponentially decreasing (weight decay) during iterations. Consequently, large values of η will indicate small time to convergence. However, the ideal point may be located between intervals, requiring that the value be reduced so that convergence being possible (PONTI *et al.*, 2017), which $\bigtriangledown$ indicates the gradient:

$$W_{t+1} = W_t - \eta \sum_{j=1}^{B} \bigtriangledown L(W; x_j^B) \tag{2.5}$$

Adaptive Moment Estimation (Adam) is also a widely optimization algorithm applied to minimize the loss function. Differently from SGD, Adam verifies which parameters are less frequent, assigning more weight to them. Another important concept in deep networks training is the batch size, which defines the amount of instances loaded in memory. Evidently, it is not possible to load all examples at once, hence it is feasible to indicate blocks that are compatible with the task or the processing capacity. Some studies suggest that batch size should be as large as possible, occupying all memory (GOYAL *et al.*, 2017). Other studies contradict this hypothesis, showing that small batches allow greater precision in minimizing loss (LI *et al.*, 2014). Being even more rigid, a batch size with 32 examples should be the ideal, independent of the task (MASTERS; LUSCHI, 2018). After to pass all instances through the network, an epoch is completed. This whole process must be repeated a number of epochs until the network converges, measured by the applied loss function.

2.3 Manifold Alignment

Manifold alignment methods present a framework to strengthen relationships from different feature spaces into a new unified latent space by aligning underlying manifolds. In this new joint feature space, the similarity among domains is emphasized and attributes that represent distancing are softened (KOUW; LOOG, 2019). Hence, dimensionality reduction operates a vital role in eliminating attributes that produce folds misalignment. However, only space reduction is insufficient, the development of an organized arrangement of feature maps is also required to highlight the alignment that will provide the highest performance (WANG; KRAFFT; MAHADEVAN, 2011). Therefore, considering two datasets $X = \{x_1, x_2, .., x_n\}$ and $Y = \{y_1, y_2, .., y_m\}$, with n and m being the amount of examples, functions f and g provide the alignment to map X and Y into a same feature space Z (WANG; MAHADEVAN, 2009). In this scenario, each domain (dataset) represents a manifold in which properties, such as the neighborhood relationship and categorization of instances, must be preserved in the latent space. Consequently, instances with same label remain mapped in close locations, distancing them from different classes, as shown in Figure 6.

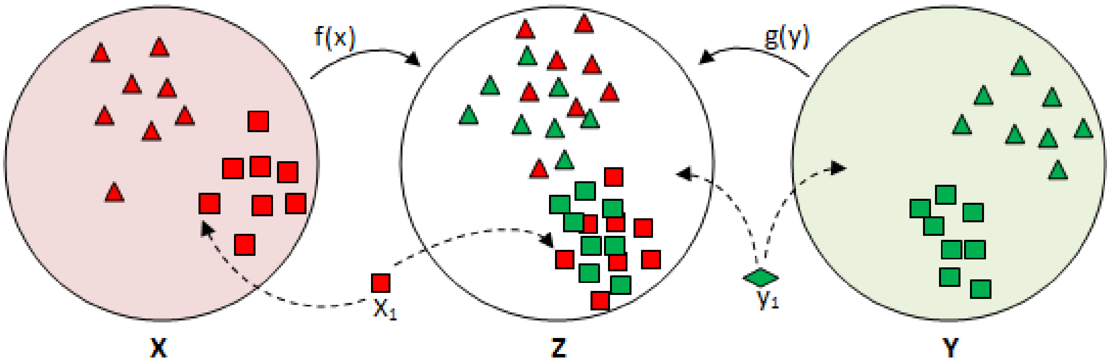

Figure 6 – Manifold Alignment. The latent space Z is generated by functions f and g that map features from X and Y, finding similarity between them. An instance x_1 is correctly mapped in both X and Z spaces. An unknown instance y_1 must be labeled in both Y and Z spaces.

Learning in manifold alignment methods can occur in two circumstances: semi-supervised or unsupervised. The first one is rarer to apply due to the required knowledge of correspondence pairs of samples (CUI *et al.*, 2014). In this approach, the unified manifold is created and then mapped to a latent space of lower dimensionality, where the local properties of each dataset were preserved (WANG; MAHADEVAN, 2009). Accordingly, this scenario makes it difficult to generalize solutions, even in almost identical tasks because different domains have different correspondence pairs. Contrarily, in the unsupervised approach this knowledge is not required and learning occurs directly from the manifold structure (CUI *et al.*, 2014).

One of the widely techniques used in unsupervised manifold alignment is Transfer Component Analysis (TCA) (PAN *et al.*, 2011). This technique attempts to learn a common set of underlying transferable components from two domains, source and target, in which the dissimilarity in data distributions must be reduced, maintaining the properties preserved in a latent subspace projection. Considering that $P(Y_s|X_s)$ and $P(Y_t|X_t)$ are two probability distributions that shape domains X and Y from a source s and a target t, there is a transformation Φ which $P(\Phi(X_s)) \approx P(\Phi(X_t))$. Therefore, TCA proposes to find the Φ using two pre-definitions: (i) the distance between the generated distributions $P(\Phi(X_s))$ and $P(\Phi(X_t))$ is relatively small; and (ii) the transformation Φ preserves important properties of X_s and X_t. With these pre-definitions, Φ ensures $P(Y_s|\Phi(X_s)) \approx P(Y_t|\Phi(X_t))$:

$$D(X'_s, X'_t) = \left|\left| \frac{1}{n} \sum_{i=1}^{n} \Phi(x_{s_i}) - \frac{1}{m} \sum_{i=1}^{m} \Phi(x_{t_i}) \right|\right|_H^2 \tag{2.6}$$

Operating in Reproducing Kernel Hilbert Space (RKHS) $||.||_H$, the distance between feature spaces is the norm between averages of instances in each distribution. Because it is highly nonlinear, Φ is generally not optimized directly due to not achieving the optimum local minimum. As a result, Maximum Mean Discrepancy Embedding (MMDE) (PAN; KWOK; YANG, 2008) is applied to embed both domains (source and target) into a low-dimensional shared feature space $\tilde{K}$. To achieve this space, in the kernel K the distance of domains ($K_{s,t}$ and $K_{t,s}$) is minimized and the data variance ($K_{s,s}$ and $K_{t,t}$) is maximized, being $K_{s,s} = X_s X_s^T$:

$$K = \begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix} \in \Re^{(n+m).(n+m)} \tag{2.7}$$

By definition D can be rewritten in terms of kernel matrices as $D = tr(KL)$ (PAN; KWOK; YANG, 2008), in which tr is the matrix trace (sum of elements on the main diagonal from the upper left to the lower right):

$$L_{i,j} = \begin{cases} \frac{1}{n^2} : \{x_i, x_j\} \in X_s \\ \frac{1}{m^2} : \{x_i, x_j\} \in X_t \\ -\frac{1}{n.m} : otherwise \end{cases} \tag{2.8}$$

However, dimensionality reduction has not yet been applied. Using empirical kernel map (SCHÖLKOPF; SMOLA; MÜLLER, 1998), K can be decomposed into $(KK^{-1/2})(K^{-1/2}K)$. Consequently, if $\tilde{W} \in \Re^{(n+m)\ d}$, which $d \ll n+m$, reduces the dimensionality of the space to d, the feature space becomes:

$$\tilde{K} = (KK^{-1/2}\tilde{W})(\tilde{W}^T K^{-1/2} K) \tag{2.9}$$

Therefore, $\tilde{K} = KWW^TK$ considering $W = K^{-1/2}\tilde{W}$. Hence, replacing K by $\tilde{K}$, the distance between X'_s and X'_t in lower dimensional space is:

$$D(X'_s, X'_t) = tr((KWW^TK)L) \tag{2.10}$$

To preserve important properties of each distribution, TCA adds a regularization term to the final distance in the decomposition of eigenvalues, where the intensity is controlled by a parameter μ (PAN *et al.*, 2011):

$$D(X_s', X_t') = tr((KWW^T K)L) + \mu \, tr(W^T W) \tag{2.11}$$

In this new transformed space, a classifier/detector is trained in $\Phi(X_s)$ and applied in the target feature space $\Phi(X_t)$ for predictions. To perform a viable analysis of the method, it is required only training sets to generate the latent space. As a result, the matrix $\tilde{W}$ is then used to reduce the test set space. Due to its theoretical basis being derived from the PCA, for the feature space transformation to be performed it is necessary to define the amount of attributes desired for the output (SANTOS; RIBEIRO; PONTI, 2019).

CHAPTER

3

FEATURE TRANSFER LEARNING USING MULTIPLE CNN LAYERS

Prior to feature learning methods for pattern recognition, the standard pipeline consisted of filtering, segmentation, feature extraction, and classification (XIE *et al.*, 2017). Some of these steps were absorbed into the deep networks (MISHKIN; SERGIEVSKIY; MATAS, 2017) due to the incorporation of successive convolutions in the grid-shaped neural architectures. This absorption allowed the predictive models to be composed of different descriptors, including both low and high level, making end-layers into receptive fields of previous layers (RAZAVIAN *et al.*, 2014).

Usually, models that involve feature extraction of CNN for pattern recognition only use layers that are very close to prediction output. This behavior can be observed in several studies, such as for skin lesions images (POMPONIU; NEJATI; CHEUNG, 2016; MAHBOD; ECKER; ELLINGER, 2017; MAJTNER; YILDIRIM-YAYILGAN; HARDEBERG, 2016). However, does this adopted convention indicate that previous (initial and inner) layers do not offer good discriminative capacity? In situations where dissimilarity between the source and target domain is evident, the semantic information contained in end-layers should be avoided or minimized (YOSINSKI *et al.*, 2014). Furthermore, the threshold among features level is still uncertain, changing for each domain and requiring a crucial study to determine the best layer to provide representativity for each problem (YOSINSKI *et al.*, 2014). Hence, we should not assume that only the pre-prediction layers provide representativeness for classification. Contrarily, as the initial and inner layers offer a low-level description of shapes, borders, and colors, they may play an important role in the task. Also, when feature extraction is performed on deep networks, the feature maps obtained can be transformed using dimensionality reduction (SANTOS; PONTI, 2018; LIN; ROYCHOWDHURY; MAJI, 2015), concatenation (SADIGH; SEN, 2018; PAN *et al.*, 2017), or even alignment of the data distributions (WEN *et al.*, 2018; SANTOS; RIBEIRO; PONTI, 2019) for TL tasks.

Due to the gap described here, we investigated whether these layers (initial and inner) on CNNs actually provide representativeness in TL tasks for image domains. Hence, in this chapter, the methods and techniques developed for feature TL using inner layers are presented. Initially, CNNs were applied to provide feature spaces analysis of several end-layers, as detailed in section 3.1. After, multi-layers feature spaces from one same CNN are fused and adapted by TCA, as described in section 3.2.

3.1 End-layers features on skin lesion classification

Initially, considering only CNNs pre-trained with ImageNet dataset (RUSSAKOVSKY *et al.*, 2015), it was desirable to develop a structure to compare the abstraction of several end-layers to evaluate the discriminative capacity, aiming to discuss feature generality and the potential for TL. Intending to further analyze these layers, a detailed methodology was developed to discuss distinct feature spaces provided from three different CNNs and their respective hidden layers: MobileNet (HOWARD *et al.*, 2017); VGG-19 (SIMONYAN; ZISSERMAN, 2014); and ResNet50 (HE *et al.*, 2016)). Moreover, it was also investigated impacts caused by distortions applied to the best feature space obtained, such as dimensionality reduction by PCA (JOLLIFFE, 1986), color quantization, and noise injection. The behavior of the feature spaces was also performed applying cross-domain and network fine-tuning. This methodology structure is illustrated in Figure 7.

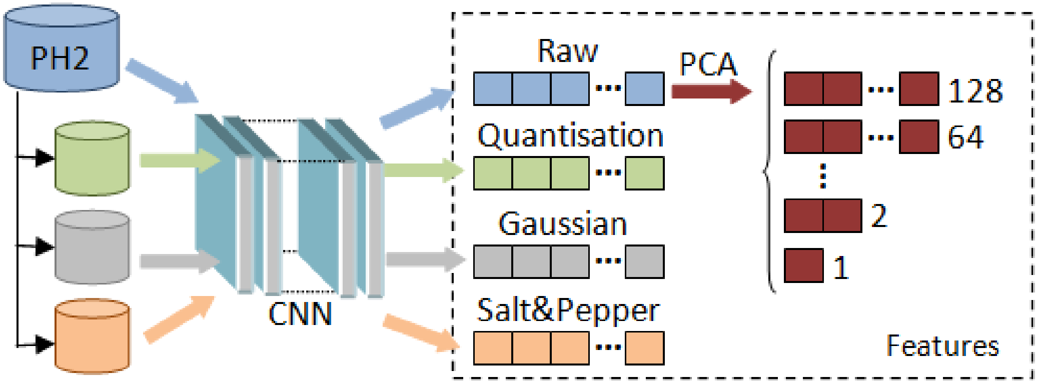

Figure 7 – Feature Extraction. From raw images using PH2 dataset, new sets were generated by color quantization and noise injection (Gaussian and Salt & Pepper). Feature maps, extracted using a pre-trained CNN, were used to evaluate skin lesion classification and generalization capacity between raw and different levels of distortions. PCA was applied to raw set to measure space reduction efficiency. From (SANTOS; PONTI, 2018).

The PH2 dataset (MENDONÇA *et al.*, 2013) was employed for these experiments, which is widely used as a benchmark for skin lesion classification. PH2 is composed of 200 dermoscopic images of two main categories: malignant (40 melanomas) and non-malignant (80 common nevi and 80 dysplastic nevi), each class is represented in Figure 8. By the lesion appearance, the categories differ in shapes, edges, colors, and texture. The adoption of this domain is due to its classification challenge, where the categorization of an image is made difficult by many aspects: images are acquired under different conditions of illumination; may have blur due to the focal field; and also present strong texture due to the appearance of skin and other confusing components. Consequently, it is expected that the analyzed behavior can be generalized to simpler domains.

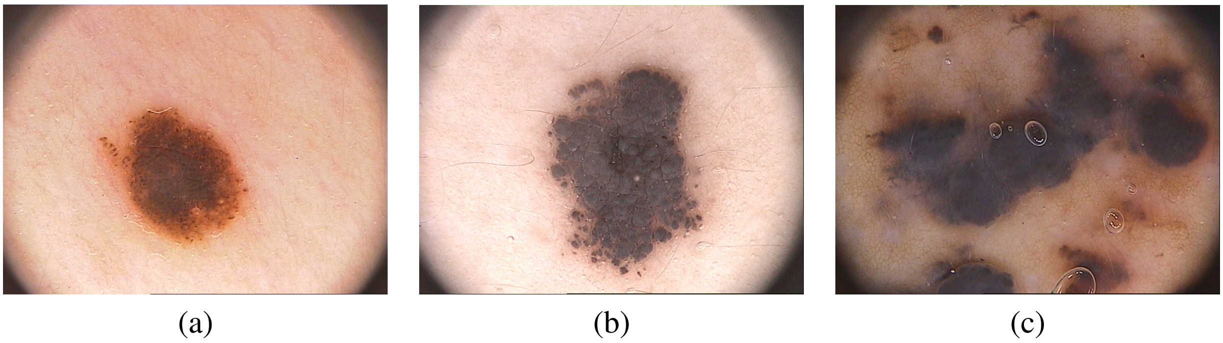

Figure 8 – Skin lesions from PH2 dataset (MENDONÇA *et al.*, 2013): (a) Common Nevus; (b) Displasic Nevus; and (c) Melanoma. The preliminary diagnosis of skin cancer includes visual analysis of low-level features, being that common nevi have more regular structures than melanomas. Additionally, the presence of confusing objects is evident in these examples in the image composition, such as the black circle on margins and some bubbles superimposed on the lesion (see image c), increasing the challenge of finding an adequate feature space.

First, images were re-sized to 224 × 224 pixels of resolution (architectures restriction for inputs) and feature spaces were extracted using activation values from the last seven layers in each CNN. Commonly, the last layer of each CNN corresponds to probabilities of 1000 classes from ImageNet (RUSSAKOVSKY *et al.*, 2015). Each CNN has different structures and, consequently, amount of attributes in each layer. Therefore, for MobileNet, layers output from 1000 to 50176 features, and for VGG-19 and ResNet50 from 1000 to 100352 features. After performing a standard scale normalization, the balanced accuracy was computed using 20-folds cross validation (each fold is class-balanced). Due to high dimensionality provided from CNN layers, the real amount of features with variance was identified, removing all attributes which have equal value assigned in all examples. These attributes only increase computational cost (BAGHERINEZHAD; RASTEGARI; FARHADI, 2017). For evaluation, it was applied Support Vector Machine (SVM). Intuitively, a more adequate feature map performs better with SVM because it is the simplest classifier and it has stronger learning guarantees (VAPNIK, 1999). The next subsection presents the results and discussion of robustness of feature spaces from CNN end-layers.

3.1.1 Robustness of feature spaces

Based on results shown in Table 1, the best performance was achieved in earlier layers: MobileNet layer -3 (94%); VGG-19 layer -7 (91.5%); and ResNet50 layer -5 (91.5%)[1]. Con-

[1] Layers are refer as -1 (last layer), then -2 (one before the last), and so on until -7. In bold, the best result from each network. This notation is also valid for Table 2.

trarily, prediction layers (-1) achieved significantly poorer results. MobileNet also provided more compact and discriminative feature spaces than VGG-19 and ResNet50 for skin lesions classification (using only 1024 features). These advantages are evidenced by the amount of attributes with variance in each layer and respective high performance. In contrast, VGG-19 generates more attributes without variance (in all hidden layers), having its performance surpassed also by ResNet50 (on layers average). The best performance achieved in ResNet50 (layer -5) demonstrates the space complexity: 100352 features. These performances corroborates the concept that smaller datasets for specific applications do not need complex networks. Hence, MobileNet (the lighter CNN) provides the best performance in accuracy, complexity, and space dimensionality.

Table 1 – CNNs pre-trained with ImageNet: 20-folds Cross Validation by Balanced Accuracy (%). Adapted from (SANTOS; PONTI, 2018)

CNN	Layer	Features	Variance	Linear SVM
MobileNet	-1	1000	100.0%	85.0 ± 12.04
	-2	1000	100.0%	92.0 ± 8.72
	-3	1024	100.0%	**94.0** ± **6.63**
	-4	1024	100.0%	93.5 ± 7.26
	-5	1024	100.0%	93.0 ± 8.43
	-6	50176	90.2%	90.5 ± 8.65
	-7	50176	100.0%	91.5 ± 7.26
VGG-19	-1	1000	100.0%	81.0 ± 12.61
	-2	4096	93.7%	88.5 ± 6.54
	-3	4096	93.7%	88.5 ± 8.53
	-4	25088	86.8%	89.0 ± 6.24
	-5	25088	86.8%	88.5 ± 7.26
	-6	100352	75.2%	91.5 ± 7.92
	-7	100352	92.8%	**91.5** ± **6.54**
ResNet50	-1	1000	100.0%	80.5 ± 11.17
	-2	2048	100.0%	90.0 ± 7.75
	-3	2048	100.0%	90.5 ± 7.4
	-4	100352	96.3%	91.5 ± 7.92
	-5	100352	100.0%	**91.5** ± **7.26**
	-6	100352	100.0%	90.5 ± 7.4
	-7	100352	100.0%	90.5 ± 9.73

To confirm high performance achieved using MobileNet pre-trained with ImageNet, a new experiment with fine-tuning was performed employing the HAM10000 dataset (TSCHANDL; ROSENDAHL; KITTLER, 2018) as source domain. HAM10000 is composed of approximately 10000 images split into seven classes (training set), two of them contained in PH2. Consequently, we used a dataset from the same domain, however, with its own particularities. For this experiment, all prediction layers were replaced by new ones to classify these seven classes. Network fine-tuning experiments were carried out with 10, 25, 50, 100, and 500 epochs using 32 images in batch size and Adam algorithm with Binary Cross-Entropy as loss function. Layers contained after the last pooling (four layers in VGG-19 and two in ResNet50) were randomly initialized, except for MobileNet where it was selected only the last two layers, which are related to the classifier. The remaining ones were maintained with their original ImageNet weights. However, all last seven layers can adapt their weights, considering all previous ones frozen. The best results in each adaptation are presented in Table 2, which MobileNet overcomes other approaches with 91.5% from layer -3. Although similar performance among fine-tuned CNNs, results shown to be slightly below when comparing to the ones without fine-tuning. It is important to note that, both VGG-19 and ResNet50 need more epochs to obtain best performances due to greater architecture complexity. As well as MobileNet, ResNet50 quickly converged to perfect training accuracy (50 epochs). However, VGG-19 remains unchanged even after 500 epochs, saturating the loss and not converging. This issue implies that the amount of examples from HAM10000 did not offer a relevant gradient in fine-tuning to improve classification loss. Intuitively, MobileNet is smaller allowing convergence, while ResNet50 convergences due to skipping layers on residual blocks. Contrarily, VGG-19 is so deep for this amount of data, not converging properly. Therefore, all further analysis were carried out on MobileNet layer -3 feature space with ImageNet weights.

Table 2 – Best results from fine-tuning with HAM10000 training dataset. Adapted from (SANTOS; PONTI, 2018)

CNN	Training Loss (%)	Training Acc. (%)	Epochs	Layer	Test Acc. (%)
MobileNet	0.7	100.0	10	-3	**91.5 ± 9.1**
VGG-19	147.7	90.7	50	-4	91.0 ± 8.89
ResNet50	0.0	100.0	50	-5	90.0 ± 7.75

Since CNN layers often output high-dimensional feature maps, dimensionality reduction is an important projection to show attributes relevance in the final classification. The space was gradually reduced from 128 features to only 1 feature by PCA, halving the size each step. PCA imposes as rule for amount of components selection a minimum between samples (200 skin lesions) and features (1024 in MobileNet layer -3). Therefore, as seen in Figure 9, Linear SVM continues achieving high performance between 64 and 16 features ($\approx$ 92% of balanced accuracy). Furthermore, Linear SVM tends to show better performance overall, implying that reducing dimensionality does not affect space linear separability. To complement the dimensionality reduction discussion, also in Figure 9, the variance is presented for the amount of features, showing that a 60.80% variance allows class separability (LSVM $\approx$ 92% using 16 features). As expected, PCA variance decreases gradually as the space contraction increases. We can also note that with just 1 feature is possible to correctly categorize 86.5% of the examples.

Feature spaces quality are significantly impacted by color quantization (PONTI; NAZARÉ; THUMÉ, 2016). To measure this influence in skin lesions classification, news sets were generated by computing 64, 32, and 16 colors per channel. As demonstrated in Table 3, as the color space contracted, performances become less linear, although not dramatically lower. Considering noisy images, Gaussian sets were generated by variances of 0.008, 0.016, and 0.032, which the balanced accuracy of Linear SVM remains relatively constant with the progressive increase of noise. However, with Salt & Pepper (probabilities were 0.005, 0.01, and 0.02) is detect an initially positive impact (SP 0.005), but then results degrade.

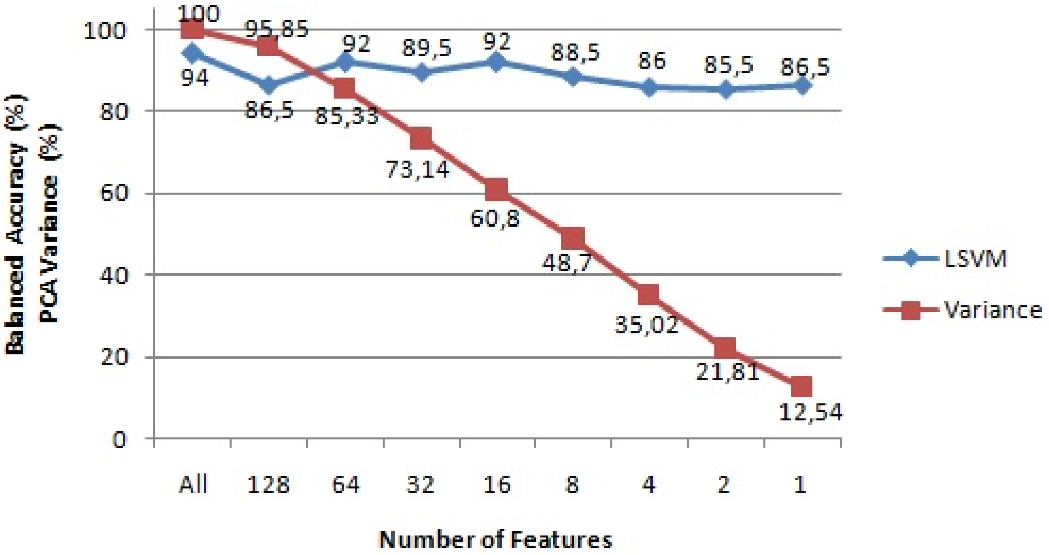

Figure 9 – Dimensionality reduction and variance by PCA using MobileNet layer -3 feature space without fine-tuning. Curves of the classifier have a lack of smoothing due to the small size of the dataset. These dimensions were selected to show a gradual feature space reduction. Adapted from (SANTOS; PONTI, 2018).

Table 3 – Quantized and noisy space of MobileNet layer -3 without fine-tuning: 20-folds Cross Validation by Balanced Accuracy (%). Adapted from (SANTOS; PONTI, 2018)

Set	Linear SVM
PH2 Quant 64	94.5 ± 4.97
PH2 Quant 32	92.5 ± 8.29
PH2 Quant 16	90.0 ± 9.49
PH2 G 0.008	93.0 ± 7.81
PH2 G 0.016	93.0 ± 6.4
PH2 G 0.032	94.5 ± 7.4
PH2 SP 0.005	95.0 ± 6.71
PH2 SP 0.01	91.5 ± 9.1
PH2 SP 0.02	90.5 ± 8.65

To study feature spaces generalization more deeply, Hold-out 50/50 experiments (balanced classes) were performed using different versions of PH2 dataset to measure how well features generalized for unseen color quantization and noisy levels. Results are divided in two parts: without and with fine-tuning. Both were performed using MobileNet layer -3 (better feature space). As expect for both color quantization and noisy addition, the generalization rate reduces according the bigger distances among color spaces and increasing noisy levels. However, additive noise causes a positive impact: the lowest average reached among Gaussian sets was 88.66% (for G 0.016); similarly for Salt & Pepper with a highest average of 90.83% (for SP

Table 4 – Feature space generalization using MobileNet layer -3: Hold-out 50/50 by Balanced Accuracy (%). Adapted from (SANTOS; PONTI, 2018).

Training Set	Testing Set	Without fine-tuning	With fine-tuning	Difference
Raw	Quant 64	**91.5**	83.0	8.5
	Quant 32	90.0	84.0	6.0
	Quant 16	86.5	84.5	2.0
Quant 64	Raw	**91.0**	90.0	1.0
	Quant 32	**90.5**	84.5	6.0
	Quant 16	87.0	83.0	4.0
Quant 32	Raw	90.0	**90.5**	-0.5
	Quant 64	**91.0**	85.0	6.0
	Quant 16	87.0	84.0	3.0
Quant 16	Raw	**91.0**	89.5	1.5
	Quant 64	**91.5**	85.5	6.0
	Quant 32	**91.5**	86.5	5.0
Raw	G 0.008	**90.5**	83.0	7.5
	G 0.016	89.0	82.0	7.0
	G 0.032	88.5	82.5	6.0
G 0.008	Raw	90.0	85.5	4.5
	G 0.016	89.0	86.5	2.5
	G 0.032	88.0	85.0	3.0
G 0.016	Raw	89.5	85.5	4.0
	G 0.008	88.5	85.5	3.0
	G 0.032	88.0	84.0	4.0
G 0.032	Raw	90.0	87.5	2.5
	G 0.008	90.0	86.0	4.0
	G 0.016	89.0	85.0	4.0
Raw	SP 0.005	89.5	82.0	7.5
	SP 0.01	88.5	80.5	8.0
	SP 0.02	88.5	79.5	9.0
SP 0.005	Raw	**92.5**	85.0	7.5
	SP 0.01	89.0	83.5	5.5
	SP 0.02	**91.0**	84.0	7.0
SP 0.01	Raw	88.0	83.0	5.0
	SP 0.005	88.0	82.0	6.0
	SP 0.02	89.5	82.5	7.0
SP 0.02	Raw	89.0	79.5	9.5
	SP 0.005	89.5	80.0	9.5
	SP 0.01	89.0	79.5	9.5

0.005). Noisy results indicated positive perturbations on data which CNN models produce a more robust space, improving the classifier to find linearly separable hyper-planes for unseen noisy/quantization levels. However, feature generalization from HAM10000 fine-tuned is lower in comparison to ImageNet training parameters, being worse, mainly, with noise injection. This peculiarity occurs due to the different data distributions between PH2 and HAM10000. Despite belonging to the same domain, these datasets have differences in the number of classes, image resolution, and diversity of confused objects. These factors, in addition to the small training set for fine-tuning, contribute to the superior performance of feature extraction with ImageNet training. All results are shown in Table 4, where accuracies in bold are higher than competing methods presented in Table 5.

Table 5 – Competing methods results versus MobileNet layer -3. Adapted from (SANTOS; PONTI, 2018).

Method	Accuracy (%)	Balanced Accuracy (%)
(BARATA; CELEBI; MARQUES, 2015)	—	84.3
(BI *et al.*, 2016)	92.0	90.31
(SALIDO; JR, 2018)	93.0	—
With Fine-tuning	84.0	91.5
Hold-out [Raw, SP 0.005]	89.5	92.5
Without Fine-tuning	95.0	94.0
SP 0.005	94.0	95.0

Comparing all the experiments with competing methods in PH2 dataset, CNN feature extraction with MobileNet produces the highest result (94% of balanced accuracy) using only raw images. ResNet50 and VGG-19 also presented higher performances (91.5%) than competing methods, which comprised pre-processing steps to achieve at most 90.31%.

3.2 Alignment of multi-layers features fusion

In addition to the finding that end-layers provide robust feature spaces, an extra experiment was designed to detect whether the initial layers of a CNN also provide representativeness for image classification. Features fusion is widely applied for image classification (ZHENG *et al.*, 2019; YU *et al.*, 2017) and it can be performed using distinct approaches, such as applying a single CNN end-layer as global descriptor and handcrafted methods to describe low-level features (CHEN *et al.*, 2018b) or combine end-layers from different CNNs (GE *et al.*, 2017). However, considering the pre-trained ResNet50 (HE *et al.*, 2016) and two domains (source and target), we extracted features from the pre-prediction layer (as global descriptor) and from the three first residual blocks (the output of each block represents the local descriptor) to merge them in a single feature map (as fusion descriptor). Consequently, three scenarios are presented for alignment of multi-layer features fusion: global descriptor with each individually local descriptor. Previously of fusion step, the local features passed on a process of selection (three different methods were performed) due to they are composed by larger amount of attributes. This process is performed separated for the training set of the classifier (source) and for the test set (target). Consequently, with the multi-layer features, the data distributions (source and target) are transformed to increase the correlation between them using TCA (PAN *et al.*, 2011). As result, the source data is applied to SVM for training and the target data for tests, as illustrated on Figure 10.

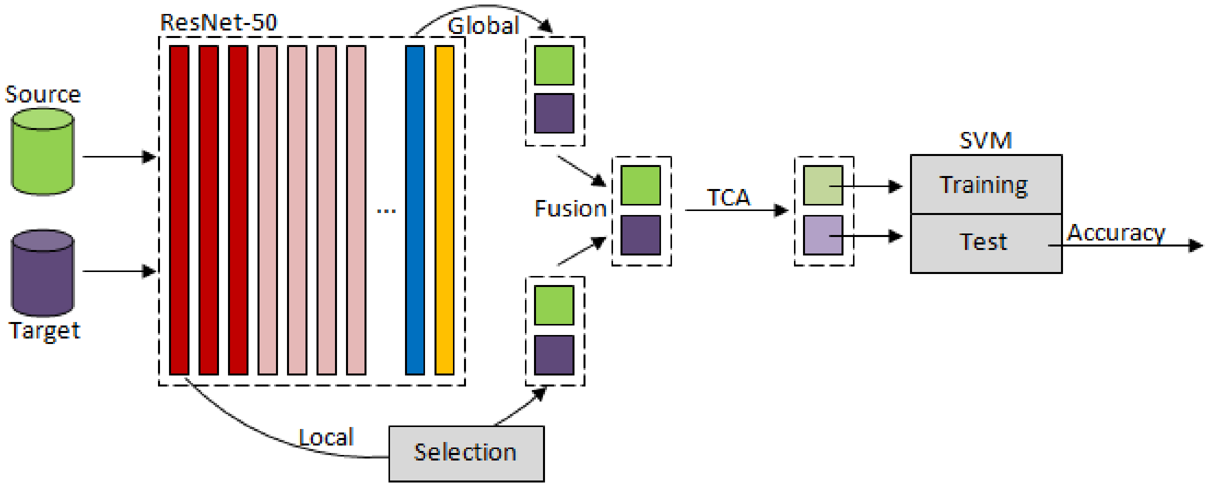

Figure 10 – Feature extraction and manifold alignment structure for multi-layers features. Considering two similar datasets, source and target, both are passed on to ResNet50 for feature extraction. Initially, an initial layer (the red ones) provides local attributes (shape, border, and color) and the pre-prediction layer (the blue one) provides global attributes (texture and semantics). In the following, feature fusion is obtained through map concatenation using both feature spaces. Using TCA, the resulting features fusion are transformed and assigned to train and test the SVM classifier, source and target, respectively. The experiments were performed in two scenarios: (i) using ResNet50 pre-trained with ImageNet; and (ii) performing fine-tuning with the source dataset. Local (low-level) and global (high-level) feature maps are also transformed and classified individually for comparison purposes. From (SANTOS; PONTI, 2019a).

The output from pre-prediction layer (average pooling) has 2048 attributes. The output from the residual blocks has the same shape: 256 maps of 55 × 55 size, resulting in 774400 attributes. Due to the large number of attributes from local descriptors, feature selection was applied to choose which ones will compose the fusion maps. Hence, three methodologies were adopted, detailed on Figure 11, aiming a comparative analysis of performances: PCA; Flatten Pooling; and Pooling 2D. **PCA** is applied only to the source dataset, choosing 256 components. In sequence, the chosen components were applied to the target dataset. However, some datasets do not have 256 examples in the test set. Hence, in these specific cases, it was determined 128 components. In **Flatten Pooling**, the feature maps are fully converted from matrix to vector without any spatial relationship. In the following, a value $x = 100$ was adopted to split the vector into small symmetric segments, in which the average is calculated, forming the 7744 final attributes. For **Pooling 2D** was considered a square region in the attribute space to calculate the average. The adopted region was 55 × 55, where each map provides only one attribute, i.e 256 features. After the feature selection, feature fusion is performed using the 2048 global attributes with one of the local features selection method (256, 7744, or 256). The variation in the number of attributes is suppressed due to TCA transformation with Radial Basis Function (RBF) kernel, which defines the real amount of attributes for classification with Linear SVM. Consequently, the resulting feature maps size are 256, 192, 128, 96, or 64.

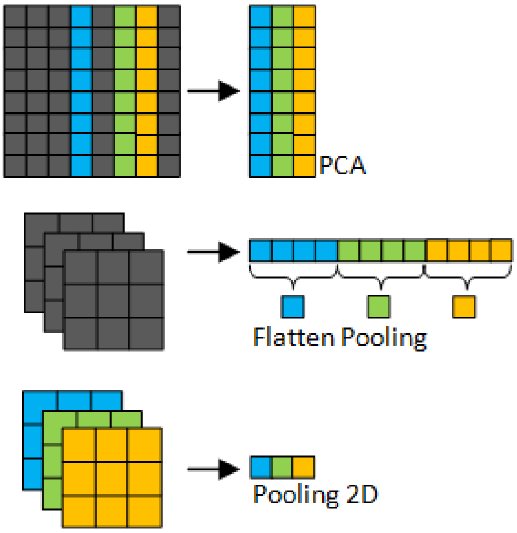

Figure 11 – Feature selection methods for local attributes: PCA considers all images to find the principal components. In contrast, Flatten Pooling and Pooling 2D are performed in an individual manner. From (SANTOS; PONTI, 2019a).

For the experiments, the pre-trained and fine-tuned ResNet50 were considered. The fine-tuning setup applied is the original training (SGD with mini-batch size of 256, learning rate of 0.1 with weight decay of 0.0001, and momentum of 0.9 (HE *et al.*, 2016)) during 100 epochs. Aiming to maintain the initial layers frozen, only the last seven layers were allowed to adapt with the new domain. This configuration offers a better observation of the global and fusion performances. For both network weights, four sets of different image domains were tested: fruits; objects; skin lesions; and photos. This diversity is extremely important to emphasize the discriminative capacity of initial layers, due to variation of styles, scene composition, and degree of task difficulty. All datasets are illustrated in the Figures 12-15, in which the first row from each figure is considered as a source dataset and the other one as the target dataset. Also, all images were resized to 224 × 224.

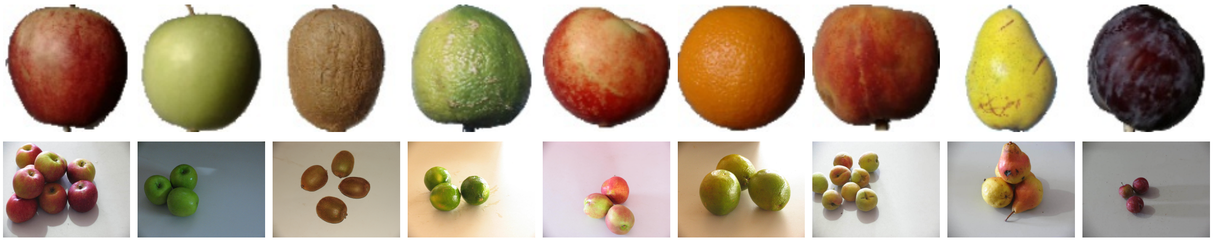

Figure 12 – Examples from: (top) Fruits-360 (MUREŞAN; OLTEAN, 2018); and (bottom) Supermarket Produce (ROCHA *et al.*, 2008). Fruits-360 is composed by, approximately, 53000 images of 100 × 100 pixels resolution (training set), divided in 103 classes. In contrast, Supermarket Produce has only 2000 images of 1024 × 768 pixels of 11 categories. Although both datasets belong to same domain, they differ in the number of elements in each image, background, object size, and illumination. For this setup, only 9 common labels were used, reducing the amount of images from both datasets. From the left to the right: red apple; green apple; kiwi; lime; nectarine; orange; peach; pear; and plum.

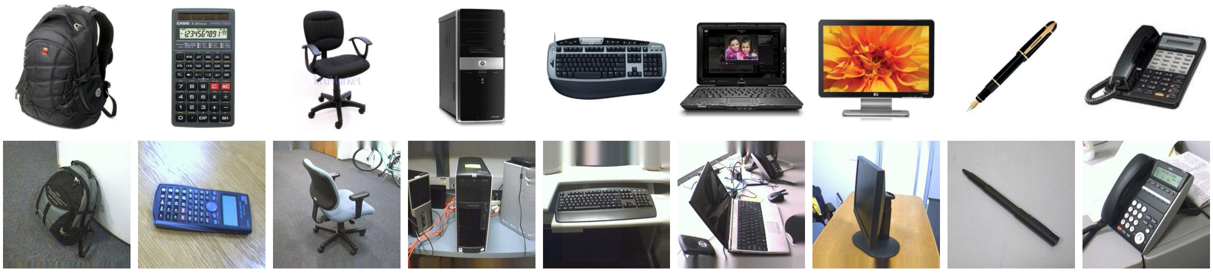

Figure 13 – Examples from: (top) Amazon (SAENKO *et al.*, 2010); and (bottom) Webcam (SAENKO *et al.*, 2010). Amazon has 2817 images of 300 × 300 pixels resolution downloaded from the web into 31 classes. Although Webcam has exactly the same categories, the 795 images vary from one example to another in the resolution. Both sets contain device items, differentiating due to the background, perspective, and presence of clutter. From the left to the right: backpack; calculator; desk chair; desktop computer; keyboard; laptop; monitor; pen; and phone.

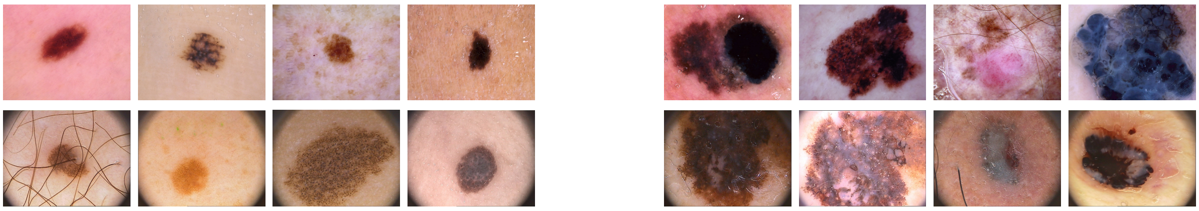

Figure 14 – Examples from: (top) HAM10000 (TSCHANDL; ROSENDAHL; KITTLER, 2018); and (bottom) PH2 (MENDONÇA *et al.*, 2013). HAM10000 contains images of 600 × 450 pixels of 7 distinct classes. PH2 is a smaller dataset, which has only 200 images of 768 × 574 pixels and 2 classes. The datasets differ due to margins composition and presence of clutter, such as hair and bubbles. The malignancy of a lesion is defined by the uniformity of shapes, colors, and texture. Despite the 7 categories from HAM10000, only two common were considered (nevus and melanomas), reducing the number of images in HAM10000 to approximately 7800. The first four images on left indicate common nevus and the others on right represent melanomas.

Figure 15 – Examples from Corel1000 (WANG; LI; WIEDERHOLD, 2001). This dataset is fully balanced, comprising 10 classes of 100 examples. Each image has a resolution of 384 × 256 pixels. For these experiments, the full set was splitted randomly, in proportion of 80/20, for training and test sets. From the left to the right: food; native people; beach; architecture; bus; dinosaur; elephant; flower; horse; and mountain.

3.2.1 Combination of different descriptors

Based on the described datasets, after feature extraction on ResNet50 (trained on ImageNet or fine-tuned), Tables 6-9 show the feature TL results from one source dataset to another target dataset on SVM classifier[2]. Specifically in Table 6 is highlighted that, on average, the pre-prediction layer (global descriptor) offers better feature space when the network does not incorporate the new semantics contained in Fruits-360, i.e using ImageNet weights, with 30.79%

[2] Values in bold (fusion) represent higher accuracy when compared with global results (Glob.). Values in italic (fusion) indicate when the fine-tuning performance overcomes its respective ImageNet result.

versus 27.17% of accuracy. However, multi-layer fusion performances are highly applicable to Supermarket Produce dataset, either with fine-tuning or without. Considering the three multi-layer fusion methodologies on average, the fine-tuned performance increases 8.36% using the first residual block. Individually, PCA has a larger variation in the first block (20.66%), reaching its peak with network fine-tuning (41.46%), then the accuracy gradually decays. Flatten Pooling (Flat.) has a small better performance in the second block. And, Pooling 2D is practically constant in all blocks. All these results indicate that Fruits-360 and Supermarket Produce are datasets with predominantly low-level features, such as shapes and edges, evidenced when the global performance is reduced with network fine-tuning.

Table 6 – Classification accuracy (%) of **Supermarket Produce** comparing feature extraction (FE) from ResNet50 pre-trained with ImageNet versus fine-tuned with **Fruits-360**. From (SANTOS; PONTI, 2019a).

FE	**Feat.**	**Glob.**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**
ImageNet	256	28.24	20.41	**36.48**	26.35	19.96	**37.18**	25.3	19.61	**37.33**	26.35
	192	30.24	20.11	**36.08**	29.64	20.31	**37.33**	29.09	20.46	**37.48**	29.44
	128	34.23	21.61	**38.32**	32.98	21.31	**38.87**	32.19	20.46	**37.77**	32.83
	96	33.63	22.01	**39.42**	32.24	22.36	**40.52**	31.74	21.36	**40.27**	32.39
	64	27.59	19.86	**37.77**	**27.84**	20.51	**39.62**	27.4	19.91	**38.92**	**27.79**
	Avg.	30.79	20.8	**37.61**	29.81	20.89	**38.7**	29.14	20.36	**38.35**	29.76
Fine-tuning	256	23.75	***41.37***	**36.33**	***35.73***	***37.97***	**36.98**	***35.83***	***33.63***	**37.18**	***35.83***
	192	25.65	***41.22***	***36.48***	***36.48***	**38.37**	**36.93**	***36.58***	***30.44***	**36.58**	***36.58***
	128	31.39	***41.47***	***38.87***	***38.82***	***37.48***	***39.27***	***38.77***	***32.58***	**37.43**	***38.77***
	96	29.74	***41.37***	***39.97***	***39.52***	***38.42***	***40.67***	***39.47***	*29.14*	**39.52**	***39.52***
	64	25.3	***41.87***	***37.97***	***39.87***	***38.62***	**39.37**	***39.92***	***31.59***	**38.37**	***39.92***
	Avg.	27.17	***41.46***	***37.92***	***38.08***	***38.17***	**38.64**	***38.11***	***31.48***	**37.82**	***38.12***

In Table 7 is noticed a decrease in the performance of multi-layer fusion features of objects domain in relation to fruits domain. Considering Amazon (as source) and Webcam (as target), fusion results are better at about 1.25% on average when compared with global results: 52.52% versus 51.25% using ImageNet weights and 51.58% versus 50.36% with network fine-tuning. Despite this equivalence, multi-layer fusion features still offers significant improvement using Flatten Pooling in the first block without fine-tuning, on average 55.75% versus 51.25%. PCA has better performance in the second block when fine-tuning is applied and Pooling 2D remains practically constant. These results confirm that Webcam is a dataset with greater variance, requiring more semantics from the global descriptor.

Considering skin lesion images, different texture represents a decisive attribute to diagnose an injury as malignant or not. Hence, the semantic contained in the global descriptor is more relevant for classification. Based on this requirement and evidenced by the results presented in Table 8, the fusion features do not increase the accuracy, neither for ImageNet weights (85.7% versus 85.62%) or for fine-tuning (86.0% versus 85.67%) on average. A few multi-layer fusion

Table 7 – Classification accuracy (%) of **Webcam** comparing feature extraction (FE) from ResNet50 pre-trained with ImageNet versus fine-tuned with **Amazon**. From (SANTOS; PONTI, 2019a).

W	**Feat.**	**Glob.**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**
ImageNet	256	40.63	**42.01**	**47.04**	**42.01**	40.25	**47.17**	**42.14**	**41.64**	**45.79**	**41.89**
	192	48.18	46.67	**52.08**	**48.81**	45.79	**51.95**	**48.81**	46.16	**50.69**	**48.3**
	128	51.95	51.45	**56.86**	**53.08**	48.55	**53.58**	**53.46**	**52.7**	**53.58**	**53.46**
	96	55.35	54.47	**59.37**	55.22	53.46	**58.99**	**55.72**	53.21	**55.85**	**55.6**
	64	60.13	59.12	**63.4**	**62.01**	58.99	**65.53**	**61.64**	60.0	**63.14**	**61.64**
	Avg.	51.25	50.74	**55.75**	**52.23**	49.41	**55.44**	**52.35**	50.74	**53.81**	**52.18**
Fine-tuning	256	39.37	**40.63**	**46.04**	**40.0**	***41.38***	**46.67**	**40.0**	**39.75**	**45.53**	**40.13**
	192	47.55	44.65	**51.45**	46.54	*45.91*	***52.7***	46.54	45.91	**49.43**	46.67
	128	48.55	48.43	**54.34**	**49.31**	***49.06***	**52.83**	46.56	**50.44**	***54.47***	**49.18**
	96	*55.47*	53.84	***60.13***	***55.72***	*54.34*	**58.49**	***56.35***	45.91	***57.48***	**56.1**
	64	*60.88*	***61.51***	***64.91***	60.88	*60.0*	**64.91**	60.75	***61.51***	**62.77**	**61.13**
	Avg.	50.36	49.81	**55.37**	**50.49**	*50.14*	**55.12**	50.04	48.7	***53.94***	**50.64**

Table 8 – Classification accuracy (%) of **PH2** comparing feature extraction (FE) from ResNet50 pre-trained with ImageNet versus fine-tuned with **HAM10000**. From (SANTOS; PONTI, 2019a).

W	**Feat.**	**Glob.**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**
ImageNet	256	88.0	–	**88.5**	87.5	–	86.5	87.5	–	88.0	87.5
	192	86.0	–	**88.0**	**86.5**	–	**89.5**	**87.5**	–	**87.5**	**86.5**
	128	83.0	84.0	**85.5**	**83.5**	**83.5**	**86.5**	**83.5**	**84.0**	**85.5**	**83.5**
	96	85.0	84.5	**86.0**	85.0	**85.5**	**86.0**	85.0	84.0	**85.5**	85.0
	64	86.5	84.0	86.5	86.0	85.0	**87.0**	86.0	83.5	**87.0**	86.5
	Avg.	85.7	84.17	**86.9**	85.7	84.67	**87.1**	**85.9**	83.83	**86.5**	**85.8**
Fine-tuning	256	87.5	–	**88.0**	87.0	–	***89.0***	87.0	–	***89.0***	87.0
	192	86.5	–	***89.0***	***87.5***	–	**88.0**	**87.5**	–	86.5	***87.5***
	128	*85.0*	*84.5*	***87.5***	83.5	83.5	**85.5**	83.5	*84.5*	84.0	83.5
	96	86.0	84.5	84.0	85.5	84.5	84.0	85.0	*85.0*	84.0	85.0
	64	85.0	***85.5***	***87.5***	**85.5**	84.5	**86.0**	**85.5**	*84.5*	**87.0**	**85.5**
	Avg.	**86.0**	*84.83*	***87.5***	*85.8*	84.17	**86.5**	85.7	*84.7*	**86.1**	85.7

results present slight superiority to global results. In general, all of them presented themselves in an equivalent form in all residual blocks.

In contrast with previous results, for Corel1000, PCA results stand out the global results and Pooling approaches (see Table 9). Due to similar data distribution (both training and test sets belong to the same dataset), the selected components from training set are, practically, the same ones that should be selected in the test set, which leads to increased performance. Setting the global performance, almost all fusion results excel them, except fine-tuned Pooling 2D when the features are extracted using the second and third residual blocks, which maintains the same result.

Table 9 – Classification accuracy (%) of **Corel1000 (test set)** comparing feature extraction (FE) from ResNet50 pre-trained with ImageNet versus fine-tuned with **Corel1000 (training set)**. The dataset was splitted in 80% for training and 20% for test. From (SANTOS; PONTI, 2019a).

W	**Feat.**	**Glob.**	**Fusion 1th block**			**Fusion 2th block**			**Fusion 3th block**		
			PCA	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**	**PCA**	**Flat.**	**2D**
ImageNet	256	91.5	–	**92.0**	91.5	–	**93.5**	91.0	–	**93.5**	91.0
	192	93.5	–	**95.5**	**94.0**	–	**96.0**	**94.0**	–	**95.0**	**94.0**
	128	94.5	**96.0**	**96.5**	**95.0**	**96.0**	**95.0**	**95.0**	**96.5**	**95.5**	**95.0**
	96	95.0	95.0	**95.5**	**95.5**	**96.0**	**96.0**	**95.5**	**95.5**	**96.5**	**95.5**
	64	95.0	95.0	**96.0**	95.0	**95.5**	**96.5**	95.0	**96.0**	**95.5**	95.0
	Avg.	93.9	**95.33**	**95.1**	**94.2**	**95.83**	**95.4**	**94.1**	**96.0**	**95.2**	**94.1**
Fine-tuning	256	*92.5*	–	***93.5***	*92.0*	–	***95.0***	*92.0*	–	92.5	*92.0*
	192	*94.0*	–	**95.5**	***94.5***	–	**95.5**	***94.5***	–	***96.5***	***94.5***
	128	*95.5*	**96.0**	**96.5**	***96.0***	**96.0**	94.5	*95.5*	**96.5**	95.5	*95.5*
	96	95.0	***95.5***	***95.5***	95.0	**95.5**	**96.0**	95.0	**95.5**	**96.5**	95.0
	64	95.0	**95.5**	**95.5**	95.0	**95.5**	**95.5**	95.0	***96.5***	**95.5**	95.0
	Avg.	*94.4*	***95.67***	***95.3***	***94.5***	**95.67**	**95.3**	*94.4*	***96.17***	***95.3***	*94.4*

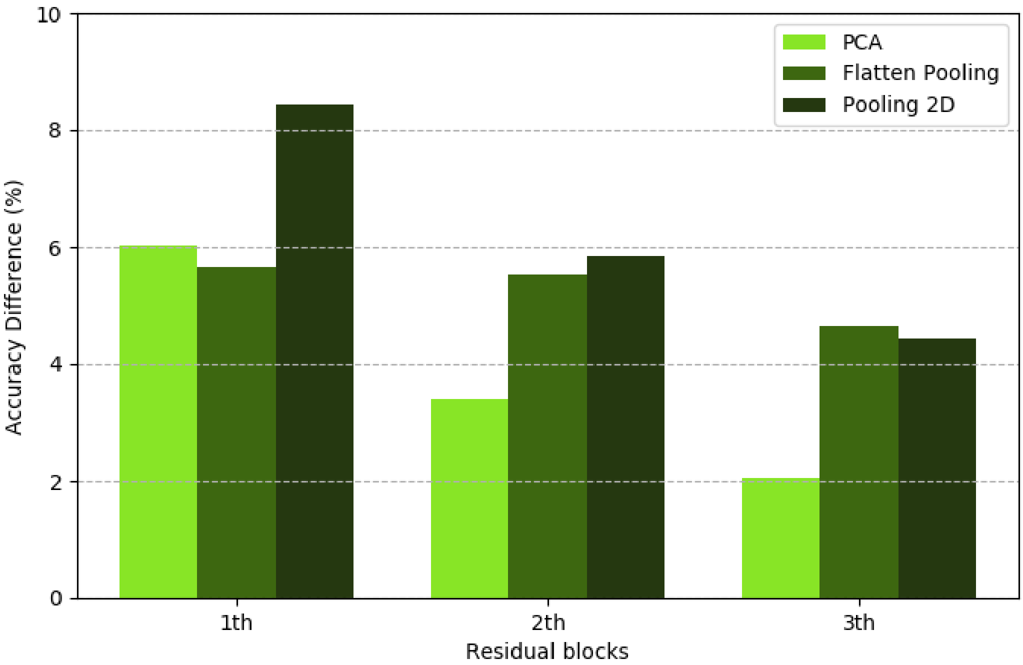

Figure 16 – Accuracy difference classification on average between local and global results of Supermarket Produce using Fruits-360 to fine-tune the ResNet50. From (SANTOS; PONTI, 2019a).

Figure 16 shows the importance of initial layers from a CNN for image classification due to relevant representativeness of the extracted feature spaces. The bars represent a comparative of local and global descriptors, illustrating the difference of performances (average among TCA performances) between them. It is interesting to note how the first block offers better performance in relation to the others. Hence, as the layers become more internal, the accuracy decreases gradually, confirming that end-layers are more representative when they are used alone. Evidently, the performance gain of multi-layer fusion is achieved with increase of computational cost. Due to the requirement of evaluating the variance in all features, PCA presents greater computational complexity. Flatten Pooling and Pooling 2D, however, only demand region delimitation and average calculation. Therefore, the computational cost is more related to dataset size than to the task complexity.

3.3 Final comments on multiple CNN layers

Based on the results from skin lesions classification (section 3.1), it is notable the discriminative capacity contained in end-layers of CNNs, being reinforced by results of dimensionality reduction and noise injection. Descriptors contained in these layers have the potential to provide representative feature spaces, either for feature extraction using only the target domain or network fine-tuning. Specifically in these experiments, fine-tuning did not offer better generalization, indicating that the amount of examples was not sufficient to improve network convergence. Additionally, in the experiments with multi-layers feature space combination (section 3.2), we explored descriptors from low-level of a CNN to complement end-layers in scenarios of feature TL. Different image domains were evaluated through fusion and data alignment, showing that images with well behaved composition are better classified by merging features from multi-layers. Global descriptors are more adequate to be used in domains with more clutter or composed of larger intra-class variance.

In this sense, these results offered important guidelines for the use of pre-trained CNNs and fine-tuning for feature extraction, considering multiple levels of descriptors from these architectures. Researchers can leverage pre-trained CNNs with ImageNet and other very large datasets to obtain feature spaces even for different image domains exploring more initial and inner layers. The depth of the CNN must be considered when performing fine-tuning: architectures with less capacity or employing skipping layers seem to converge better.

CHAPTER

4

FEATURE TRANSFER LEARNING IN ONE-CLASS SCENARIOS WITH A GENERALIZATION ANALYSIS

One of the reasons for a model not to be completely adaptable to several domains is the absence of evaluation by generalization measures. Often, studies are evaluated only by classical measures of the assigned task. Consequently, the model is suitable only for a given dataset, however, the behavior of this model in other datasets is rarely evaluated. In this sense, we noticed the difficulty of comparing one methodology with another, or even verifying which dataset offers the most representativeness in a given task for a determined model. This difficulty can be observed in both video anomaly detection and image classification tasks.

In terms of anomaly detection videos, TL can be employed in different scenarios, such as surveillance of human crowds (GUO *et al.*, 2016; HU *et al.*, 2016; CHAKER; AGHBARI; JUNEJO, 2017), pedestrian detection (ROSHTKHARI; LEVINE, 2013; HASAN *et al.*, 2016; PONTI *et al.*, 2017), and analysis of directions (human or vehicle motion) (EPAILLARD; BOUGUILA, 2016). In this context, learning implies inferring a function $f: X \rightarrow Y$ from a training feature space X to find an output feature space $Y = \{-1, +1\}$ (CHANDOLA; BANERJEE; KUMAR, 2009), where the definition of anomaly differs for each context, containing everything that is not in normal patterns (JIANG; WU; KATSAGGELOS, 2009), such as a clandestine boat in the middle of the sea or a car in a pedestrian boardwalk for surveillance scenarios. This is particularly challenging in typical anomaly detection scenarios, which only normal behavior is available for training (WEN *et al.*, 2015). Additionally, the similarity of data distribution is affected due to the variation of illumination, camera perspective, and amount of clutter in the scene (ROSHTKHARI; LEVINE, 2013; HAO *et al.*, 2017). These factors contribute to the distancing between the activities and visual content of different domains. Hence, a good measure of domain generalization can indicate which dataset is more suitable to a target video.

Aiming at greater assurance of robustness in feature TL methods, this chapter details the theoretical foundation of the Cross-Domain Feature Space Generalization Measure (CDFG) (see section 4.1) proposed to measure the domain generalization of feature spaces. This measure was direct applied to one-class problem with surveillance video anomaly detection (detailed on subsections 4.2.2 and 4.2.3) and to image classification networks in different levels of learning (described on subsection 5.2.4).

4.1 Cross-Domain Feature Space Generalization Measure

Machine learning is a field that includes the idea of developing theoretical guarantees to support what is called "learning" within the context of each algorithm. Consistently, the most stable theory is Statistical Learning Theory (SLT) (VAPNIK, 1999; LUXBURG; SCHÖLKOPF, 2011). Since its introduction, SLT has been widely used to ensure the quality of machine learning studies and, more specifically, to support the mathematical proofs that guarantee SVM generates optimum classifiers (MELLO; PONTI, 2018). Hence, based on the SLT tools, we proposed a new metric to evaluate cross-domain TL systems, asking: how can one measure generalization of a feature space produced by some method?

One of the pertinent concepts in SLT is the generalization of a solution, which represents a divergence that measures how well a classifier (or detector) performance with unseen (test) data is consistent with its performance on seen (training) data. This concept is mathematically expressed as:

$$|R_{emp}(f_n) - R(f_n)|, \tag{4.1}$$

where $R_{emp}(f_n)$ is the risk of a classifier f_n evaluated over the training set (the empirical risk) and $R(f_n)$ is the true risk (expectancy of loss) of same f_n over "all data" (called expected risk). The idea of true risk is totally abstract because it is an intractable quantity. However, this concept highlights the importance of not losing ourselves only with classic metrics and training costs, which may not completely represent the quality of a model (MELLO; PONTI, 2018). Consequently, our view is that TL methodologies cannot be evaluated solely by performance metrics, such as Receiver Operating Characteristic (ROC) for anomaly detection or accuracy for classification. Indeed, if we aim to measure how well a system trained in a domain performs in a dissimilar domain, the idea of generalization fits perfectly. Hence, two metrics were proposed: Partial Cross-domain Feature Space Generalization (G_{part}); and Complete Cross-domain Feature Space Generalization (G_{comp}):

$$G_{part}(f_n^A) = \left| \underset{x \in \mathscr{X}^A}{R(f_n^A)} - \underset{x \in \mathscr{X}^B}{R(f_n^A)} \right| \tag{4.2}$$

$$G_{comp}(f_n^A, f_n^B) = \frac{1}{2}\left(\left|\underset{x\in\mathscr{X}^A}{R(f_n^A)} - \underset{x\in\mathscr{X}^B}{R(f_n^A)}\right| + \left|\underset{x\in\mathscr{X}^B}{R(f_n^B)} - \underset{x\in\mathscr{X}^A}{R(f_n^B)}\right|\right) \tag{4.3}$$

Considering two domains (A and B) and their respective feature spaces ($\mathscr{X}^A$ and $\mathscr{X}^B$), the expression $\underset{x\in\mathscr{X}^A}{R(f_n^A)}$ denotes the risk of classifier f_n^A, trained using A, over the feature space $\mathscr{X}^A$ and $\underset{x\in\mathscr{X}^B}{R(f_n^A)}$ denotes the risk of the same classifier f_n^A over the feature space of the second domain, $\mathscr{X}^B$; the mirrored definition is valid for $\underset{x\in\mathscr{X}^B}{R(f_n^B)}$ and $\underset{x\in\mathscr{X}^A}{R(f_n^B)}$.

The two functions represent different levels of domain generalization, in which important guidelines should be followed. Hence, G_{part} and G_{comp} are meaningful metrics if: (i) the set of admissible functions from the classifier/detector are the same, e.g. same parameters on SVM training setup; (ii) both feature spaces $\mathscr{X}^A$ and $\mathscr{X}^B$ are described by the same set of descriptors; and (iii) the domain mapping method has no prior knowledge of unseen (test) data on either domain. These constraints are fundamental to maintain consistency among TL models. Hence, the first restriction assures the generalization reliability due to maintain the same comparative measure and classifier configuration. Additionally, by the second restriction is guaranteed the same amount of features and the same methodology for attributes generation. Finally, the complete independence of training sets in relation to test sets attests the uniform evaluation of the task.

Based on the G_{part} and G_{comp} definitions, we are going to introduce three particular levels of analysis to measure domain generalization. Considering the pair results of two methods α and β, composed of classification/detection algorithm and TL techniques to build the feature space, the first level of the principle of empirical risk minimization for each methodology is obtained by the expression:

$$G_{part}(f_\alpha^A) < G_{part}(f_\beta^A) \tag{4.4}$$

With this inequality satisfied, one could claim that method α is capable of generalizing well from domain A to domain B. One can also verify the G_{part} measure from the "opposite direction" and confirm if the α methodology is also better than β at generalizing from B to A, as expressed by the inequality:

$$G_{part}(f_\alpha^B) < G_{part}(f_\beta^B) \tag{4.5}$$

Grounded in these two expressions, G_{part} provides an understanding of the first level of domain generalization: how well the space obtained from one domain is applicable to another and how this applicability is captured by the chosen methodology. Therefore, G_{part} is a good representation of how well the domain A offer well-suited learning to domain B with consistent performance. However, this level of generalization is “one-way” direction, i.e we are mapping the feature transferred only in the $A \longrightarrow B$ direction. To obtain a more precise and rigorous analysis of the TL method itself, we should compare using the G_{comp} measure:

$$G_{comp}(f_{\alpha}^{A}, f_{\alpha}^{B}) < G_{comp}(f_{\beta}^{A}, f_{\beta}^{B}) \tag{4.6}$$

Using G_{comp} a better measure of the quality of the transfer system is presented, testing its robustness over different contexts and pairs of domains. However, the best use of CDFG Measure is applying both G_{part} and G_{comp} at the same time, since the G_{comp} can be influenced by high discrepancy between the two G_{part} that compose it. Hence, it is primordial that all three comparisons are taken into account to compare any two competing methods. In all these expressions, lower results imply less divergence, where the concept of generalization is more substantial.

4.2 Features generalization for surveillance videos

To evaluate the practical scenario of CDFG Measure (SANTOS; RIBEIRO; PONTI, 2019) on one-class scenario, it was performed a cross-domain feature generalization experiment extracting features via a pre-trained VGG-19 (SIMONYAN; ZISSERMAN, 2014) with ImageNet dataset (RUSSAKOVSKY *et al.*, 2015) to detect anomalous activity in video, as shown in Figure 17. Experiments were designed on transferring knowledge by: (i) cross-feature embedding using 4096 features from VGG-19 second last layer, which only relates one training set to another test set. In this first scenario, considered the baseline, the detector is trained with the training set of one video and tested with the test frames of another video, without any pre-processing or TL techniques; (ii) cross-domain linear transformation with 80 features by PCA (JOLLIFFE, 1986), selecting the main components from training set and applying them to the test set; and (iii) transformed space by TCA (PAN *et al.*, 2011) with also 80 features, in which both training sets were used to generate a new latent space.

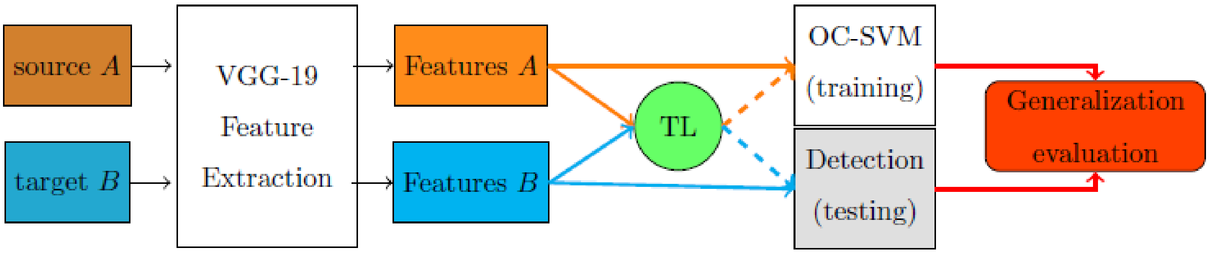

Figure 17 – Experimental setup for features generalization on one-class scenario: both source and target domains feature spaces embedding are independently computed via the same deep network model, then the source A is used to train an One-Class SVM, while target B is tested on this trained model. Transfer learning (TL) can be used to transform such spaces before training/testing (indicated in dashed lines). From (SANTOS; RIBEIRO; PONTI, 2019).

Seven anomaly detection videos/datasets were used in these experiments, which one differs from others in domains (natural and urban scenarios), frames resolution, amount of training examples, illumination, perspectives, and presence of undesirable objects in the scene composition, as presented in Figure 18. For natural scenario, Canoe (JODOIN; KONRAD; SALIGRAMA, 2008), Boat-River (ZAHARESCU; WILDES, 2010), and Boat-Sea (ZAHARESCU; WILDES, 2010) are three distinct videos that show the presence (anomaly) or absence of boats from different perspectives. Canoe consists of 1050 frames of 240 × 320 pixels in which the first 200 are intended for training. Boat-River has higher resolution (576 × 740) with only 80 training frames. With the presence of occlusion (bridge), Boat-Sea is a video less correlated with the first two, with 100 frames for training of 576 × 720 resolution. In urban scenario, Ped1 and Ped2 (MAHADEVAN *et al.*, 2010) are two datasets composed of several training and test videos with the same concept of anomalies (cyclists, skaters, and others), however with different camera positions. In contrast, Belleview (ZAHARESCU; WILDES, 2010) and Train (ZAHARESCU; WILDES, 2010) characterize vehicle conversion and movement of people within the wagons as anomaly, respectively. Therefore, the definition of what is anomaly is quite diverse in these scenarios. All frames from Canoe, Boat-River, Boat-Sea, and Train videos were converted to gray scale via: $I = 0.299R + 0.587G + 0.114B$.

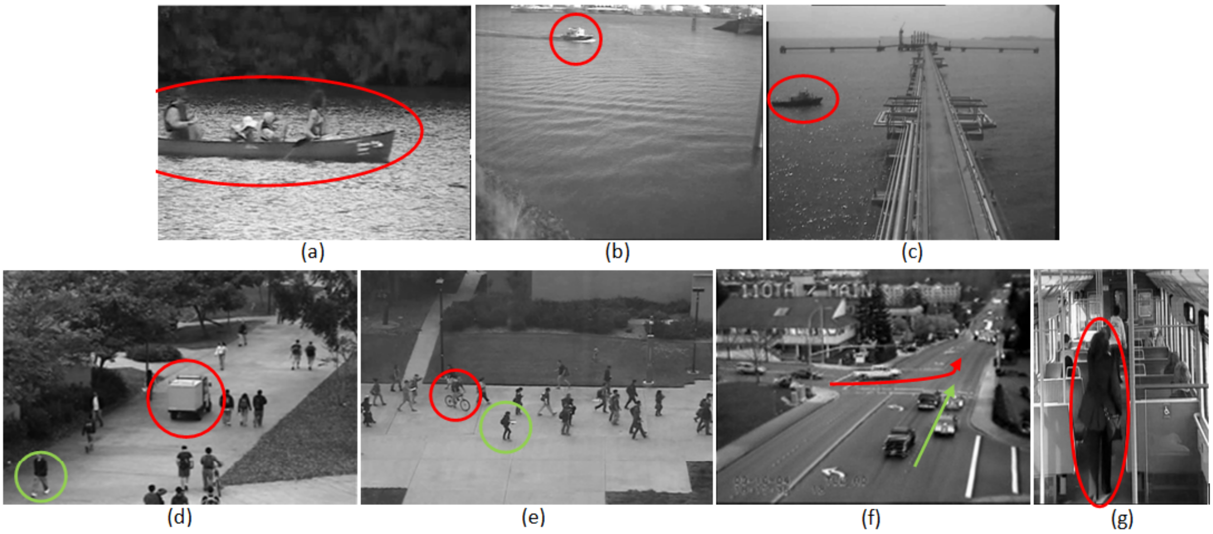

Figure 18 – Samples of test frames from: (a) Canoe; (b) Boat-River; (c) Boat-Sea; (d) Ped1; (e) Ped2; (f) Belleview; and (g) Train. Anomalous events are represented in red (boats, trucks, cyclists, vehicle conversions, and passenger movement). Examples of normal events are in green (pedestrians and straight-line pass). Adapted from (SANTOS; RIBEIRO; PONTI, 2019).

For evaluation, anomaly detection was performed in frame-level criterion, measuring Area Under the Curve (AUC) and Equal Error Rate (EER). As a detector, One-Class SVM (CHEN; ZHOU; HUANG, 2001) with linear kernel was used, allowing $v = 0.25$ as the amount of outliers. With the CDFG Measure it is possible to quantify which dataset provides the best learning rate for a target domain. Also, the results intensify the research for more robust solutions, in which only classical evaluation metrics are insufficient to distinguish the real discriminative feature space. Hence, the CDFG Measures can be explored in the context of choosing which feature extraction method better suits some task, or to merge different datasets in order to accumulate a larger training set and, therefore, increase learning guarantees. The results are divided into three subsections: anomaly detection evaluation (subsection 4.2.1); transfer learning decomposition (subsection 4.2.2); and negative transfer learning analysis (subsection 4.2.3).

4.2.1 Anomaly detection evaluation

Table 10 presents the anomaly detection results for natural scenarios (Canoe, Boat-River, and Boat-Sea) using: (i) the original VGG-19 feature embedding (Full VGG-19) with 4096 attributes; (ii) after transformation with PCA (reduction to 80 features); and (iii) applying unsupervised TL by TCA (also with 80 features)[1]. Considering the metrics AUC and EER to compare the three methodologies, it is observed that TCA is better in 66.6% of the pairs tested, mainly when the source domain is Boat-River. Furthermore, the average AUC across all TCA sets was 86.68%, meanwhile for Full VGG-19 was 70.47%. As expected, the PCA performance is exceeded by TCA in all standpoint: number of pairs with higher result (1 versus 6); average AUC (69.52% versus 86.88%); and average EER (33.3% versus 16.86%). Also, there are outstanding results of TCA when compare to Full VGG-19 and PCA: Boat-Sea ⟶ Boat-River with an improvement of 30.35% (in relation to Full VGG-19); and Boat-River ⟶ Boat-Sea in 31.41% to PCA. Therefore, as expected, the feature space provided by TCA overcomes Full VGG-19 and PCA for natural scenarios[2].

Table 10 – Anomaly detection in natural scenarios. From (SANTOS; RIBEIRO; PONTI, 2019).

Source ⟶ Target	Full VGG-19		PCA		TCA	
	AUC	EER	AUC	EER	AUC	EER
Canoe ⟶ Canoe	**92.63**	**12.65**	63.97	39.66	71.66	32.49
Boat-River ⟶ Canoe	92.75	12.65	53.61	48.1	**99.1**	**3.04**
Boat-Sea ⟶ Canoe	92.75	12.65	85.44	18.56	**97.5**	**4.64**
Boat-River ⟶ Boat-River	63.24	36.75	74.35	25.64	**90.59**	**9.4**
Canoe ⟶ Boat-River	**63.24**	**36.75**	50.42	49.57	61.11	38.88
Boat-Sea ⟶ Boat-River	64.52	35.47	61.96	38.03	**94.87**	**5.12**
Boat-Sea ⟶ Boat-Sea	54.97	46.15	**97.01**	**9.89**	91.37	16.48
Canoe ⟶ Boat-Sea	55.0	46.15	83.39	26.37	**86.99**	**19.79**
Boat-River ⟶ Boat-Sea	55.2	46.15	55.54	43.96	**86.95**	**21.91**
Average	70.47	28.37	69.52	33.3	**86.88**	**16.86**

Considering only the urban scenarios (see Table 11), Full VGG-19 overcomes PCA and TCA in number of pairs with better performances (7 of 16 pairs), especially when Ped2 or Belleview is the target domain. However, the variation between the Full VGG-19 and TCA averages is practically negligible: 63.84% versus 62.8% for AUC; and 39.41% versus 40.51%

[1] Due to restrict size of Boat-River training set, which contains only 80 examples.

[2] In Tables 10-15, the best result from each row is in bold.

for EER. Contrarily to the results with natural scenarios, PCA has positive results in urban domains. It is important to emphasize that the concept of anomalies among the urban domains is very dissimilar, implying that the feature learning should not be totally transferred, what causes negative transfer. Hence, considering only domains with the same concept of anomalies (Ped1 and Ped2), TCA stands out when compare to Full VGG-19 and PCA in AUC and EER averages.

Table 11 – Anomaly detection in urban scenarios. From (SANTOS; RIBEIRO; PONTI, 2019).

Source ⟶ Target	**Full VGG-19**		**PCA**		**TCA**	
	AUC	**EER**	**AUC**	**EER**	**AUC**	**EER**
Ped1 ⟶ Ped1	50.91	51.4	**71.46**	**35.17**	62.94	39.68
Ped2 ⟶ Ped1	50.82	51.46	**64.01**	**39.13**	60.39	41.7
Belleview ⟶ Ped1	51.77	50.75	**76.12**	**30.56**	58.86	45.89
Train ⟶ Ped1	53.42	51.75	60.65	39.72	**71.02**	**33.66**
Ped2 ⟶ Ped2	**80.34**	**26.26**	55.24	44.13	74.16	33.26
Ped1 ⟶ Ped2	**80.18**	**26.25**	56.95	46.14	67.06	38.54
Belleview ⟶ Ped2	**80.88**	**26.26**	69.46	34.63	65.16	38.01
Train ⟶ Ped2	**81.81**	**25.69**	61.77	41.89	50.11	52.51
Belleview ⟶ Belleview	68.91	33.47	50.54	51.38	**72.63**	**32.24**
Ped1 ⟶ Belleview	**68.67**	**33.45**	56.22	45.45	68.39	35.25
Ped2 ⟶ Belleview	**68.73**	**33.47**	60.42	40.63	65.24	39.12
Train ⟶ Belleview	**69.1**	**32.92**	54.36	49.31	68.65	34.35
Train ⟶ Train	53.97	47.2	**57.67**	**42.84**	51.88	51.47
Ped1 ⟶ Train	54.02	46.73	**57.75**	**46.16**	53.98	43.96
Ped2 ⟶ Train	54.13	46.67	55.47	49.0	**55.56**	**42.68**
Belleview ⟶ Train	53.85	46.84	50.63	51.46	**58.86**	**45.89**
Average	**63.84**	**39.41**	59.92	42.97	62.8	40.51

Although TCA is superior to Full VGG-19 and PCA, those classic metrics (AUC and EER) are not enough to guarantee the feature space generalization. Analyzing those performances in isolation gives an imprecision due to the great diversity of results achieved. For these reasons, the CDFG Measure (SANTOS; RIBEIRO; PONTI, 2019) offers a more detailed and reliable comparison if one methodology overcomes other.

4.2.2 Transfer learning decomposition

Using the anomaly detection metrics (AUC and EER), the CDFG Measure was evaluated on features spaces from Full VGG-19 cross-domain, PCA cross-domain, and TL by TCA[3]. At the first moment, it was applied exclusively the inequations (4.4) and (4.5) to obtain G_{part}, in which the results are presented in Table 12. In general, as it can observe, TL by TCA overcomes the competing methods in both metrics. Hence, for G_p AUC averages, TCA was 8.47%, PCA in 17.27%, and Full VGG-19 in 22.43%. This same behavior occurs with G_p EER, in which TCA achieved the best rate with 8.1%. In view of similarity between domains, the natural videos

[3] The closer to zero, the better

of Boat-River and Boat-Sea are closer in the feature space mapping by TCA in both directions. Moreover, in context of different concepts of anomalies, there are also great applicability of TL in the features spaces generated by TCA, implying that the transfer rate is more relevant from Ped1 to Belleview. As expected, Ped1 offers high learning rates for Ped2, however the opposite direction does not occur in the same intensity. Another interesting highlight is the PCA performance when compared to Full VGG-19, demonstrating that dimensionality reduction increases the performance during cross-feature. This last remark contradicts the isolated analysis from Table 11, implicating the importance of CDFG Measure.

Table 12 – Partial CDFG Measure. From (SANTOS; RIBEIRO; PONTI, 2019).

Source $\longrightarrow$ Target	**Full VGG-19**		**PCA**		**TCA**	
	G_p **AUC**	G_p **EER**	G_p **AUC**	G_p **EER**	G_p **AUC**	G_p **EER**
Boat-River $\longrightarrow$ Canoe	29.51	24.1	20.74	22.46	**8.51**	**6.36**
Boat-Sea $\longrightarrow$ Canoe	37.78	33.5	11.57	8.67	**6.13**	**11.84**
Canoe $\longrightarrow$ Boat-River	29.39	24.1	13.55	9.91	**10.55**	**6.39**
Boat-Sea $\longrightarrow$ Boat-River	9.55	10.68	35.05	28.14	**3.5**	**11.36**
Canoe $\longrightarrow$ Boat-Sea	37.63	33.5	19.42	13.29	**15.33**	**12.7**
Boat-River $\longrightarrow$ Boat-Sea	8.04	9.4	18.81	18.32	**3.64**	**12.51**
Ped2 $\longrightarrow$ Ped1	29.52	25.2	**8.77**	**5.0**	13.77	8.44
Belleview $\longrightarrow$ Ped1	17.14	17.28	25.58	20.82	**14.27**	**10.47**
Ped1 $\longrightarrow$ Ped2	29.27	25.15	14.51	10.97	**4.12**	**1.14**
Belleview $\longrightarrow$ Ped2	11.97	7.21	18.92	16.75	**7.47**	**5.77**
Ped1 $\longrightarrow$ Belleview	17.76	17.95	15.24	10.28	**5.45**	**4.43**
Ped2 $\longrightarrow$ Belleview	11.61	7.21	**5.18**	**3.5**	8.92	5.86
Average	22.43	19.6	17.27	14.0	**8.47**	**8.10**

Table 13 – Complete CDFG Measure. From (SANTOS; RIBEIRO; PONTI, 2019).

Datasets	**Full VGG-19**		**PCA**		**TCA**	
	G_c **AUC**	G_c **EER**	G_c **AUC**	G_c **EER**	G_c **AUC**	G_c **EER**
(Canoe, Boat-River)	29.45	24.1	17.14	16.18	**9.53**	**6.37**
(Canoe, Boat-Sea)	37.70	33.5	15.49	10.98	**10.73**	**12.27**
(Boat-River, Boat-Sea)	8.79	10.04	26.93	23.23	**3.57**	**11.93**
(Ped1, Ped2)	29.4	25.2	11.6	7.99	**8.95**	**4.79**
(Ped1, Belleview)	17.5	17.6	20.4	15.6	**9.86**	**7.45**
(Ped2, Belleview)	11.79	7.21	12.05	10.12	**8.19**	**5.82**

The Partial CDFG Measure, G_{part}, excludes more complex and pertinent aspects to the feature spaces, except in cases where there will be only contribution from one domain to other. This scenario is noticeable in cases where the domain source comprised large amounts of data and, consequently, it is sufficient to provide information to itself, not requiring similar domains or prior learning. However, in more complex and accurate scenarios, G_{comp} offers a deeper analysis of feature TL methodologies. For this transfer level, results in Table 13, TCA offers even more generalization in relation to the competing methods. Considering similar domains, the latent space generated by TCA is highly applicable (all using G_c AUC): Boat-River and Boat-Sea with 3.57%; Canoe and Boat-River with 9.53%; and Ped1 and Ped2 with 8.95%. Even when the concept of anomalies is different, the performance gain with TCA is evidenced (Ped2 and Belleview with 8.19%).

The last level of feature TL performance is to guarantee that the three inequalities (4.4, 4.5, and 4.6) are satisfied simultaneously. In this scenario, we can ensure that G_{comp} is contemplated without one G_{part} compensating the opposite direction. In view of the experiments performed, it is noticed that G_{comp} had compensation for TCA in (Ped1, Ped2) and (Belleview, Ped2). Although Ped1 and Ped2 have the same concept of anomalies, the position of the cameras hinders the direct TL, requiring pre-processing methods to facilitate the use of previously acquired knowledge. In the second urban scenario, the anomaly concept of Belleview and Ped2 is very different, semantically and visually: Belleview targets vehicles conversion, while Ped2 anomalies are related to the presence of vehicles on the scene.

4.2.3 Negative transfer learning analysis

A major concern in feature TL is to apply only the acquired knowledge that favors the improvement of the task for the new target domain. For this purpose, it is important to evaluate if the source domain is sufficiently related to target domain to avoid the negative transfer (TORREY; SHAVLIK, 2010). The negative transfer occurs when the TL method achieved a lower performance when compared with a method which does not perform TL (PAN; YANG *et al.*, 2010). In this context, G_{part} and G_{comp} should be applied to measure if the source domain or the methodology are suitable for a designated task.

Table 14 – Partial CDFG Measure: negative TL analysis. From (SANTOS; RIBEIRO; PONTI, 2019).

Source ⟶ Target	**Full VGG-19**		**PCA**		**TCA**	
	G_p **AUC**	G_p **EER**	G_p **AUC**	G_p **EER**	G_p **AUC**	G_p **EER**
Train ⟶ Ped1	**0.55**	**4.55**	2.98	3.12	19.14	17.81
Train ⟶ Ped2	27.84	21.51	4.1	0.95	**1.77**	**1.04**
Train ⟶ Belleview	15.13	14.28	**3.31**	**6.47**	16.77	17.12
Ped1 ⟶ Train	**3.11**	**4.67**	13.71	10.99	8.96	4.28
Ped2 ⟶ Train	26.21	20.41	**0.23**	**4.87**	18.6	9.42
Belleview ⟶ Train	15.06	13.37	**0.09**	**0.08**	13.77	13.65

Tables 14 and 15 present correlation results between videos/datasets of urban scenarios for G_{part} and G_{comp}, more specifically between the Train video and the other ones (Ped1, Ped2, and Belleview). The Train video presents an anomaly concept distinct from the others, in which the dissimilarity between them (background and objects) are highly perceivable. Considering G_{part} (detailed in Table 14), it is point out that Train only offers good levels of feature TL to Ped2, due to Full VGG-19 and PCA had performed better than TCA, characterizing a negative transfer scenario. Consequently, G_{comp} (see Table 15) implies that Train is not a suitable domain for Ped1, Ped2, or Belleview.

Table 15 – Complete CDFG Measure: negative TL analysis. From (SANTOS; RIBEIRO; PONTI, 2019).

Datasets	**Full VGG-19**		**PCA**		**TCA**	
	G_c **AUC**	G_c **EER**	G_c **AUC**	G_c **EER**	G_c **AUC**	G_c **EER**
(Train, Ped1)	**1.83**	**4.61**	8.35	7.06	14.1	11.0
(Train, Ped2)	27.0	21.0	**2.17**	**2.91**	10.2	5.23
(Train, Belleview)	15.1	13.8	**1.7**	**3.28**	15.3	15.4

4.3 Final comments on generalization analysis

The results achieved in these experiments express the applicability of CDFG Measure to indicate which domains offer the most learning rate for a target domain in a given task. As mentioned earlier, classic evaluation metrics do not provide a perform of model generalizability. We can highlight this fact due to the wide variety of results (AUC and EER) achieved with Full VGG-19, PCA, and TCA. With the CDFG Measure we confirmed that TCA (the only method that applies TL) stands out from the others, indicating when the transfer should occur, avoiding the negative transfer. Hence, CDFG Measure is an evaluation method that offers quantitative analysis of learning guarantees from models built for transfer of knowledge.

CHAPTER 5

FEATURE TRANSFER LEARNING IN SEMI-SUPERVISED SETTINGS

In terms of image classification, when fine-tuning is applied in deep networks for TL, one of the concerns is about how much data is required (SRIVASTAVA *et al.*, 2014). Supervised networks always will ask for annotate large databases, which is very expensive (SHAO; ZHU; LI, 2015). In this context, if unlabeled data is available, how to use those to improve the final learned representation? Unsupervised or, even more suitable to this scenario, semi-supervised networks are architectures that do not require much labels (REN *et al.*, 2019; LING *et al.*, 2018). Hence, a combination of CNN and AE provides a hybrid network that conciliates labeled and unlabeled data simultaneously (KUZNIETSOV; STÜCKLER; LEIBE, 2017), even for feature TL tasks.

Related to this topic, in this chapter, we introduce Weighted Label Loss (WLL) as a new loss function for hybrid semi-supervised networks, which considers the available labeling rate (described in section 5.1) to balance the individual CNN and AE loss functions. To evaluate WLL and feature TL on different levels of learning (supervised, semi-supervised, and unsupervised), experiments with image classification using different domains were performed with classical metrics, detailed in subsections 5.2.1–5.2.3, and by CDFG Measure on subsection 5.2.4.

5.1 Weighted Label Loss

In scenarios with simultaneously labeled and unlabeled data, two approaches are often presented in the literature: supervised learning where unlabeled data is discarded; or unsupervised learning where labels are neglected. However, semi-supervised techniques have been presented as an alternative to incorporate all available data, labeled or not, into the same learning model (REN *et al.*, 2019; LING *et al.*, 2018; KUZNIETSOV; STÜCKLER; LEIBE, 2017). Generically, our semi-supervised network ensemble method is modeled to learn labeled and unlabeled data in a hybrid architecture composed of a CNN and an AE. In this structure, intermediate layers are shared between them and the proposed loss function (WLL) relates the amount of labeled examples provided to optimize each branch of the network, as shown in Figure 19 (top). Therefore, the model applies supervised classification (see Figure 19 (bottom-left)) and unsupervised reconstruction (see Figure 19 (bottom-right)) functions for learning representations, combining them according to the percentage of existing labels to balance the individual loss. Consequently, this structure can be adaptable to any amount of data: only labeled; only unlabeled; or partially labeled.

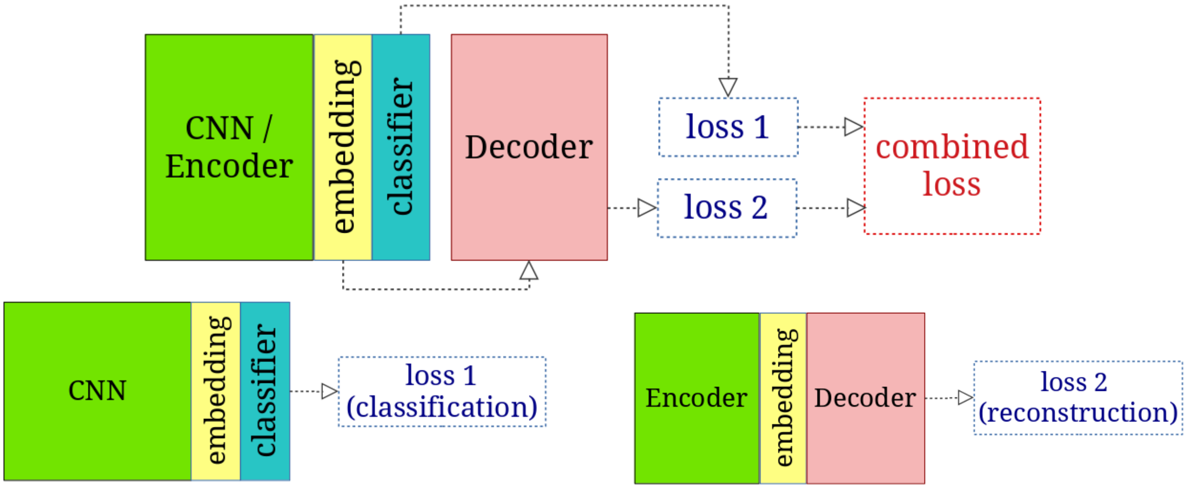

Figure 19 – Overview of the generic semi-supervised architecture: (top) combination of supervised and unsupervised networks and their losses (classification and reconstruction) to learn a feature embedding from labeled and unlabeled data; (bottom-left) the supervised CNN is trained with labels; and (bottom-right) the AE offers unsupervised training. From (SANTOS *et al.*, 2020).

The semi-supervised network is trained in two steps: (i) the AE branch is trained using only the unlabeled training data; and (ii) the hybrid network is fine-tuned using the remaining labeled data. This order of training initializes the CNN/Encoder parameters and prepare them to be adjusted during the second training (LING *et al.*, 2018). Hence, given a percentage of labeled data P from the training set, the first stage trains the AE from scratch using the $100\% - P\%$ unlabeled examples. In the following, the whole network is fine-tuned using the remaining $P\%$ of labeled examples. As the proportion of labeled data changes in different scenarios, WLL was proposed to weight the individual losses according to the percentage of P.

The Equation 5.1 describes the respective weight of the loss function from supervised branch w_{sup} while Equation 5.2 defines the unsupervised branch weight w_{uns}. Considering both equations, we have $0.5 < w_{sup} < 1.0$ and $0 < w_{uns} < 0.5$ for WLL. Consequently, this balancing ensures that $w_{sup} + w_{uns} = 1$. To apply these equations, the constraint $0 < P < 100$ should occur; otherwise, the semi-supervised learning is not characterized. Therefore, when $P = 100$ all data are labeled and only the supervised branch must be considered; when $P = 0$ all data are unlabeled and only the unsupervised branch must be considered.

$$w_{sup} = 0.5 + 0.5 \cdot \frac{P}{100} \tag{5.1}$$

$$w_{uns} = 0.5 - 0.5 \cdot \frac{P}{100} \tag{5.2}$$

To measure the progress from the supervised network training is employed the Cross-entropy loss $l^{(ce)}$. The unsupervised network training is measured via reconstruction error of MSE (ε). Accordingly, the semi-supervised network loss function Weighted Label Loss is given by:

$$WLL = w_{sup} \cdot l^{(ce)} + w_{uns} \cdot \varepsilon \tag{5.3}$$

5.2 Features embedding on semi-supervised learning

To apply the WLL for semi-supervised image classification tasks, it is required an architecture that consists of three parts: (i) an encoder whose output forks to (ii) a CNN Top with a classifier layer and (iii) a decoder with a reconstruction loss function. Hence, the flow of encoder and CNN Top composes the supervised learning; the flow of encoder and decoder composes the unsupervised learning. The proposed model ensemble is a general structure that can be configured using different sequence of layers.

For the experiments, two types of architectures were investigated: sequential convolutional and pooling layers forming the SmallNet (SN); and residual blocks for SmallResNet (SRN). Inspired by ResNet, SRN has variants changing the amount of residual blocks: SRN-1 using 1 residual block; SRN-2 for 2 residual blocks; and SRN-4 when the network has 4 residual blocks. All layers apply ReLu as activation function, except the dense layer on CNN Top with softmax activation, as described in Figure 20. Detailing the networks, **SN** has three layers in the encoder: a convolution of 8 filters of 3×3 size and stride[1] of 2×2; another convolution with 8 filters of 3×3 size, however, with stride of 1×1; and a max-pooling of 2×2. After the flatten layer is put the prediction layer on the CNN Top. For **SRN** an initial convolution layer of 64 filters of 7×7 is followed by a max-pooling of 2×2 with stride 2×2. Then, each residual block is composed of three convolutional layers of different sizes: 64 filters of 1×1, 3×3, and 1×1. Before the flatten layer, the dimensionality is reduced via an average pooling of 7×7. All **decoders** have layers in reverse order in relation to the encoder. Consequently, the decoder output has the same size of the encoder input. The number of parameters from each network is shown in Table 16.

[1] The stride indicates how the window will slide in the output; in small ones may occur overlapping.

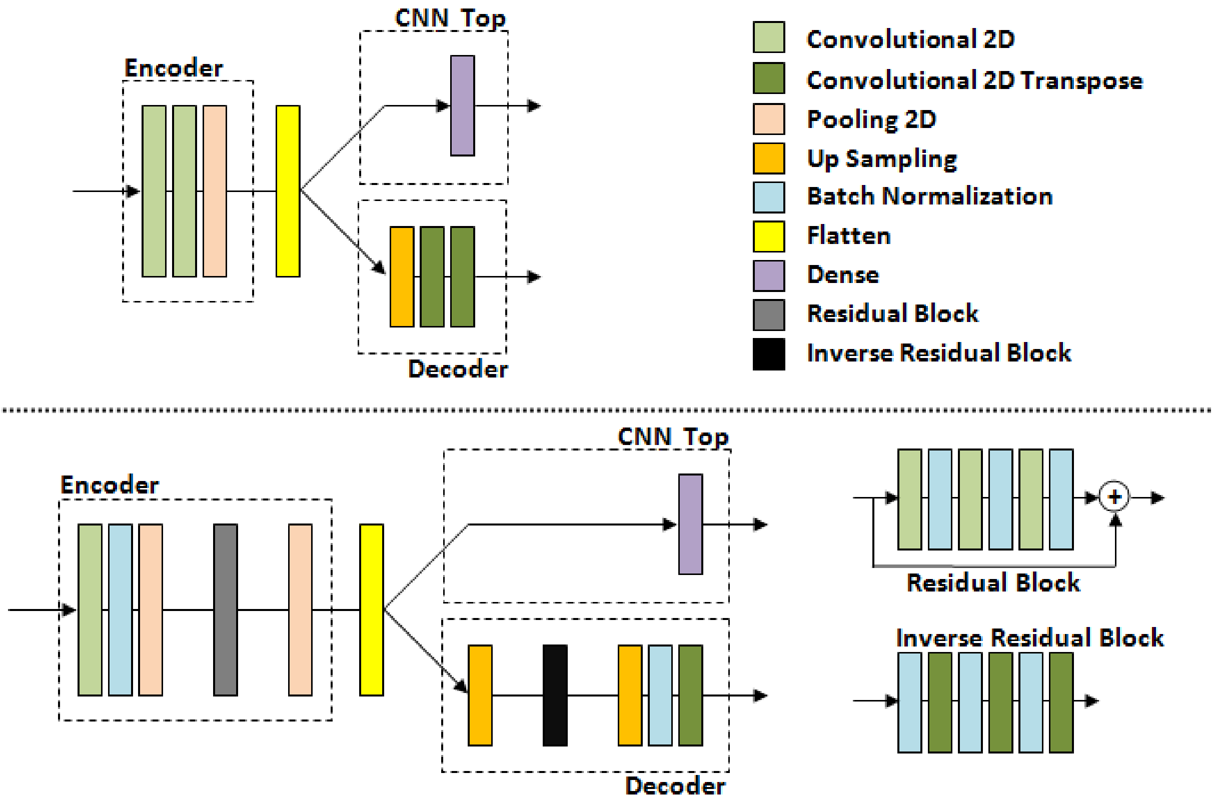

Figure 20 – Designed networks: (top) SmallNet; and (bottom) SmallResNet. Considering any image as input, it proceeds through the encoder until it reaches the flatten layer (yellow block) to generate the embedding; then, the feature map is sent to the CNN Top and to the decoder. The residual blocks vary in number (always sequential) and they are highlighted to show the internal composition of layers; the "+" represents the sum of the both flows. Adapted from (SANTOS *et al.*, 2020).

Table 16 – Number of weight parameters in each part of the networks, considering an input of 28 × 28 pixels resolution.

Model	Code	Encoder	CNN Top	Decoder	Total
SN	288	664	2890	657	4211
SRN-1	64	49472	650	49409	99531
SRN-2	64	95488	650	95425	191563
SRN-4	64	187520	650	187457	375627

Although SN is a lighter network having less layers, it has more parameters on CNN Top than SRNs due to the code size. The rationale of using such architectures is to explore networks with different capacities, where the variation in the complexity allows to evaluate whether the same behavior is identifiable in all of them. In all experiments, Adam (TZENG *et al.*, 2017; KINGMA; BA, 2014) was used as optimizer with batch size of 32 images (MASTERS; LUSCHI, 2018) during 10 epochs, defined empirically. The WLL and networks were tested using two different domains: digit and natural images on supervised, unsupervised, and semi-supervised learning.

Three different **digit image** datasets were used in the experiments performing classification within the dataset and feature TL: MNIST (LECUN *et al.*, 1998); SVHN (NETZER *et al.*, 2011); and USPS[2]. For this domain, the SN and SRNs networks receive an input of $28 \times 28 \times 1$ (see Table 17 for specific details of each dataset). Hence, SVHN images were cropped from the central pixels. Images from USPS were, first, scaled to $32 \times 32 \times 1$ followed by the same cropping of SVHN. Additionally, SVHN images were convert to gray scale using $I = 0.299R + 0.587G + 0.114B$. Comparing visual aspects, MNIST and USPS are more similar due to same background and centralized handwritten digits. However, SVHN has photographed number of buildings, i.e it presented different colors and contrasts, non-centered digits, and presence of more than one number in the same image, as shown in Figure 21.

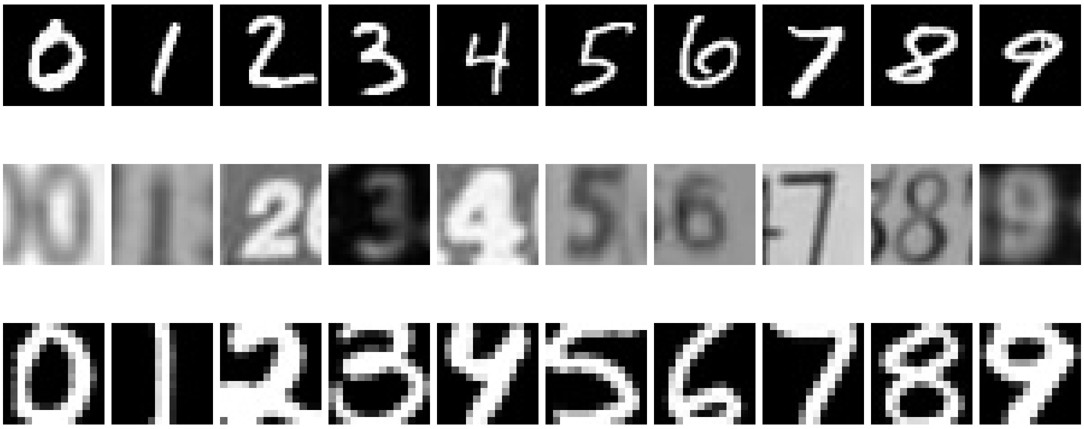

Figure 21 – Digit datasets after the pre-processing: (top) MNIST; (center) SVHN; and (bottom) USPS.

Table 17 – Datasets description used in semi-supervised networks. Adapted from (SANTOS *et al.*, 2020).

Dataset	Original Resolution	Final Resolution	Training Examples	Testing Examples	Classes
MNIST	$28 \times 28 \times 1$	$28 \times 28 \times 1$	60000	10000	10
SVHN	$32 \times 32 \times 3$	$28 \times 28 \times 1$	73257	26032	10
USPS	$16 \times 16 \times 1$	$28 \times 28 \times 1$	7291	2007	10
CIFAR-10	$32 \times 32 \times 3$	$32 \times 32 \times 3$	50000	10000	10
STL-10	$96 \times 96 \times 3$	$32 \times 32 \times 3$	5000 (labeled) + 100000 (unlabeled)	800	10

In addition to digit image datasets, CIFAR-10 (KRIZHEVSKY; HINTON, 2009) and STL-10 (COATES; NG; LEE, 2011) were also used as **photographic natural images** for semi-supervised classification. Despite these two datasets have 10 classes, only 1 of them are not presented in both datasets. Another interesting aspect of STL-10 is the unlabeled training images available, being an adequate dataset for semi-supervised learning experiments. In particular, when STL-10 is the training and the test set, the pixel resolution of $96 \times 96 \times 3$ is held. Figure 22 shows examples from each dataset.

[2] https://cs.nyu.edu/ roweis/data.html

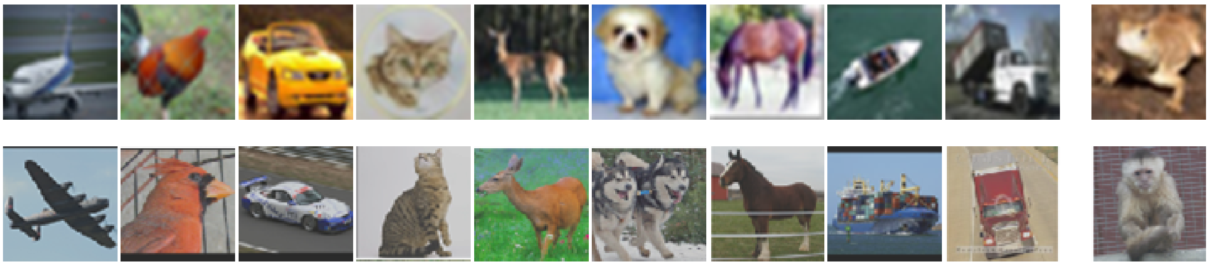

Figure 22 – Photographic natural datasets: (top) CIFAR-10; and (bottom) STL-10. These datasets have 9 classes in common, differentiating in one class: CIFAR-10 has the class “frog”; while STL-10 has the class “monkey”.

Focus on learning discriminative representations, the networks are used as feature extraction modules after the training, selecting the flatten layer to generate the feature maps. Hence, a source dataset is used to train the network, which provides the weights for a feature target dataset. With the feature vector, the SVM classifier is applied to perform a 5-fold cross validation, resulting in a final accuracy and standard deviation. This setup is validated within the same dataset, i.e training and testing with MNIST, and through feature TL, for example training the network with MNIST and testing with SVHN. The experiments include different percentages of labels provided: $P = 100\%$ for supervised learning using only the CNN branch; $P = 0\%$ for unsupervised learning using only the AE branch; and $P = 90\%$, 70%, 50%, 30%, and 10% for semi-supervised learning using WLL as loss function.

5.2.1 Supervised and unsupervised feature transfer classification

First, Tables 18 and 19 present the results of supervised and unsupervised learning, i.e $P = 100\%$ and $P = 0\%$, when the training and test sets coming from the same dataset. Accordingly, the supervised learning performances of SN and SRNs are compared with some competing methods: CNN inter-class (FEI *et al.*, 2018); supervised embedding function in AEs (DAE) (PAUL; MAJUMDAR; MUKHERJEE, 2018); and dropout layers in a CNN (MCNN-DS) (YANG; YANG, 2018). Comparing these models with SN and SRN architectures[3], our networks are simpler, involving only a few layers and a softmax loss function. Despite the high performances from those, SRN-4 presents comparable (USPS) or better (MNIST and SVHN) results. As expect, the performance is gradually increased with more residual blocks on SRN architectures. Furthermore, analyzing the SN performance on MNIST, the result is equivalent to competing methods. For unsupervised learning, the competing methods involve clustering embedding (TZOREFF; KOGAN; CHOUKROUN, 2018) and Gaussian variational AEs (DILOKTHANAKUL *et al.*, 2016). Using only the AE branch, SRN-4 has its performance only 1% lower than clustering embedding for MNIST. As expected, the unsupervised results are lower than supervised learning, especially with SVHN (decrease of 82.17% to 56.58% on SN) due to be a more challenging dataset. However, when the test is performed for MNIST (99.38%

[3] For each column on Tables 18-21, results in bold indicate the highest performance.

to 96.43% applying SRN-4) or USPS (91.43% to 88.19% at SRN-1) the results are closer.

Table 18 – Classification accuracy (%) on supervised feature learning. From (SANTOS *et al.*, 2020).

Model	**MNIST**	**SVHN**	**USPS**
CNN inter-class	—	92.68	—
DAE	97.12	33.1	**95.44**
MCNN-DS	98.43	—	—
SN	97.92 ± 0.42	82.17 ± 0.27	91.23 ± 1.42
SRN-1	99.17 ± 0.21	92.54 ± 0.25	91.43 ± 0.72
SRN-2	99.35 ± 0.16	93.78 ± 0.33	92.73 ± 1.52
SRN-4	**99.38** ± **0.14**	**94.36** ± **0.27**	93.42 ± 1.14

Table 19 – Classification accuracy (%) on unsupervised feature learning. From (SANTOS *et al.*, 2020).

Model	**MNIST**	**SVHN**	**USPS**
Clustering	**97.4**	—	—
Gaussian VAE	92.77 ± 1.6	—	—
SN	94.21 ± 0.5	**56.58** ± **0.78**	86.15 ± 2.83
SRN-1	92.32 ± 0.32	42.79 ± 0.32	**88.19** ± **0.85**
SRN-2	95.39 ± 0.41	44.83 ± 1.12	84.11 ± 1.16
SRN-4	96.43 ± 0.43	35.99 ± 0.68	77.08 ± 1.13

With feature TL, Table 20 shows the performances on supervised approach. Considering MNIST and SVHN, the highest accuracies were on SN; considering MNIST and USPS, the highest accuracies were on SRN-2. These results indicate that smaller networks may offer domain generalization due to they provide a more general feature space. An interesting observation occurs when MNIST is applied as source dataset: both accuracies of SVHN and USPS decay. As described before, SVHN has plenty of image variance, making it a more challenging dataset to classify; USPS has few training examples (only 12% of the total MNIST examples), reducing a possible network overfitting.

Table 20 – Classification accuracy (%) on supervised feature TL from source training dataset (S) to a target test dataset (T). From (SANTOS *et al.*, 2020).

Model	**S: MNIST** **T: SVHN**	**S: SVHN** **T: MNIST**	**S: MNIST** **T: USPS**	**S: USPS** **T: MNIST**
SN	**74.28** ± **0.39**	**97.03** ± **0.58**	85.15 ± 1.23	95.13 ± 0.38
SRN-1	50.8 ± 0.24	89.04 ± 0.54	89.74 ± 1.25	95.56 ± 0.31
SRN-2	45.5 ± 0.56	92.67 ± 0.55	**90.78** ± **1.18**	**95.83** ± **0.38**
SRN-4	47.5 ± 0.47	94.2 ± 0.55	89.44 ± 1.54	94.51 ± 0.5

The results of unsupervised feature TL in Table 21 were compared to competing methods, which apply: adversarial function to align domains (ADDA) (TZENG *et al.*, 2017); adversarial function to train AE (RAAN) (CHEN *et al.*, 2018a); geometrical and distribution shift reduced on sub-spaces (JGSA) (ZHANG; LI; OGUNBONA, 2017); marginal and class-conditional distributions matching (DICD) (LI *et al.*, 2018); pseudo-labels to unlabeled samples trained using asymmetry (3AT) (SAITO; USHIKU; HARADA, 2017); and cross-domain transformation with label inference (Label Inf.) (SENER *et al.*, 2016). For this scenario, SN overcomes all competing methods and SRN architectures. Comparing these results to the supervised feature TL (in Table 20), to train the network with a different source dataset allows to generate a better feature space for target datasets: SVHN is a remarkable case, in which the accuracy is improved from 54.82% to 74.28%.

Table 21 – Classification accuracy (%) on unsupervised feature TL from source training dataset (S) to a target test dataset (T). From (SANTOS *et al.*, 2020).

Model	S: MNIST T: SVHN	S: SVHN T: MNIST	S: MNIST T: USPS	S: USPS T: MNIST
ADDA	—	76.0 ± 1.8	89.4 ± 0.2	90.1 ± 0.8
RAAN	—	89.2	89.0	92.1
JGSA	—	—	80.44	68.15
DICD	—	—	77.83	65.2
3AT	52.8	86.0	—	—
Label Inf.	40.3	78.8	—	—
SN	**54.82 ± 0.39**	**94.39 ± 0.52**	**90.68 ± 1.07**	**94.48 ± 0.37**
SRN-1	32.27 ± 0.54	91.95 ± 0.58	89.24 ± 1.23	86.85 ± 0.54
SRN-2	39.44 ± 0.79	93.55 ± 0.46	89.39 ± 1.46	88.83 ± 0.75
SRN-4	48.01 ± 0.55	91.08 ± 0.76	89.19 ± 1.6	86.42 ± 0.73

Analyzing deeply the networks proposed for unsupervised feature TL, SN has only 1% of the SRN-4 parameters and provides more significant performances for MNIST as source and SVHN as target, 6.81%. In this set, MNIST and SVHN are not remarkably similar due to the different acquisition, number of digits into the image, and background. Moreover, comparing SN to SRN-2, there is an increase of 5.65% from USPS (source) to MNIST (target). Once more, USPS has a very reduced training set, indicating that high performances in small networks may occurs due to the parameters fluctuation.

5.2.2 Semi-supervised feature transfer learning on digit images

In the previous subsection, supervised and unsupervised learning employed training using only the CNN or AE branchs. However, for the semi-supervised experiments, the networks are trained using the WLL. Seeing Figure 23, the classification accuracies are illustrated within the same dataset for MNIST, SVHN, and USPS (the first three graphs) and for feature TL on the other four graphs (training set ⟶ test set). For each network was tested two balancing in the same loss function: the first one employs the WLL directly; the second one does not distinguish the weights, i.e the individual loss functions have the same importance. Therefore, when the WLL is applied, the CNN loss function receives the weight of: 0.95 for 90% of labeled

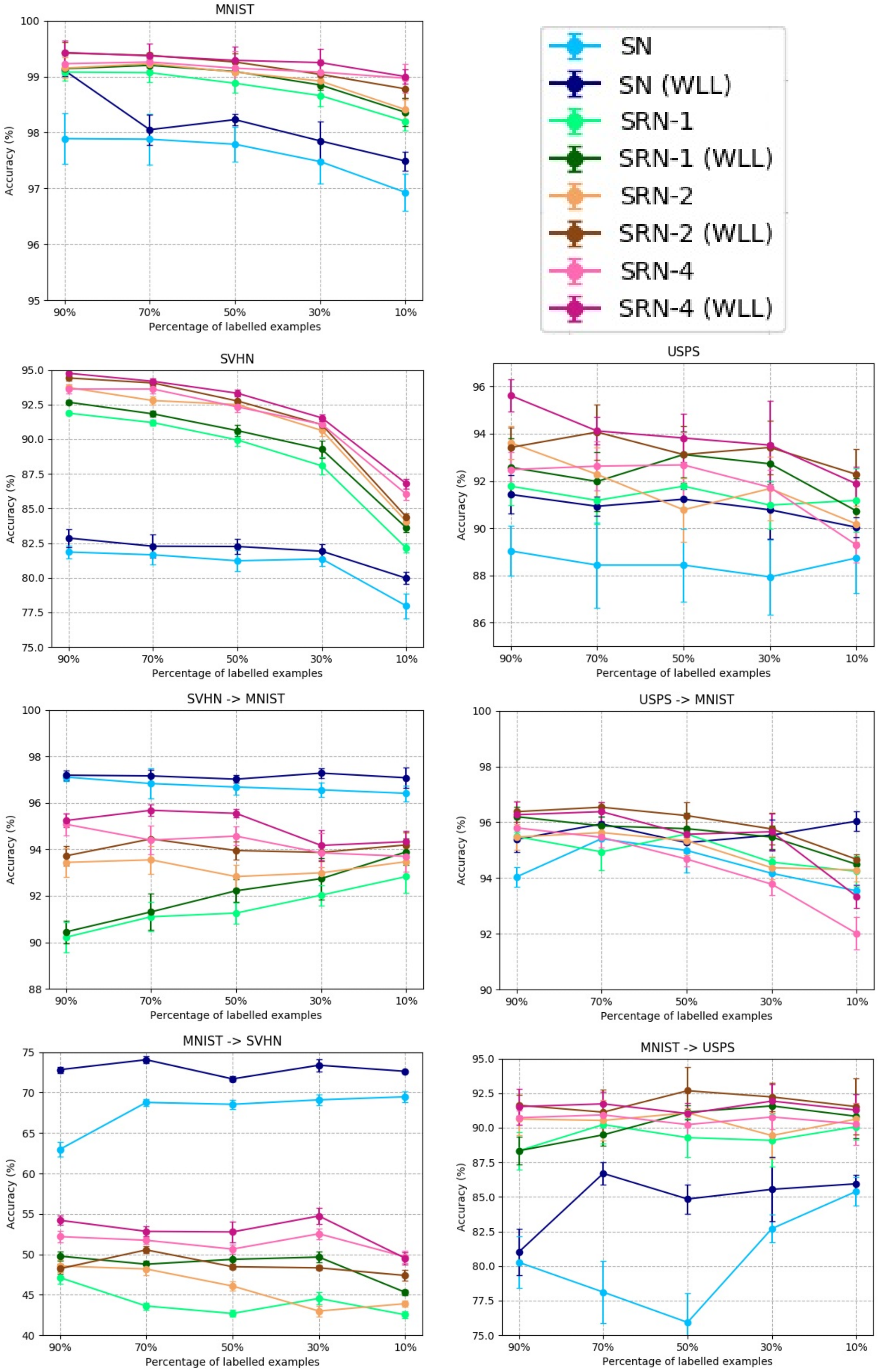

Figure 23 – Semi-supervised classification accuracies with different proportions of labels per training data. The results include training and testing with the same dataset (MNIST, SVHN, and USPS) as well as across datasets. From (SANTOS *et al.*, 2020).

examples; 0.85 for 70%; 0.75 for 50%; 0.65 for 30%; and 0.55 when it has only 10% of the labeled examples.

Analyzing the overall results, the semi-supervised results using WLL are superior than regular combination of individual losses. Furthermore, the standard deviations imply the consistency of WLL. A few semi-supervised performances are even better than the supervised approach for feature TL when there is 90% of labeled examples: MNIST $\longrightarrow$ SVHN at 6.71% (from 47.5% to 54.21%) and MNIST $\longrightarrow$ USPS at 2.09% (from 89.44% to 91.53%) in SRN-4; and SVHN $\longrightarrow$ MNIST in all networks with approximately 1% increased. As expected, the classification accuracy decreases as the proportion of labeled data to train the network, especially for feature TL scenarios. Despite of that, WLL offers a slower decrease. Seeing more deeply, USPS demonstrates more consistent performances on SRN networks. Contrarily, SVHN has a broader variance in different architectures because it is a more complex dataset than USPS and MNIST. When SVHN is employed with MNIST, either as source or as target, SN overcomes the other networks results (74.06% on SN, 48.78% on SRN-1, 50.55% on SRN-2, and 52.84% on SRN-4 for MNIST $\longrightarrow$ SVHN with 70% of labelled examples). Still, considering SVHN as source dataset for feature TL, SRNs have their performances improved due to greater parameters fluctuation obtained during the first training (using only the AE branch).

Based on all results presented on supervised, semi-supervised, and unsupervised for digit images domain, Tables 22 and 23 summarize all performances on SRN-4 using the same dataset and on SN using feature TL. As mentioned before, the accuracies gradually decay according to the proportion of labels employed during the training. However, it is interesting to observe how a few proportion of labeled examples (only 10%) provides a high increased performances from unsupervised approach: considering SVHN as source and target dataset there is an almost 51% of gain in accuracy (35.99% to 86.79%); and for MNIST as source and SVHN as target the accuracy goes 54.82% to 72.63%, representing an increase of 17.81%.

Table 22 – Comparative of classification accuracy (%) using SRN-4. From (SANTOS *et al.*, 2020).

Training	**Test**	**Super.**	**90%**	**70%**	**50%**	**30%**	**10%**	**Unsup.**
MNIST	MNIST	99.38	99.43	99.37	99.29	99.25	99.00	96.43
SVHN	SVHN	94.36	94.76	94.18	93.32	91.54	86.79	35.99
USPS	USPS	93.42	95.62	94.12	93.82	93.52	91.88	77.05

Table 23 – Comparative of feature TL classification accuracy (%) using SN. From (SANTOS *et al.*, 2020).

Training	**Test**	**Super.**	**90%**	**70%**	**50%**	**30%**	**10%**	**Unsup.**
MNIST	SVHN	74.28	72.83	74.06	71.69	73.38	72.63	54.82
SVHN	MNIST	97.03	97.19	97.16	97.02	97.28	97.08	94.39
MNIST	USPS	85.15	81.02	86.7	84.85	85.55	85.95	90.68
USPS	MNIST	95.13	95.39	95.94	95.27	95.54	96.04	94.48

To measure the excellent results from semi-supervised SN using WLL, we compared these performances with SDEC (REN *et al.*, 2019), which integrates prior knowledge of pairwise constraints to transform features. In particular, this model does not report the proportion of labeled examples used, only the number of pairwise constraints. However, the comparison is possible to be executed. Hence, considering MNIST, SDEC provides 86.11% of top accuracy in semi-supervised learning while our worst unsupervised result achieved 92.32% of accuracy, i.e an increase of 6.21%. In terms of semi-supervised, SN with 10% of labeled data achieved an accuracy of 97.49%. This behavior extends to USPS, which SDEC reaches 76.39% while our unsupervised approach achieves 77.08% and 90.04% using only 10% of labels.

In addition to the classification results, we discuss the training of the deep architecture and its convergence. Since our focus is to evaluate the proposed WLL function, we show in Figure 24 the loss values for each epoch. Generally, SRN-4 converges faster as more labeled examples are available. Also, after some epochs, we may observe that the loss tends to zero regardless of the amount of labels provided. In addition, SVHN is more challenging for this architecture in relation to MNIST and USPS, requiring more epochs to reach the zero loss.

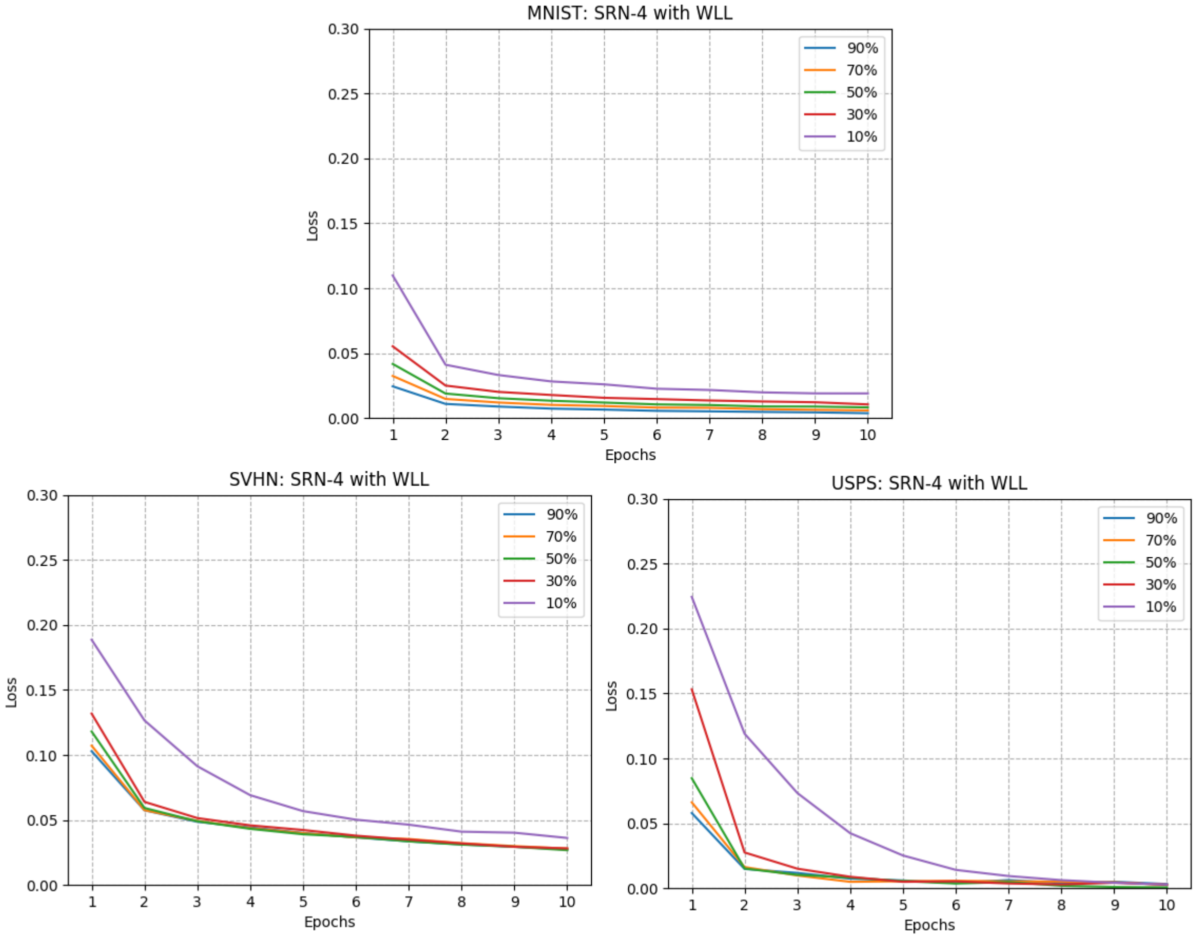

Figure 24 – WLL training loss according to the epochs on SRN-4. From (SANTOS *et al.*, 2020).

5.2.3 Semi-supervised feature transfer learning on natural images

Additionally to the digit images domain experiments, semi-supervised learning also was performed with the photographic natural domain. Consequently, CIFAR-10 and STL-10 were tested with our more complex architecture, SRN-4. First, using only labeled examples to train the SRN-4, STL-10 was experimented at different proportion of labels, as detailed in Table 24.

Table 24 – Semi-supervised classification accuracy (%) on STL-10 using SRN-4. All unlabeled examples from STL-10 were not considered in this experiment. The best result for each row is highlighted in bold. From (SANTOS *et al.*, 2020).

P (%)	without WLL	with WLL
90	**50.36** ± **0.74**	50.35 ± 1.36
70	48.43 ± 1.17	**49.73** ± **0.59**
50	46.99 ± 0.78	**47.74** ± **1.19**
30	45.66 ± 1.22	**47.59** ± **0.72**
10	44.04 ± 1.07	**44.06** ± **1.06**
Avg.	47.01	**47.9**

Table 25 – Photographic image domain semi-supervised accuracy (%) using SRN-4 in transferred features embedding. In these experiments, the classification loss (CNN) had 0.525 of weight and the reconstruction loss (AE) had 0.475 of weight. The best result for each row is highlighted in bold. From (SANTOS *et al.*, 2020).

Training set	Testing set	Labels	without WLL	with WLL
STL unlabeled + STL-10	STL-10	10	49.34 ± 1.24	**51.95** ± **0.47**
STL unlabeled + STL-9	CIFAR-9	9	40.57 ± 1.05	**42.18** ± **0.6**
STL unlabeled + STL-10	CIFAR-10	11	38.27 ± 0.69	**40.11** ± **1.59**

Considering the unlabeled and labeled examples for the STL-10 training set, three additional experiments were performed using photographic natural domain for feature TL: the first one classifies the STL-10 test set; during the second experiment, CIFAR-10 is tested using only the common classes between the two datasets, i.e the “monkey” (STL-10) and “frog” (CIFAR-10) examples were removed; and the last one maintains all classes from these two datasets. In this setup, all unlabeled examples from STL-10 are used to train the AE and then the labeled examples are used to fine-tune the hybrid network. Due to this setup, we adopted $P = 5\%$ for WLL, which represents the amount of labeled data approximately. Given the obtained accuracies, the network trained with WLL excels a better performance when compared to regular balancing, reinforcing previous results with digit images domain, as detailed in Table 25. In these results, it can be observed that the unlabeled examples during the training implied an 4% increase in the accuracy, from 47.9% on average to 51.95% for STL-10. Moreover, the accuracy from unlabeled examples in the training set (51.95%, where $P = 5\%$) overcomes all variations of P tested (see Table 24). Comparing both performances on CIFAR-10, it is interesting to note that the accuracy remains equivalent (42.18% — 40.11%). The first result was obtained using only the common classes, however, in the second one the class “frog” was not learned during the network training, which indicates that the architecture generalizes well to unknown classes. Once more, we compared our worst result from STL-10 (44.06% using 10% of labeled data) to SDEC (REN *et al.*, 2019) with 38.66%, reinforcing the representative capacity of WLL.

Aiming to compare the performances achieved from unlabeled examples of STL-10, we considered the employability of a pre-trained ResNet50 with ImageNet as an encoder. This particular encoder is composed of the first four residual blocks and simulates the first step of training, i.e only the AE branch with STL-10 unlabeled examples. In the following, for the supervised branch was coupled the softmax prediction layer and for the unsupervised branch was coupled the same SRN-4 decoder, changing only the number of filters to be equivalent to original ResNet50 encoder. Due to the larger number of parameters in the encoder output, a pooling layer of 14 × 14 was adopted. Also, to have the same number of features in comparison to SRN-4, PCA is applied to reduce the dimensionality on the feature map extracted. And, WLL maintains its $P = 5\%$. For this structure, as it can be observed in Figure 25, even when the encoder does not incorporate similarity from the source photographic domain, WLL offers greater domain generalization capacity: for STL-10 the accuracy gain was 1.19%; for CIFAR-9 an increase of 2.53%; and for CIFAR-10 at 0.91%. Furthermore, comparing directly the ResNet50 encoder pre-trained with ImageNet to the SRN-4 encoder trained with STL-10 unlabeled examples, the performance drops approximately 8% on STL-10. This result is particularly important because highlights the relevance of unlabeled images for network training. In contrast, the result of CIFAR-10 increases by around 6%, indicating that the domain generalization is provided by complex networks trained in large datasets.

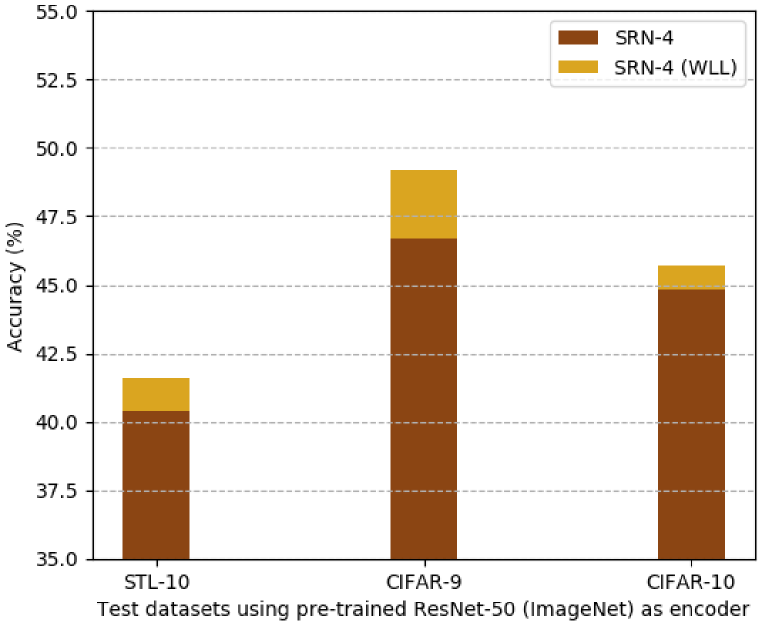

Figure 25 – Classification accuracy (%) using the ResNet50 pre-trained with ImageNet as encoder for SRN-4. From (SANTOS *et al.*, 2020).

5.2.4 Cross-domain generalization rate

A more detailed analysis of the SN and SRN networks can be performed to measure domain generalization with CDFG Measure (see section 4.1). Tables 26 and 27 present the Partial CDFG Measure for supervised and unsupervised approaches for digit images domain, respectively. In both paradigms of learning, SN is more adaptable among these datasets than SRNs. The divergence of feature TL is greater in supervised learning due to the low transfer from MNIST to SVHN. Despite the overall superiority of SN, SRN-1 overcomes all results when the network training is performed with MNIST to be tested with USPS. Consequently, the Partial CDFG behavior reflects the Complete CDFG (described in Table 28), reinforcing that SN as the network with highest rates of feature TL.

Table 26 – Partial CDFG Measure on supervised feature TL (%) from source training dataset (S) to a target test dataset (T). From (SANTOS *et al.*, 2020).

Model	**S: MNIST** **T: SVHN**	**S: SVHN** **T: MNIST**	**S: MNIST** **T: USPS**	**S: USPS** **T: MNIST**	**Avg.**
SN	**7.89**	**0.89**	6.08	**2.79**	**4.41**
SRN-1	41.74	10.13	**1.69**	3.61	14.3
SRN-2	48.28	6.68	1.95	3.52	15.11
SRN-4	46.86	5.18	3.98	4.87	15.22

Table 27 – Partial CDFG Measure on unsupervised feature TL (%) from source training dataset (S) to a target test dataset (T). From (SANTOS *et al.*, 2020).

Model	**S: MNIST** **T: SVHN**	**S: SVHN** **T: MNIST**	**S: MNIST** **T: USPS**	**S: USPS** **T: MNIST**	**Avg.**
SN	**1.76**	**0.18**	4.53	**0.27**	**1.68**
SRN-1	10.52	0.37	**1.05**	5.47	4.35
SRN-2	5.39	1.84	5.28	6.56	4.77
SRN-4	12.02	5.35	12.11	10.01	9.87

Table 28 – Complete CDFG Measure on feature TL (%). From (SANTOS *et al.*, 2020).

Model	**MNIST and SVHN**		**MNIST and USPS**		**Avg**
	Superv.	**Unsuperv.**	**Superv.**	**Unsuperv.**	
SN	**4.39**	**0.97**	4.43	**2.4**	**3.05**
SRN-1	25.93	5.44	**2.65**	3.26	9.32
SRN-2	27.48	3.61	2.73	5.92	9.93
SRN-4	26.02	8.68	4.42	11.06	12.54

The semi-supervised domain generalization results are presented in Figure 26 for Complete CDFG Measure. As expected, due to greater similarity, the domain generalization between MNIST and USPS is highest than for MNIST and SVHN. Furthermore, SN generalizes better than all SRNs, repeating the same behavior from supervised and unsupervised approaches. Observing only the Partial CDFG, SVHN offers more learning to MNIST than the reverse direction. Among the SRNs, domain generalization is improved with the addition of more residual blocks. Consequently, these results corroborate the previous one, in which restricted AE have better potential for cross-domain feature representation (CAVALLARI; RIBEIRO; PONTI, 2018).

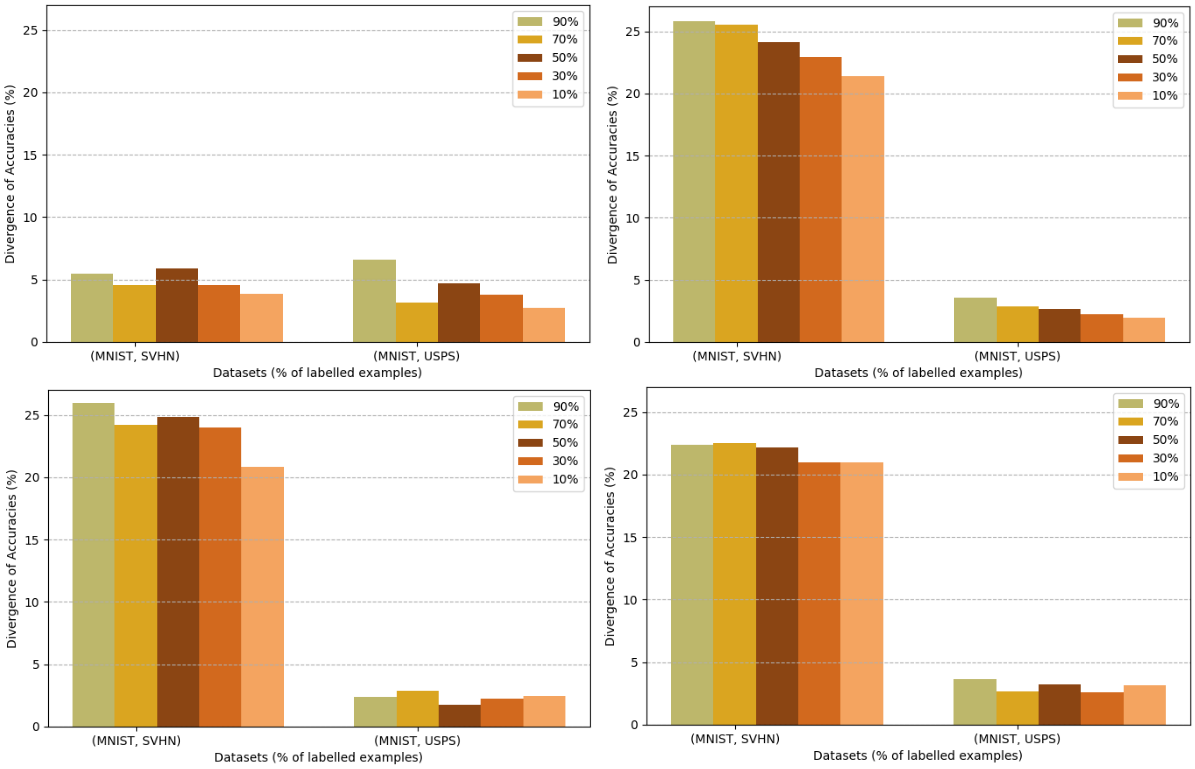

Figure 26 – Complete CDFG Measure for semi-supervised classification: (top-left) SN; (top-right) SRN-1; (bottom-left) SRN-2; and (bottom-right) SRN-4. The percentage indicates the labeled examples rate during the network training. From (SANTOS *et al.*, 2020).

5.3 Final comments on semi-supervised learning

In general, the designed architectures presented high performances than competing methods, which apply more complex structures than only deep model training and feature extraction, including in cross-dataset scenarios. In semi-supervised networks, we used the unlabeled data to initialize network parameters, increasing the likelihood to incorporate more source domain features. As described in the experiments performed with WLL, few labeled instances provided a considerable increase in the performance of predictive models, in which all available data contain important information, labeled or unlabeled. As expected, the classification accuracy decreases as the proportion of labeled data to train the network, especially for feature TL scenarios. Despite of that, WLL offers as lower decrease. Additionally, semi-supervised performances can even surpass scenarios with fully labeled data.

Evidently, the learning transferability from a model is influenced by the similarity between the involved domains (source and target), measured using CDFG Measure. Since WLL loss function is dependent only on the proportion of labeled examples, it is adaptable to different scenarios, especially in tasks involving transfer learning, with low computational complexity.

CHAPTER 6

CONCLUSION

This chapter reports the contributions and publications generated from the studies carried out during this research project, as well as the conclusions and future work.

6.1 Contributions

The first significant contribution was the analysis of features spaces that can be obtained and combined using pre-trained supervised deep learning networks. Many previous studies investigated only the pre-prediction layer, however our results offer guidelines for the scientific community to delve into initial and inner layers that may provide better descriptors for different tasks. Another contribution of this research was the CDFG Measure. The comparison among TL methodologies used to be performed only by specific metrics, which involved an analysis within each dataset and/or task. The proposed measure offers a way to estimate how data from one domain can offer a better model to be transferred to another. Additionally, such measure is independent of the desired task and classic metrics. Another important contribution was the Weighted Label Loss, which can be applied in deep hybrid networks, where supervised and unsupervised learning occurs simultaneously. The WLL balancing provides the learning from the unlabeled training set, in which is extremely relevant to the network. Otherwise, these unlabeled examples would not be considered to provide knowledge for the model.

Hence, the exploration of the initial and inner layers from supervised deep networks, either by analysis of the feature spaces (see section 3.1) or by fusion of the multi-layers activation maps (see section 3.2), highlights the affirmation contained in the first hypothesis: ***"different inner layers of supervised deep networks should be considered, and potentially combined, when obtaining feature spaces in order to improve image and video recognition in transfer learning scenarios"***. This hypothesis was validated using different image domains with feature extraction, network fine-tuning, and alignment of feature spaces obtained by TCA. Therefore, in situations of domain generalization involving supervised networks, it is essential that the initial and inner layers are analyzed for the model to be more robust.

The second hypothesis ***"the descriptive capacity of a model to transfer the acquired learning should be measured by metrics of each task and by different levels of divergence"*** was validated by two different models: anomaly detection in videos considering CNN feature extraction and alignment of the obtained spaces (detailed in section 4.2); and by classifying images in a hybrid network involving both supervised and unsupervised learning simultaneously, described in section 5.2. As can be observed by the results obtained through the proposed generalization measure (see section 4.1), the models guarantee greater reliability in the use of the acquired learning, indicating which domains are more appropriate in each case. Clearly, without this measure, classical metrics only assess whether the model fits a specific dataset.

For the third hypothesis ***"in partially labeled data transfer learning scenarios, labeled and unlabeled examples should be used jointly to increase the performance of the feature space"***, a hybrid architecture was developed that incorporates all available data, labeled or not, in different forms of training and individual loss functions (see section 5.2). In this architecture, using WLL (detailed in section 5.1) as the main loss function, we can note that small labeled data rates significantly improve prediction over unsupervised evaluation. Additionally, in some cases, semi-supervised results outperform supervised accuracy in the same scenario.

Therefore, with the validation of the three specific hypotheses, we restate the validity of the general hypothesis: ***"deep networks for feature transfer learning tasks should be properly analyzed at different hierarchical levels of representations and paradigms of learning, considering both classical and generalization measures"***.

6.2 Research published studies

This section describes the publications directly related to this research theme that were produced in the course of the project, involving features TL between different domains, either in images or videos:

- Santos, F. P.; Ponti, M. A. "Robust feature spaces obtained from pre-trained deep network layers for skin lesion classification" (SANTOS; PONTI, 2018) in the 31th Conference on Graphics, Patterns and Images (SIBGRAPI-2018). This study analyzes feature spaces extracted from pre-trained CNN (with and without fine-tuning) and distortion behavior applied to them, potentializing their use for feature TL (see methodology and results in section 3.1). The paper contribution includes: (i) use of several CNN models and different layers for feature extraction and skin lesions image classification with and without fine-tuning; (ii) a detailed study of the impact of dimensionality reduction, colors space contraction, and noisy effects in the feature space; and (iii) feature generalization analysis between raw and distorted sets;

- Santos, F. P.; Ponti, M. A. "Alignment of local and global features from multiple layers of convolutional neural network for image classification" (SANTOS; PONTI, 2019a) in the 32th Conference on Graphics, Patterns and Images (SIBGRAPI-2019). In this model was analyzed features fusion from multi-layers of a CNN, where the resulting latent space is aligned by TCA for different domains (see details of the method and results in section 3.2). The contribution includes: (i) a novel that aggregates multi-layer features fusion from a CNN and manifold alignment for image classification; (ii) practical evidence that multi-layer features fusion provides better performance for feature TL in low-level appearance datasets; and (iii) extensive experimentation in different scenarios of images;

- Santos, F. P.; Ribeiro, L. S. F.; Ponti, M. A. "Generalization of feature embeddings transferred from different video anomaly detection domains" (SANTOS; RIBEIRO; PONTI, 2019) in the Journal of Visual Communication and Image Representation (2019). This paper proposes generalization measures to prove the efficiency of a cross-domain methods using anomaly detection task in videos (detailed in section 4.1 and experiments in section 4.2). The contributions from this paper were: (i) a framework for feature TL applied in the task of anomaly detection in videos; and (ii) a novel evaluation approach regarding generalization of feature embedding;

- Santos, F. P.; Zor, C.; Kittler, J.; Ponti, M. A. "Learning image features with fewer labels using a semi-supervised deep convolutional network" (SANTOS *et al.*, 2020). This study proposed a hybrid architecture of CNN and AE with a differentiated loss function by the adoption of the labels rate provided to balance individual loss (described in section 5.1 and experiments in section 5.2). Consequently, the contributions were: (i) a semi-supervised architecture which relates supervised and unsupervised learning simultaneously, allowing training with unlabeled data; (ii) a dynamic loss function that weights the amount of labeled examples provided during training; and (iii) an in-depth analysis showing that the method allows good transfer of feature embedding learning between different datasets.

6.3 Related published studies

In addition to the research theme publications, this section reports the studies produced in different contexts from image processing, such as relevant sampling and smoothing:

- Ponti, M. A.; Costa, G. B. P.; Santos, F. P.; Silveira, K. U. "Supervised and unsupervised relevance sampling in handcrafted and deep learning features obtained from image collections" (PONTI *et al.*, 2019) in the Applied Soft Computing Journal (2019). This paper investigated relevant samples selection from images collections to provide learning to classifiers. Using both supervised and unsupervised methods, handcrafted and DL features were generated to Optimum-Path Forest Selection, SVM-Selection, and k-Medoids Centroid Selection. Each of these three methods determine $\sqrt{N} \cdot C$ samples from training sets, which N is the set size and C is the number of classes. Experiments showed potential to reduce the original training set size to 30% with a small accuracy decrease;

- Santos, F. P.; Ponti, M. A. "Homogeneity index as stopping criterion for anisotropic diffusion filter" (SANTOS; PONTI, 2019b) in the 18th International Conference on Computer Analysis of Images and Patterns (CAIP-2019). In this study, it was investigated a new stopping criterion for the Anisotropic Diffusion Filter. Due to the costly iterations, this method considers the image homogeneity and the existing parameters to establish the optimal smoothing time close to ideal. Experiments show a significant improvement in the reduction of the number of iterations (approximately 78% in the images considered) without losing quality from the resulting image.

6.4 Final considerations and future directions

Specifically to the objectives outlined at the beginning of this research, the developed predictive models and techniques sought to obtain discriminative, compact, and generalizable features spaces. To achieve this purpose, we investigated how to apply network fine-tuning and manifold alignment methods to obtain representations of image and video datasets and integrate supervised and unsupervised learning into a hybrid architecture. Additionally, due to the gap in the previous literature, we proposed a generalization metric to evaluate the transfer ability of methods considering a source dataset. The built architectures and analyzes were validated considering the classic metrics of each task (accuracy for classification; and AUC and EER for anomaly detection) and CDFG Measure for generalization purposes. Additionally, our predictive models were experimented by different datasets in the same context: fruits, objects, skin lesions, natural photos, and digits for images; and surveillance videos of urban and natural scenarios. Our studies provided important guidelines for feature extraction in pre-trained CNNs and deep investigation of initial and inner layers, assessing which datasets increase learning guarantees, even in semi-supervised scenarios where WLL can be used with different loss functions in any hybrid architecture.

The results and discussions presented makes it evident how challenging this theme presents itself to researchers. While we have state-of-the-art architectures for classification and anomaly detection tasks, we often face a lack of labeled data availability. For this particularity, as we can note, semi-supervised architectures provide an interesting step-forward in which the WLL loss function contributes significantly in partially labeled data scenarios. Intuitively, this loss function does not solve the issue of data labeling cost, however, it allows unlabeled data to be leveraged in predictive models with acceptable performance. Another relevant aspect is to be able to find manners to deal with differences within the same domain, such as spatial projection, noise, and different forms of acquisition. In this case, CDFG Measure is an innovative divergence metric that allows one to detect whether one data distribution is sufficiently compatible with another to provide knowledge that will help the model achieve better performance. Consequently, if a model is developed to reduce spatial projection differences, for example, CDFG Measure will express the quality of this methodology quantitatively. Another topic is the representativeness of each layer to provide feature maps. Initial layers are known to have greater descriptive capability for low-level features and the latter ones incorporate texture and semantics. Because of this hierarchical structure, early and middle layers were in the background with descriptors. However, it is interesting how these layers can help to improve performance when used properly. Of course this incorporation comes with additional computational costs, however the performance gain can be significant, which implies the advantage especially in critical systems, such as imaging diagnostics, where response time need not be immediate and accuracy is highly required. For feature TL tasks we can observe how these items complement each other in the same scenario: we can train a predictive model with partially labeled data, exploring multiple layers and evaluating its generalizability ability.

Based on the motivation to delve deep into different manners of finding a single and robust solution for the same task and domain, respecting the particularities of each scenario, to build generalizable models provide a realistic path to deploying real systems. Therefore, this study suggests new research questions to be explored in future work. For example, considering the CDFG Measure we can observe that it relates only two data distributions at a time. In an opportunity to move forward in this matter, one requirement is its expansion into *N* domains while maintaining its theoretical foundation. Consequently, this assessment may predict how well data fusion provides learning for a target dataset. Another relevant unanswered question is how to adopt temporal units in deep predictive models. In this scenario, several configurations are possible: incorporation of temporal layers in conventional CNNs, development of multi-stream networks in which to extract parallel and immersive visual and movement features, and exploration of new loss functions that relate these layers in pre-trained networks. Finally, novel architectures and training strategies to improve the use of both labeled and unlabeled examples are still to be investigated.

BIBLIOGRAPHY

BAGHERINEZHAD, H.; RASTEGARI, M.; FARHADI, A. Lcnn: Lookup-based convolutional neural network. In: **Proc. IEEE CVPR**. [S.l.: s.n.], 2017. Citations on pages 34, 36 e 45.

BAHETI, B.; GAJRE, S.; TALBAR, S. Detection of distracted driver using convolutional neural network. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2018. p. 1032–1038. Citations on pages 26 e 27.

BALDI, P.; HORNIK, K. Neural networks and principal component analysis: Learning from examples without local minima. **Neural networks**, Elsevier, v. 2, n. 1, p. 53–58, 1989. Citation on page 38.

BARATA, C.; CELEBI, M. E.; MARQUES, J. S. Improving dermoscopy image classification using color constancy. **IEEE journal of biomedical and health informatics**, IEEE, v. 19, n. 3, p. 1146–1152, 2015. Citation on page 50.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 35, n. 8, p. 1798–1828, 2013. Citations on pages 31 e 35.

BI, L.; KIM, J.; AHN, E.; FENG, D.; FULHAM, M. Automatic melanoma detection via multi-scale lesion-biased representation and joint reverse classification. In: IEEE. **2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)**. [S.l.], 2016. p. 1055–1058. Citation on page 50.

CAVALLARI, G.; RIBEIRO, L.; PONTI, M. Unsupervised representation learning using convolutional and stacked auto-encoders: A domain and cross-domain feature space analysis. In: IEEE. **2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2018. p. 440–446. Citation on page 83.

CHAKER, R.; AGHBARI, Z. A.; JUNEJO, I. N. Social network model for crowd anomaly detection and localization. **Pattern Recognition**, Elsevier, v. 61, p. 266–281, 2017. Citation on page 59.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, ACM, v. 41, n. 3, p. 15, 2009. Citation on page 59.

CHEN, Q.; LIU, Y.; WANG, Z.; WASSELL, I.; CHETTY, K. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 7976–7985. Citation on page 75.

CHEN, Y.; DUFFNER, S.; STOIAN, A.; DUFOUR, J.-Y.; BASKURT, A. Pedestrian attribute recognition with part-based cnn and combined feature representations. In: **VISAPP2018**. [S.l.: s.n.], 2018. Citation on page 50.

CHEN, Y.; ZHOU, X. S.; HUANG, T. S. One-class svm for learning in image retrieval. In: IEEE. **Image Processing, 2001. Proceedings. 2001 International Conference on**. [S.l.], 2001. v. 1, p. 34–37. Citation on page 63.

COATES, A.; NG, A.; LEE, H. An analysis of single-layer networks in unsupervised feature learning. In: **Proceedings of the fourteenth international conference on artificial intelligence and statistics**. [S.l.: s.n.], 2011. p. 215–223. Citation on page 73.

CUI, Z.; CHANG, H.; SHAN, S.; CHEN, X. Generalized unsupervised manifold alignment. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2014. p. 2429–2437. Citation on page 39.

DENG, L.; YU, D. *et al.* Deep learning: methods and applications. **Foundations and Trends® in Signal Processing**, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014. Citation on page 32.

DILOKTHANAKUL, N.; MEDIANO, P. A.; GARNELO, M.; LEE, M. C.; SALIMBENI, H.; ARULKUMARAN, K.; SHANAHAN, M. Deep unsupervised clustering with gaussian mixture variational autoencoders. **arXiv preprint arXiv:1611.02648**, 2016. Citation on page 74.

DUAN, L.; XU, D.; TSANG, I. W.-H.; LUO, J. Visual event recognition in videos by learning from web data. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 34, n. 9, p. 1667–1680, 2012. Citation on page 32.

EPAILLARD, E.; BOUGUILA, N. Proportional data modeling with hidden markov models based on generalized dirichlet and beta-liouville mixtures applied to anomaly detection in public areas. **Pattern Recognition**, Elsevier, v. 55, p. 125–136, 2016. Citation on page 59.

FEI, J.; RUI, T.; SONG, X.; ZHOU, Y.; ZHANG, S. More discriminative convolutional neural network with inter-class constraint for classification. **Computers & Electrical Engineering**, Elsevier, v. 68, p. 484–489, 2018. Citation on page 74.

GE, Z.; DEMYANOV, S.; BOZORGTABAR, B.; ABEDINI, M.; CHAKRAVORTY, R.; BOWLING, A.; GARNAVI, R. Exploiting local and generic features for accurate skin lesions classification using clinical and dermoscopy imaging. In: IEEE. **2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)**. [S.l.], 2017. p. 986–990. Citation on page 50.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. Deep learning. MIT Press. 2016. Citations on pages 23, 31 e 32.

GOYAL, P.; DOLLÁR, P.; GIRSHICK, R.; NOORDHUIS, P.; WESOLOWSKI, L.; KYROLA, A.; TULLOCH, A.; JIA, Y.; HE, K. Accurate, large minibatch sgd: training imagenet in 1 hour. **arXiv preprint arXiv:1706.02677**, 2017. Citation on page 38.

GUO, Y.; LIU, Y.; OERLEMANS, A.; LAO, S.; WU, S.; LEW, M. S. Deep learning for visual understanding: A review. **Neurocomputing**, Elsevier, v. 187, p. 27–48, 2016. Citations on pages 32, 34 e 59.

HAO, T.; WU, D.; WANG, Q.; SUN, J.-S. Multi-view representation learning for multi-view action recognition. **Journal of Visual Communication and Image Representation**, Elsevier, v. 48, p. 453–460, 2017. Citation on page 59.

HASAN, M.; CHOI, J.; NEUMANN, J.; ROY-CHOWDHURY, A. K.; DAVIS, L. S. Learning temporal regularity in video sequences. In: IEEE. **Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on**. [S.l.], 2016. p. 733–742. Citation on page 59.

HAYKIN, S. Neural networks: principles and practice. **Bookman**, 2001. Citations on pages 31, 33 e 36.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. Citations on pages 34, 44, 50 e 52.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Citations on pages 34, 36 e 44.

HU, J.; LU, J.; TAN, Y.-P. Deep transfer metric learning. In: IEEE. **Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on**. [S.l.], 2015. p. 325–333. Citation on page 24.

HU, Y.; CHANG, H.; NIAN, F.; WANG, Y.; LI, T. Dense crowd counting from still images with convolutional neural networks. **Journal of Visual Communication and Image Representation**, Elsevier, v. 38, p. 530–539, 2016. Citation on page 59.

JIANG, F.; WU, Y.; KATSAGGELOS, A. K. Detecting contextual anomalies of crowd motion in surveillance video. In: IEEE. **Image Processing (ICIP), 2009 16th IEEE International Conference on**. [S.l.], 2009. p. 1117–1120. Citation on page 59.

JODOIN, P.-M.; KONRAD, J.; SALIGRAMA, V. Modeling background activity for behavior subtraction. In: IEEE. **Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on**. [S.l.], 2008. p. 1–10. Citation on page 63.

JOLLIFFE, I. T. Principal component analysis and factor analysis. In: **Principal component analysis**. [S.l.]: Springer, 1986. p. 115–128. Citations on pages 38, 44 e 62.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014. Citation on page 72.

KORNBLITH, S.; SHLENS, J.; LE, Q. V. Do better imagenet models transfer better? In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 2661–2671. Citation on page 26.

KOUW, W. M.; LOOG, M. A review of single-source unsupervised domain adaptation. **arXiv preprint arXiv:1901.05335**, 2019. Citation on page 39.

KRIZHEVSKY, A.; HINTON, G. **Learning multiple layers of features from tiny images**. [S.l.], 2009. Citation on page 73.

KUZNIETSOV, Y.; STÜCKLER, J.; LEIBE, B. Semi-supervised deep learning for monocular depth map prediction. In: IEEE. **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.], 2017. p. 2215–2223. Citation on page 69.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citation on page 73.

LI, M.; ZHANG, T.; CHEN, Y.; SMOLA, A. J. Efficient mini-batch training for stochastic optimization. In: ACM. **Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2014. p. 661–670. Citation on page 38.

LI, S.; SONG, S.; HUANG, G.; DING, Z.; WU, C. Domain invariant and class discriminative feature learning for visual domain adaptation. **IEEE Transactions on Image Processing**, IEEE, v. 27, n. 9, p. 4260–4273, 2018. Citation on page 76.

LIN, T.-Y.; ROYCHOWDHURY, A.; MAJI, S. Bilinear cnn models for fine-grained visual recognition. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1449–1457. Citation on page 43.

LING, Z.; LI, X.; ZOU, W.; GUO, S. Semi-supervised learning via convolutional neural network for hyperspectral image classification. In: IEEE. **2018 24th International Conference on Pattern Recognition (ICPR)**. [S.l.], 2018. p. 1–6. Citations on pages 69 e 70.

LONG, M.; CAO, Y.; WANG, J.; JORDAN, M. Learning transferable features with deep adaptation networks. In: JMLR. **Journal of Machine Learning Research, 2015. 32th International Conference on Machine Learning**. [S.l.], 2015. Citation on page 31.

LU, J.; BEHBOOD, V.; HAO, P.; ZUO, H.; XUE, S.; ZHANG, G. Transfer learning using computational intelligence: a survey. **Knowledge-Based Systems**, Elsevier, v. 80, p. 14–23, 2015. Citations on pages 23, 29 e 32.

LUXBURG, U. von; SCHÖLKOPF, B. Statistical learning theory: Models, concepts, and results. In: MAX-PLANCK-GESELLSCHAFT. **Handbook of the History of Logic, Vol. 10: Inductive Logic**. Amsterdam, Netherlands: Elsevier North Holland, 2011. v. 10, p. 651–706. Citation on page 60.

MAHADEVAN, V.; LI, W.; BHALODIA, V.; VASCONCELOS, N. Anomaly detection in crowded scenes. In: IEEE. **Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on**. [S.l.], 2010. p. 1975–1981. Citation on page 63.

MAHBOD, A.; ECKER, R.; ELLINGER, I. Skin lesion classification using hybrid deep neural networks. **arXiv preprint arXiv:1702.08434**, 2017. Citation on page 43.

MAJTNER, T.; YILDIRIM-YAYILGAN, S.; HARDEBERG, J. Y. Combining deep learning and hand-crafted features for skin lesion classification. In: IEEE. **Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on**. [S.l.], 2016. p. 1–6. Citation on page 43.

MASCI, J.; MEIER, U.; CIREŞAN, D.; SCHMIDHUBER, J. Stacked convolutional auto-encoders for hierarchical feature extraction. **Artificial Neural Networks and Machine Learning–ICANN 2011**, Springer, p. 52–59, 2011. Citation on page 35.

MASTERS, D.; LUSCHI, C. Revisiting small batch training for deep neural networks. **arXiv preprint arXiv:1804.07612**, 2018. Citations on pages 38 e 72.

MELLO, R. F.; PONTI, M. A. **Machine Learning: A Practical Approach on the Statistical Learning Theory**. [S.l.]: Springer, 2018. Citations on pages 26 e 60.

MELLO, R. F. de; FERREIRA, M. D.; PONTI, M. A. Providing theoretical learning guarantees to deep learning networks. **arXiv preprint arXiv:1711.10292**, 2017. Citations on pages 28 e 32.

MENDONÇA, T.; FERREIRA, P. M.; MARQUES, J. S.; MARCAL, A. R.; ROZEIRA, J. Ph 2-a dermoscopic image database for research and benchmarking. In: IEEE. **2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)**. [S.l.], 2013. p. 5437–5440. Citations on pages 44, 45 e 53.

MISHKIN, D.; SERGIEVSKIY, N.; MATAS, J. Systematic evaluation of convolution neural network advances on the imagenet. **Computer Vision and Image Understanding**, Elsevier, v. 161, p. 11–19, 2017. Citations on pages 32 e 43.

MUREŞAN, H.; OLTEAN, M. Fruit recognition from images using deep learning. **Acta Universitatis Sapientiae, Informatica**, Sciendo, v. 10, n. 1, p. 26–42, 2018. Citation on page 52.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814. Citation on page 32.

NAZARE, T.; COSTA, G. P. da; CONTATO, W.; PONTI, M. A. Deep convolutional neural networks and noisy images. In: **Iberoamerican Conference on Pattern Recognition (CIARP 2017)**. [S.l.: s.n.], 2017. LNCS 10657. Citation on page 34.

NETZER, Y.; WANG, T.; COATES, A.; BISSACCO, A.; WU, B.; NG, A. Y. Reading digits in natural images with unsupervised feature learning. In: **NIPS workshop on deep learning and unsupervised feature learning**. [S.l.: s.n.], 2011. v. 2011, p. 5. Citation on page 73.

PAN, S. J.; KWOK, J. T.; YANG, Q. Transfer learning via dimensionality reduction. In: **AAAI**. [S.l.: s.n.], 2008. v. 8, p. 677–682. Citation on page 40.

PAN, S. J.; TSANG, I. W.; KWOK, J. T.; YANG, Q. Domain adaptation via transfer component analysis. **IEEE Transactions on Neural Networks**, IEEE, v. 22, n. 2, p. 199–210, 2011. Citations on pages 39, 41, 50 e 62.

PAN, S. J.; YANG, Q. *et al.* A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, Institute of Electrical and Electronics Engineers, Inc., 345 E. 47 th St. NY NY 10017-2394 USA, v. 22, n. 10, p. 1345–1359, 2010. Citations on pages 23, 24, 27, 31 e 67.

PAN, Y.; YAO, T.; LI, H.; MEI, T. Video captioning with transferred semantic attributes. In: **CVPR**. [S.l.: s.n.], 2017. v. 2, p. 3. Citation on page 43.

PATHIRAGE, C. S. N.; LI, J.; LI, L.; HAO, H.; LIU, W.; NI, P. Structural damage identification based on autoencoder neural networks and deep learning. **Engineering Structures**, Elsevier, v. 172, p. 13–28, 2018. Citation on page 35.

PAUL, A.; MAJUMDAR, A.; MUKHERJEE, D. P. Discriminative autoencoder. In: IEEE. **2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2018. p. 3049–3053. Citation on page 74.

POMPONIU, V.; NEJATI, H.; CHEUNG, N.-M. Deepmole: Deep neural networks for skin mole lesion classification. In: IEEE. **Image Processing (ICIP), 2016 IEEE International Conference on**. [S.l.], 2016. p. 2623–2627. Citation on page 43.

PONTI, M.; NAZARÉ, T. S.; THUMÉ, G. S. Image quantization as a dimensionality reduction procedure in color and texture feature extraction. **Neurocomputing**, Elsevier, v. 173, p. 385–396, 2016. Citation on page 47.

PONTI, M.; RIBEIRO, L. S.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: **30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T 2017)**. [S.l.: s.n.], 2017. p. 17–41. Citations on pages 24, 25, 26, 32, 34, 35, 36, 37, 38 e 59.

PONTI, M. A.; COSTA, G. B. P. da; SANTOS, F. P.; SILVEIRA, K. U. Supervised and unsupervised relevance sampling in handcrafted and deep learning features obtained from image collections. **Applied Soft Computing**, Elsevier, v. 80, p. 414–424, 2019. Citation on page 87.

RAVISHANKAR, H.; SUDHAKAR, P.; VENKATARAMANI, R.; THIRUVENKADAM, S.; ANNANGI, P.; BABU, N.; VAIDYA, V. Understanding the mechanisms of deep transfer learning for medical images. In: **Deep Learning and Data Labeling for Medical Applications**. [S.l.]: Springer, 2016. p. 188–196. Citation on page 32.

RAZAVIAN, A. S.; AZIZPOUR, H.; SULLIVAN, J.; CARLSSON, S. Cnn features off-the-shelf: an astounding baseline for recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2014. p. 806–813. Citations on pages 32 e 43.

REN, Y.; HU, K.; DAI, X.; PAN, L.; HOI, S. C.; XU, Z. Semi-supervised deep embedded clustering. **Neurocomputing**, Elsevier, v. 325, p. 121–130, 2019. Citations on pages 29, 69, 79 e 81.

RIBANI, R.; MARENGONI, M. A survey of transfer learning for convolutional neural networks. In: **32th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T 2019)**. [S.l.: s.n.], 2019. p. 47–57. Citation on page 24.

ROCHA, A.; HAUAGGE, D. C.; WAINER, J.; GOLDENSTEIN, S. Automatic produce classification from images using color, texture and appearance cues. In: IEEE. **2008 XXI Brazilian Symposium on Computer Graphics and Image Processing**. [S.l.], 2008. p. 3–10. Citation on page 52.

ROSHTKHARI, M. J.; LEVINE, M. D. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. **Computer vision and image understanding**, Elsevier, v. 117, n. 10, p. 1436–1452, 2013. Citation on page 59.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M. *et al.* Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, n. 3, p. 211–252, 2015. Citations on pages 36, 44, 45 e 62.

SADIGH, S.; SEN, P. Improving the resolution of cnn feature maps efficiently with multisampling. **arXiv preprint arXiv:1805.10766**, 2018. Citations on pages 26, 27 e 43.

SAENKO, K.; KULIS, B.; FRITZ, M.; DARRELL, T. Adapting visual category models to new domains. In: SPRINGER. **European conference on computer vision**. [S.l.], 2010. p. 213–226. Citation on page 53.

SAITO, K.; USHIKU, Y.; HARADA, T. Asymmetric tri-training for unsupervised domain adaptation. In: JMLR. ORG. **Proceedings of the 34th International Conference on Machine Learning-Volume 70**. [S.l.], 2017. p. 2988–2997. Citation on page 76.

SALIDO, J.; JR, C. R. Using deep learning for melanoma detection in dermoscopy images. **International Journal of Machine Learning and Computing**, v. 8, n. 1, p. 61–68, 2018. Citation on page 50.

SANTOS, F. P. dos; PONTI, M. A. Robust feature spaces from pre-trained deep network layers for skin lesion classification. In: IEEE. **2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2018. p. 189–196. Citations on pages 26, 27, 32, 34, 43, 44, 46, 47, 48, 49, 50 e 86.

____. Alignment of local and global features from multiple layers of convolutional neural network for image classification. In: IEEE. **2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2019. p. 241–248. Citations on pages 26, 27, 51, 52, 54, 55, 56 e 87.

____. Homogeneity index as stopping criterion for anisotropic diffusion filter. In: SPRINGER. **International Conference on Computer Analysis of Images and Patterns**. [S.l.], 2019. p. 269–280. Citation on page 88.

SANTOS, F. P. dos; RIBEIRO, L. S.; PONTI, M. A. Generalization of feature embeddings transferred from different video anomaly detection domains. **Journal of Visual Communication and Image Representation**, Elsevier, v. 60, p. 407–416, 2019. Citations on pages 25, 26, 27, 31, 41, 43, 62, 63, 64, 65, 66, 67, 68 e 87.

SANTOS, F. P. dos; ZOR, C.; KITTLER, J.; PONTI, M. A. Learning image features with fewer labels using a semi-supervised deep convolutional network. **Under review**, 2020. Citations on pages 26, 27, 37, 70, 72, 73, 75, 76, 77, 78, 79, 80, 81, 82, 83 e 87.

SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. **Neural computation**, MIT Press, v. 10, n. 5, p. 1299–1319, 1998. Citation on page 40.

SENER, O.; SONG, H. O.; SAXENA, A.; SAVARESE, S. Learning transferrable representations for unsupervised domain adaptation. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2016. p. 2110–2118. Citation on page 76.

SENGUPTA, B.; FRISTON, K. J. How robust are deep neural networks? **arXiv preprint arXiv:1804.11313**, 2018. Citations on pages 26 e 29.

SHAO, L.; ZHU, F.; LI, X. Transfer learning for visual categorization: A survey. **IEEE transactions on neural networks and learning systems**, IEEE, v. 26, n. 5, p. 1019–1034, 2015. Citations on pages 23, 29, 31 e 69.

SHI, Z.; HAO, H.; ZHAO, M.; FENG, Y.; HE, L.; WANG, Y.; SUZUKI, K. A deep cnn based transfer learning method for false positive reduction. **Multimedia Tools and Applications**, Springer, p. 1–17, 2018. Citation on page 26.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014. Citations on pages 34, 44 e 62.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citations on pages 36 e 69.

TORREY, L.; SHAVLIK, J. Transfer learning. In: **Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques**. [S.l.]: IGI Global, 2010. p. 242–264. Citations on pages 24 e 67.

TSCHANDL, P.; ROSENDAHL, C.; KITTLER, H. The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. **arXiv preprint arXiv:1803.10417**, 2018. Citations on pages 46 e 53.

TZENG, E.; HOFFMAN, J.; DARRELL, T.; SAENKO, K. Simultaneous deep transfer across domains and tasks. In: IEEE. **Computer Vision (ICCV), 2015 IEEE International Conference on**. [S.l.], 2015. p. 4068–4076. Citation on page 24.

TZENG, E.; HOFFMAN, J.; SAENKO, K.; DARRELL, T. Adversarial discriminative domain adaptation. In: **Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017. v. 1, p. 4. Citations on pages 72 e 75.

TZOREFF, E.; KOGAN, O.; CHOUKROUN, Y. Deep discriminative latent space for clustering. **arXiv preprint arXiv:1805.10795**, 2018. Citation on page 74.

VAPNIK, V. N. An overview of statistical learning theory. **IEEE transactions on neural networks**, IEEE, v. 10, n. 5, p. 988–999, 1999. Citations on pages 45 e 60.

WANG, C.; KRAFFT, P.; MAHADEVAN, S. **Manifold alignment**. [S.l.]: CRC Press, 2011. Citation on page 39.

WANG, C.; MAHADEVAN, S. A general framework for manifold alignment. In: **AAAI fall symposium: manifold learning and its applications**. [S.l.: s.n.], 2009. p. 53–58. Citation on page 39.

WANG, J.; ZHENG, H.; HUANG, Y.; DING, X. Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, 2017. Citation on page 24.

WANG, J. Z.; LI, J.; WIEDERHOLD, G. Simplicity: Semantics-sensitive integrated matching for picture libraries. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, n. 9, p. 947–963, 2001. Citation on page 53.

WEN, H.; GE, S.; CHEN, S.; WANG, H.; SUN, L. Abnormal event detection via adaptive cascade dictionary learning. In: IEEE. **Image Processing (ICIP), 2015 IEEE International Conference on**. [S.l.], 2015. p. 847–851. Citation on page 59.

WEN, J.; LIU, R.; ZHENG, N.; ZHENG, Q.; GONG, Z.; YUAN, J. Exploiting local feature patterns for unsupervised domain adaptation. **arXiv preprint arXiv:1811.05042**, 2018. Citation on page 43.

WU, Z.; WANG, X.; JIANG, Y.-G.; YE, H.; XUE, X. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In: ACM. **Proceedings of the 23rd ACM international conference on Multimedia**. [S.l.], 2015. p. 461–470. Citation on page 24.

XIE, F.; FAN, H.; LI, Y.; JIANG, Z.; MENG, R.; BOVIK, A. Melanoma classification on dermoscopy images using a neural network ensemble model. **IEEE transactions on medical imaging**, IEEE, v. 36, n. 3, p. 849–858, 2017. Citation on page 43.

XIE, M.; JEAN, N.; BURKE, M.; LOBELL, D.; ERMON, S. Transfer learning from deep features for remote sensing and poverty mapping. In: AAAI. **Computer Vision and Pattern Recognition, 2016. 30th AAAI Conference on Artificial Intelligence.** [S.l.], 2016. Citation on page 31.

XU, D.; OUYANG, W.; RICCI, E.; WANG, X.; SEBE, N. Learning cross-modal deep representations for robust pedestrian detection. **arXiv preprint arXiv:1704.02431**, 2017. Citation on page 24.

YANG, J.; YANG, G. Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer. **Algorithms**, Multidisciplinary Digital Publishing Institute, v. 11, n. 3, p. 28, 2018. Citation on page 74.

YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. How transferable are features in deep neural networks? In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 3320–3328. Citations on pages 23, 24, 28, 33, 37 e 43.

YU, Q.; CHANG, X.; SONG, Y.-Z.; XIANG, T.; HOSPEDALES, T. M. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. **arXiv preprint arXiv:1711.08106**, 2017. Citation on page 50.

ZAHARESCU, A.; WILDES, R. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2010. p. 563–576. Citation on page 63.

ZHANG, J.; LI, W.; OGUNBONA, P. Joint geometrical and statistical alignment for visual domain adaptation. **arXiv preprint arXiv:1705.05498**, 2017. Citation on page 76.

ZHENG, Y.; HUANG, J.; CHEN, T.; OU, Y.; ZHOU, W. Cnn classification based on global and local features. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Real-Time Image Processing and Deep Learning 2019**. [S.l.], 2019. v. 10996, p. 109960G. Citation on page 50.

ZHUANG, N.; YAN, Y.; CHEN, S.; WANG, H.; SHEN, C. Multi-label learning based deep transfer neural network for facial attribute classification. **Pattern Recognition**, Elsevier, v. 80, p. 225–240, 2018. Citation on page 24.

ICMC USP
SÃO CARLOS