Features transfer learning for image and video recognition tasks

Fernando Pereira dos Santos[1] and Moacir Antonelli Ponti
Institute of Mathematical and Computer Sciences (ICMC)
University of São Paulo (USP), São Carlos, SP, Brazil
E-mails: fernando_persan@alumni.usp.br,ponti@usp.br

***Abstract*—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; how to transfer; and when to transfer. Hence, we address these questions through distinct learning paradigms, transfer learning techniques, and several datasets and tasks. Therefore, our contributions are: an analysis of multiple descriptors contained in supervised deep networks; a new generalization metric that can be applied to any model and evaluation system; and a new architecture with a loss function for semi-supervised deep networks, in which all available data provide the learning.**

I. Introduction

In recent years, machine learning has collaborated with computer vision, showing high performances in pattern recognition tasks. In particular, representation learning [1] allowed obtaining feature spaces tailored for particular applications, using data driven end-to-end approaches. Such methods rely heavily in availability of massive data annotation, which was an incentive for transfer learning (TL) methods [2].

Let S be large sample of data from a source domain X_s with labels Y_s. After training, we have an estimate of the joint probability function $P(X_s, Y_s)$. Then one wants to estimate the joint probability distribution of, another, target domain, i.e. $P(X_t, Y_t)$, for which we have a sample T that is smaller (in number of instances) than S. At first, in order to obtain transfer learning both functions are assumed to be well represented by the same feature space or share the same data distribution. However, this assumption is not always true in real-world applications. When the T differs from S, one may need to consider to fully reconstruct the original model from scratch. This approach can be expensive and sometimes impossible considering the size of T [2], in particular when considering the high human cost to collect and annotate large databases [3]. In this scenario, the possibility of reusing similar and large datasets would diminish efforts to collect and annotate new data [4]. For this purpose, TL leverage concepts already learned, for example as a classifier or detector, and apply those to facilitate the search of parameters for new classifiers or detectors [2]. If the source and the target datasets are sufficiently similar, e.g. their output space Y_s and Y_t is equivalent, or $P(X_s)$ does not diverge drastically from $P(X_t)$, the learned model has acceptable performance using either datasets S or T [5]. 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 [6]. Therefore, TL should be analyzed in three perspectives [7]: **what to transfer** by investigating the similarity between domains in which common peculiarities must be highlighted and discrepancies must be minimized; **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 to avoid negative transfer [4], that occurs when the acquired knowledge worsens the model performance [8].

Convolutional Neural Networks (CNNs) are currently widely explored for TL since such methods are able to represent general low-level and high-level image features [9]–[11]. In order to investigate TL with CNNs, two approaches were explored in this thesis: fine-tuning from a model pre-trained with a large dataset [11]; and manifold alignment in the feature space, such as the method of Transfer Component Analysis (TCA) [12], designed to strengthen relationships from different feature spaces into a new unified latent space by aligning underlying manifolds.

The meaning of TL also changes accordingly with the assigned task: in classification, the trained model should be sufficiently representative to allow distinguishing coexisting labels in both source and target domains, or allow to adapt the feature space for new labels at the target. When considering anomaly detection, TL should be used to enhance the similarity between normal and abnormal instances, so the main objective is to learn a common concept of normality. In both tasks, although the aim and meaning of the methods is different, there is a common underlying task: **to make sure the features extracted or learned from source data can transfer, as best as possible, to the target data**. In this thesis, we use the concept of generalization, which is a divergence measure of the error of some model considering the real distribution of the data (often unavailable and estimated via a test set) with respect to the sampled training data [13]. Hence, we advocate transfer learning should consider not only metrics of performance (such as errors and accuracies), but also with how the learned feature spaces generalize between source and target domains.

[1] Ph.D. Thesis

A. Thesis Contributions

We investigated how to leverage previously acquired knowledge and how to evaluate its generalization in image and video recognition tasks, in particular:

1) investigate **different layers of pre-trained CNNs** to obtain better transfer learning for skin lesion classification, a domain that differs from photographic content, including an analysis of model complexities and robustness (section II);
2) a **Cross-Domain Feature Space Generalization (CDFG) Measure**, allowing comparison of TL methodologies beyond performance metrics, estimating how well data from one domain transfers to another (section III);
3) a **semi-supervised deep network** combining classification and reconstruction tasks using a Weighted Label Loss (WLL), where supervised and unsupervised learning work together to improve representation learning for transfer learning scenarios, leveraging unlabeled examples that otherwise cannot not be used (section IV).

II. Feature TL using multiple CNN layers

Usually, models that involve CNN feature extraction only use layers that are very close to prediction output [14]–[16]. However, **does this adopted convention indicate that 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 [9]. Hence, we should not assume that only the end-layers provide representativeness. Contrarily, as the initial and inner layers offer low-level features, they may play an important role in the task.

A. End-layers features on skin lesion classification

Initially, considering only CNNs pre-trained by ImageNet dataset [17], we compare several end-layers regarding discriminative capacity, as well as impact caused by distortions applied to the best feature space obtained. Dimensionality reduction by PCA, color quantization, and noise injection were studied in this sense. The PH2 dataset [18] was employed for these experiments, which is widely used as a benchmark for skin lesion classification.

As shown in Table I, MobileNet provided more compact and discriminative feature spaces with 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, having its performance surpassed also by ResNet50 on average but with the cost of dimensionality: 100352 features. This corroborated evidence that smaller datasets for specific applications do not need complex networks. Hence, MobileNet (the lighter CNN) provides the best performance in accuracy, complexity, and dimensionality.

Since CNN layers often output high-dimensional feature maps, dimensionality reduction is an important projection to

TABLE I
PH2: 20-folds Cross Validation (CV) by Balanced Accuracy [19]. Layers are refer as -1 (the lastest), then -2 (one before the last), until -7.

CNN	Layer	Features	Variance	Linear SVM (%)
MobileNet [20]	-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 [21]	-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 [22]	-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

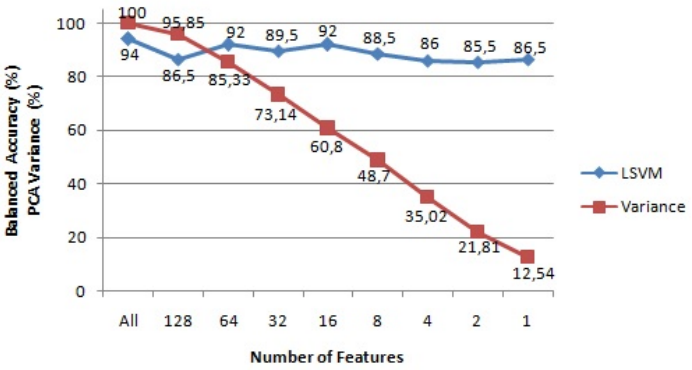

Fig. 1. Dimensionality reduction and variance by PCA [19].

show attributes relevance. The MobileNet layer -3 feature space (best performance) was gradually reduced from 128 to only 1 feature, halving the size each step. Therefore, as seen in Fig.1, it continues achieving high performance between 64 and 16 features (92% with 60.80% variance). Also, feature spaces quality are significantly impacted by color quantization [23]. To measure this influence, news sets were generated by computing 64, 32, and 16 colors per channel. As the color space contracted, Table II, performances become less linear, although not dramatically lower. Among noise injection, Salt & Pepper shows positive impact in small amounts, but negative with larger amounts.

Overall, MobileNet followed by a Linear SVM (94%) produces performances above competing methods, which comprise pre-processing steps to achieve at most 90.31% [24]. Based on these results, **it is notable the discriminative capacity contained in end-layers of CNNs**, being reinforced by results of dimensionality reduction and noise injection.

TABLE II
QUANTIZED AND NOISY: 20-FOLDS CV BY BALANCED ACCURACY [19]

Set	Linear SVM (%)
PH2 Quantization 64	94.5 ± 4.97
PH2 Quantization 32	92.5 ± 8.29
PH2 Quantization 16	90.0 ± 9.49
PH2 Gaussian 0.008	93.0 ± 7.81
PH2 Gaussian 0.016	93.0 ± 6.4
PH2 Gaussian 0.032	94.5 ± 7.4
PH2 Salt & Pepper 0.005	95.0 ± 6.71
PH2 Salt & Pepper 0.01	91.5 ± 9.1
PH2 Salt & Pepper 0.02	90.5 ± 8.65

B. Alignment of multi-layers features fusion

In addition to end-layers, we set out to leverage initial layers representativeness for image classification. Considering the pre-trained ResNet50 [22] and fine-tuning by a source domain, 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 with each local descriptor. Previously of fusion step, the local features passed on a process of selection due to larger amount of attributes. With the fusion features, the data distributions (source and target) are transformed to increase the correlation using TCA [12]. As result, the source is applied to SVM for training and the target for tests, as illustrated on Fig.2.

Due to the large number of attributes from local descriptors, three methods of feature selection was applied to choose which ones will compose the fusion maps. PCA is applied only to the source dataset, then the chosen components were applied to the target dataset. In Flatten Pooling (Flat.), the feature maps are fully converted from matrix to vector and a value $x = 100$ was adopted to split the vector into small symmetric segments, in which the average is calculated. For Pooling 2D is considered a square region (55×55) to calculate the average, where each region provides only one attribute. The variation in the number of attributes is suppressed due to TCA transformation, which defines the real amount for classification. For the experiments, the fine-tuning setup applied is the original ResNet50 training during 100 epochs and only the last seven layers were allowed to adapt with the new domain. This configuration offers a better observation of performances.

Table III presents results of three different image domains. This diversity is extremely important to emphasize the discriminative capacity of low-level descriptors, due to variation of styles, scene composition, and degree of task difficulty. Specifically to Fruits domain, multi-layer fusion structure are highly applicable. Individually, PCA has a higher accuracy in the first block ($\approx 41\%$), then the accuracy gradually decays while Flatten Pooling has a small better performance in the second block. 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 lower than fusion accuracies (bold values). For Objects domain is noticed a decrease in the performance of multi-layer fusion features in relation to Fruits domain. Despite this evidence, fusion features still offers significant improvement using Flatten Pooling in all residual blocks. PCA has better performance in the second block, however, worse than global performance 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 domain, 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, the fusion features do not increase the accuracy on average. A few multi-layer fusion results present slight superiority to global ones. In general, all of them presented themselves in an equivalent form in all residual blocks.

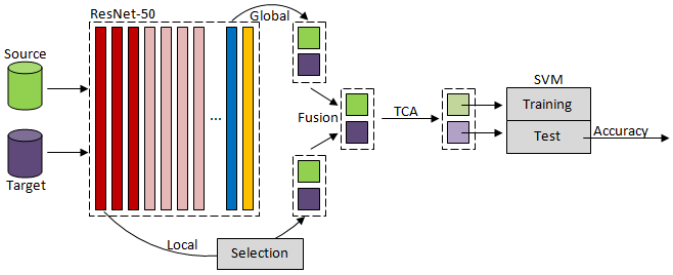

Fig. 2. Considering two datasets, both are passed on to fine-tuned ResNet50 for feature extraction. Initial layers (the red ones) provide low-level features and the pre-prediction layer (the blue one) provides high-level features. In the following, feature fusion is obtained through map concatenation. Using TCA, the resulting feature map is transformed and assigned to SVM [25].

Here, 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.

III. GENERALIZATION OF TRANSFERRED FEATURES: A ONE-CLASS STUDY CASE

One of the reasons for a model not to be completely adaptable to several domains is the absence of generalization, in which these models are often evaluated only by classical measures of the assigned task. Hence, a good measure of domain generalization can indicate which dataset is more suitable to others to improve TL performances.

A. Cross-Domain Feature Space Generalization Measure

We proposed a new metric to evaluate cross-domain TL, asking: **how can one measure generalization of a feature space produced by some method?** The concept of generalization is expressed as $|R_{emp}(f_n) - R(f_n)|$, where $R_{emp}(f_n)$ is the risk of a classifier f_n evaluated over the training set and

TABLE III

ACCURACY (%) FROM FINE-TUNED RESNET50 USING LINEAR SVM. PH2 IS A SMALL DATASET FOR PCA WITH 256 AND 192 FEATURES. VALUES IN BOLD INDICATE THAT THE FUSION PERFORMANCE IS SUPERIOR TO THE GLOBAL ONE [25].

	Training set ⟶ Testing set	TCA Features	Global	Fusion 1st output			Fusion 2nd output			Fusion 3rd output		
				PCA	Flat.	2D	PCA	Flat.	2D	PCA	Flat.	2D
Fruits	Fruits360 [26] ⟶ Supermarket Produce [27]	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**
Objects	Amazon [28] ⟶ Webcam [28]	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**
Skin Lesions	HAM10000 [29] ⟶ PH2 [18]	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**

$R(f_n)$ is the true risk of same f_n over "all data". This idea is totally abstract because it is an intractable quantity, which reveals the importance of not losing ourselves only with classic metrics and training costs. Hence, two metrics were proposed:

$$G_{part}(f_n^A) = \left| \underset{x \in \mathcal{X}^A}{R(f_n^A)} - \underset{x \in \mathcal{X}^B}{R(f_n^A)} \right| \tag{1}$$

$$G_{comp}(f_n^A, f_n^B) = \frac{1}{2}\left(G_{part}(f_n^A) + G_{part}(f_n^B) \right) \tag{2}$$

Considering two domains (A and B) and their respective training feature spaces ($\mathcal{X}^A$ and $\mathcal{X}^B$), $R(f_n^A)$ denotes the risk of test on classifier f_n^A trained over the feature space $\mathcal{X}^A$ and over the feature space $\mathcal{X}^B$. The two functions represent different levels of domain generalization, in which important guidelines should be followed: (i) the set of admissible functions from the classifier/detector are the same; (ii) both feature spaces are described by the same set of descriptors; and (iii) the domain mapping method has no prior knowledge of test data on either domain. Based on the G_{part} and G_{comp}, we introduce three particular levels of domain generalization. Considering the pair results of two methods α and β, the first level of generalization is obtained by:

$$G_{part}(f_\alpha^A) < G_{part}(f_\beta^A) \tag{3}$$

With this inequality satisfied, one could claim that method α is capable of generalizing well from domain A to domain B. Also, we can verify the G_{part} from the "opposite direction" and confirm if the α methodology is also better than β at generalizing from B to A. However, to obtain a more precise and rigorous analysis from both directions, we should compare using G_{comp}:

$$G_{comp}(f_\alpha^A, f_\alpha^B) < G_{comp}(f_\beta^A, f_\beta^B) \tag{4}$$

In all these expressions, lower results imply less divergence, where the concept of generalization is more substantial.

B. CDFG Measure for surveillance videos

To evaluate the practical scenario of CDFG Measure [30] on one-class scenario, it was performed an experiment extracting features via pre-prediction layer of pre-trained VGG-19 [21]. Experiments were designed by: (i) cross-feature embedding, which only relates one training set to another test set; (ii) cross-domain transformation by PCA with 80 features, selecting the components from training set and applying them to the test set; and (iii) latent space by TCA [12], also with 80 features.

Table IV presents that Cross-feature overcomes PCA and TCA in pairs with better performances, especially when Ped2 or Belleview is the target (bold values). However, the average of results between the Cross-feature and TCA is practically negligible: 63.84 versus 62.8. It is important to emphasize that the concept of anomalies among these domains is very dissimilar, implying that the feature learning should not be totally transferred. Hence, considering only domains with the same concept of anomalies (Ped1 and Ped2), TCA stands out when compare to Cross-feature and PCA in average. Although TCA is superior, the classic metric are not enough to guarantee the feature generalization. Analyzing those performances in isolation gives an imprecision due to the great diversity of results achieved. For these reasons, the CDFG Measure [30] offers a more detailed and reliable comparison.

Applying G_{part}, Table V, TCA average (9.0) overcomes PCA (14.7) and Cross-feature (19.54). In context of different concepts of anomalies, there are also great applicability of TCA, implying that the TL is more relevant from Ped1 to Belleview. As expected, Ped1 offers high learning for Ped2, however, the opposite direction does not occur in the same intensity. Another interesting highlight is the PCA performance when compared to Cross-feature, demonstrating that dimensionality reduction increases the generalization. This last remark contradicts the isolated analysis from Table IV, implicating the importance of CDFG Measure. Considering G_{comp}, Table VI, TCA offers even more generalization in relation to the competing methods. Considering similar domains, TCA is

TABLE IV
ANOMALY DETECTION MEASURED BY AREA UNDER THE CURVE [30].

Source ⟶ Target	Cross-feature	PCA	TCA
Ped1 [31] ⟶ Ped1	50.91	**71.46**	62.94
Ped2 [31] ⟶ Ped1	50.82	**64.01**	60.39
Belleview [32] ⟶ Ped1	51.77	**76.12**	58.86
Train [32] ⟶ Ped1	53.42	60.65	**71.02**
Ped2 ⟶ Ped2	**80.34**	55.24	74.16
Ped1 ⟶ Ped2	**80.18**	56.95	67.06
Belleview ⟶ Ped2	**80.88**	69.46	65.16
Train ⟶ Ped2	**81.81**	61.77	50.11
Belleview ⟶ Belleview	68.91	50.54	**72.63**
Ped1 ⟶ Belleview	**68.67**	56.22	68.39
Ped2 ⟶ Belleview	**68.73**	60.42	65.24
Train ⟶ Belleview	**69.1**	54.36	68.65
Train ⟶ Train	53.97	**57.67**	51.88
Ped1 ⟶ Train	54.02	**57.75**	53.98
Ped2 ⟶ Train	54.13	55.47	**55.56**
Belleview ⟶ Train	53.85	50.63	**58.86**

TABLE V
PARTIAL CDFG MEASURE USING AREA UNDER THE CURVE [30].

Source ⟶ Target	Cross-feature	PCA	TCA
Ped2 ⟶ Ped1	29.52	**8.77**	13.77
Belleview ⟶ Ped1	17.14	25.58	**14.27**
Ped1 ⟶ Ped2	29.27	14.51	**4.12**
Belleview ⟶ Ped2	11.97	18.92	**7.47**
Ped1 ⟶ Belleview	17.76	15.24	**5.45**
Ped2 ⟶ Belleview	11.61	**5.18**	8.92

TABLE VI
COMPLETE CDFG MEASURE USING AREA UNDER THE CURVE [30].

Datasets	Cross-feature	PCA	TCA
(Ped1, Ped2)	29.4	11.6	**8.95**
(Ped1, Belleview)	17.5	20.4	**9.86**
(Ped2, Belleview)	11.79	12.05	**8.19**
(Train, Ped1)	**1.83**	8.35	14.1
(Train, Ped2)	27.0	**2.17**	10.2
(Train, Belleview)	15.1	**1.7**	15.3

highly applicable: 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). However, the Train video presents an anomaly concept very distinct from the others. As the results demonstrate, Train is not a suitable domain for Ped1, Ped2, or Belleview, causing negative transfer.

These results express the applicability of CDFG Measure to indicate which domains offer the most learning rate for a target domain. As mentioned earlier, classic evaluation metrics do not provide a perform of model generalizability. With the CDFG Measure we confirmed that TCA (the only TL method) 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 TL.**

IV. FEATURE TL IN SEMI-SUPERVISED SETTINGS

When fine-tuning is applied in deep networks, one of the concerns is about how much data is required [34]. In this context, **if unlabeled data is available, how to use those to improve the final learned representation?** Semi-supervised networks are architectures that do not require much labels [35]. Hence, a combination of CNN and AE provides a hybrid network that conciliates all available data simultaneously [36].

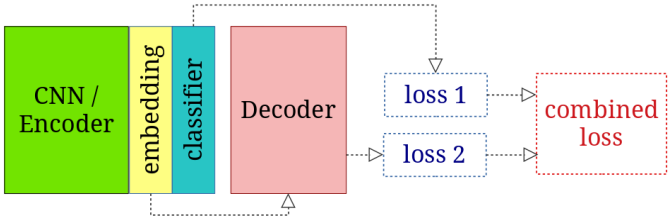

Fig. 3. Hybrid architecture: combination of supervised (loss 1) and unsupervised (loss 2) networks and their losses to learn a feature embedding [33].

A. Weighted Label Loss

Our semi-supervised architecture is composed of a CNN and an AE that share intermediate layers and optimize the combined loss function through the amount of labeled examples, as shown in Fig.3. Therefore, the model applies supervised (Cross-entropy loss $l^{(ce)}$) and unsupervised (Mean Square Error ϵ) 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. Our semi-supervised network is trained in two steps: (i) the AE is trained using only the unlabeled training data; and (ii) the hybrid network is fine-tuned using the remaining labeled data. Hence, given a percentage of labeled data P from the training set, the first stage trains the AE from scratch using $(100-P)\%$ examples. In the following, the whole network is fine-tuned using the remaining $P\%$ of labeled examples by WLL:

$$WLL = \left(0.5 + 0.5 \cdot \frac{P}{100}\right) \cdot l^{(ce)} + \left(0.5 - 0.5 \cdot \frac{P}{100}\right) \cdot \epsilon \quad (5)$$

The first term describes the classification weight w_{sup} while the second one defines the reconstruction weight w_{uns}. Moreover, we have $0.5 < w_{sup} < 1.0$ and $0 < w_{uns} < 0.5$. Consequently, this balancing ensures that $w_{sup} + w_{uns} = 1$. Also, the constraint $0 < P < 100$ should occur. Therefore, when $P = 100$ all data are labeled and only the CNN must be considered; when $P = 0$ all data are unlabeled and only the AE must be considered.

B. Features embedding on semi-supervised learning

Two architectures were investigated: sequential convolutional forming the SmallNet (SN); and residual blocks for SmallResNet (SRN), changing the number of blocks (SRN-1, SRN-2, and SRN-4). Focus on learning discriminative representations, the networks are used as feature extraction modules after the training, selecting the Embedding to generate the feature maps. Analyzing the overall results, Fig.4, WLL are superior than regular combination of individual losses.

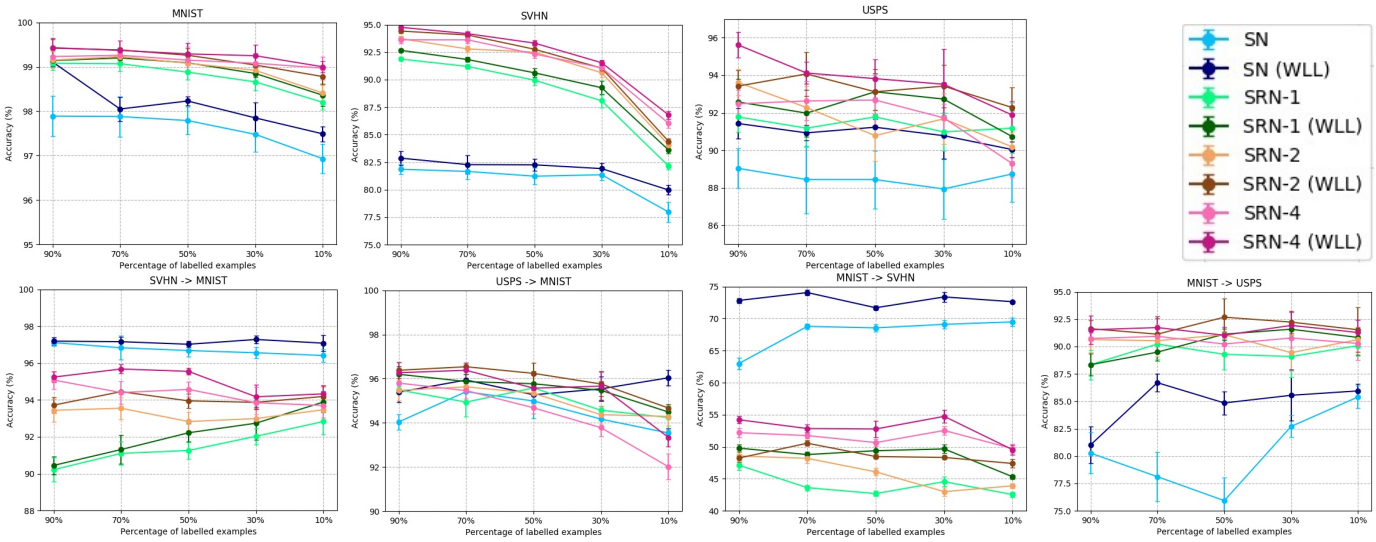

Fig. 4. Semi-supervised accuracies with different proportions of labels per training data: MNIST [37]; SVHN [38]; and USPS (https://cs.nyu.edu/∼roweis/data.html). The results include training and testing with the same dataset (first row) as well as across datasets (training set ⟶ test set). The SVM classifier is applied to perform a 5-fold cross validation. Also, for each network, we compared WLL to a regular balancing with the CNN (0.5) and AE (0.5) [33]

Furthermore, a few performances with $P = 90\%$ are even better than the supervised approach: MNIST⟶SVHN (47.5% – 54.21%) and MNIST⟶USPS (89.44% – 91.53%) in SRN-4; and SVHN⟶MNIST in all networks ($\approx 1\%$). As expected, the 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. 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, SN (74.06%) overcomes the other networks (48.78% SRN-1, 50.55% SRN-2, and 52.84% SRN-4 with $P = 70\%$). Considering SVHN as source, SRNs have their performances improved due to greater parameters fluctuation obtained during the first step of training.

Also, considering STL-10 training set (unlabeled for AE and labeled for WLL), three experiments were performed: (i) using STL-10 test set; (ii) using CIFAR-10 with only the 9 common classes; and (iii) using CIFAR-10 with all 11 classes. Given the results, WLL excels a better performance than regular combination, reinforcing previous results with digit images domain, with: (i) 49.34% to 51.95%; (ii) 40.57% to 42.18%; and (iii) 38.27% to 40.11%. Comparing both performances on CIFAR-10, it is interesting to note that the WLL accuracy remains equivalent (42.18% — 40.11%), which indicates that the architecture generalizes well to unknown classes.

In general, WLL presented high performances than regular weight and few labeled instances provided a considerable increase in the performance, where it can even surpass supervised scenarios. **Since WLL loss function is dependent only on the proportion of labeled examples, it is adaptable to different scenarios with low computational complexity**.

V. Conclusion

In this thesis, we established a new manner to development predictive models involving feature transfer learning, in which the descriptors contained in the CNNs are explored and combined to enhance the pattern recognition performance. Additionally, we contributed with a new measure to assess the model generalization in feature learning, even for unlabeled data. This can have a positive impact on methods already available, improving generalization abilities. Hence, we believe that our results represent a guideline for building new models, and allow to further explore the representativeness of deep networks with appropriate validation metrics. Future work may expand our metric to multiple domains simultaneously as well as finding new ways to combine features from different layers to improve representations.

VI. Publications

This PhD Thesis was defended in January 2020 and approved unanimously by the evaluating members resulting in 6 papers (3 journals, 3 conferences). This manuscript focus on the results published at SIBGRAPI 2018 [19]; SIBGRAPI 2019 [25]; Journal of Visual Communication and Image Representation [30]; and Neural Networks [33]. Due to space constraints we do not describe additional results also published at Applied Soft Computing [39] and CAIP 2019 [40].

Acknowledgment

The authors would like to thank UGPN-RCF for the technical visit at University of Surrey, as well as Prof. Josef Kittler and Dr. Cemre Zor. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: a survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[3] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[4] S. J. Pan, Q. Yang *et al.*, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[5] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4068–4076.

[6] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 325–333.

[7] B. Sengupta and K. J. Friston, "How robust are deep neural networks?" *arXiv preprint arXiv:1804.11313*, 2018.

[8] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global, 2010, pp. 242–264.

[9] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[10] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[11] M. Ponti, L. S. Ribeiro, T. S. Nazare, T. Bui, and J. Collomosse, "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)*, 2017, pp. 17–41.

[12] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[13] R. F. Mello and M. A. Ponti, *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Springer, 2018.

[14] V. Pomponiu, H. Nejati, and N.-M. Cheung, "Deepmole: Deep neural networks for skin mole lesion classification," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2623–2627.

[15] A. Mahbod, R. Ecker, and I. Ellinger, "Skin lesion classification using hybrid deep neural networks," *arXiv preprint arXiv:1702.08434*, 2017.

[16] T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg, "Combining deep learning and hand-crafted features for skin lesion classification," in *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*. IEEE, 2016, pp. 1–6.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira, "Ph 2-a dermoscopic image database for research and benchmarking," in *2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2013, pp. 5437–5440.

[19] F. P. dos Santos and M. A. Ponti, "Robust feature spaces from pre-trained deep network layers for skin lesion classification," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2018, pp. 189–196.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] M. Ponti, T. S. Nazaré, and G. S. Thumé, "Image quantization as a dimensionality reduction procedure in color and texture feature extraction," *Neurocomputing*, vol. 173, pp. 385–396, 2016.

[24] L. Bi, J. Kim, E. Ahn, D. Feng, and M. Fulham, "Automatic melanoma detection via multi-scale lesion-biased representation and joint reverse classification," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2016, pp. 1055–1058.

[25] F. P. dos Santos and M. A. Ponti, "Alignment of local and global features from multiple layers of convolutional neural network for image classification," in *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2019, pp. 241–248.

[26] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2018.

[27] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic produce classification from images using color, texture and appearance cues," in *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2008, pp. 3–10.

[28] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.

[29] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions," *arXiv preprint arXiv:1803.10417*, 2018.

[30] F. P. dos Santos, L. S. Ribeiro, and M. A. Ponti, "Generalization of feature embeddings transferred from different video anomaly detection domains," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 407–416, 2019.

[31] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1975–1981.

[32] A. Zaharescu and R. Wildes, "Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing," in *European Conference on Computer Vision*. Springer, 2010, pp. 563–576.

[33] F. P. dos Santos, C. Zor, J. Kittler, and M. A. Ponti, "Learning image features with fewer labels using a semi-supervised deep convolutional network," *Neural Networks*, 2020.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. Hoi, and Z. Xu, "Semi-supervised deep embedded clustering," *Neurocomputing*, vol. 325, pp. 121–130, 2019.

[36] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2215–2223.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[38] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, 2011, p. 5.

[39] M. A. Ponti, G. B. P. da Costa, F. P. Santos, and K. U. Silveira, "Supervised and unsupervised relevance sampling in handcrafted and deep learning features obtained from image collections," *Applied Soft Computing*, vol. 80, pp. 414–424, 2019.

[40] F. P. dos Santos and M. A. Ponti, "Homogeneity index as stopping criterion for anisotropic diffusion filter," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2019, pp. 269–280.